



Digital Design

Chapter 2: Combinational Logic Design

Slides to accompany the textbook *Digital Design*, First Edition,
by Frank Vahid, John Wiley and Sons Publishers, 2007.
<http://www.ddvahid.com>

Copyright © 2007 Frank Vahid

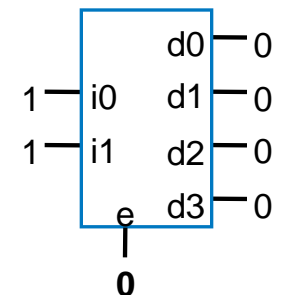
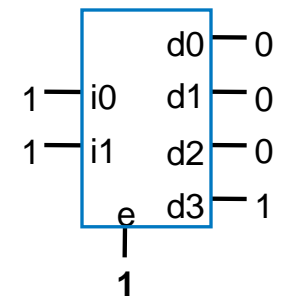
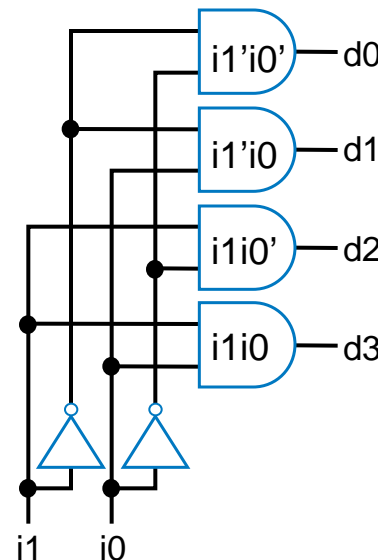
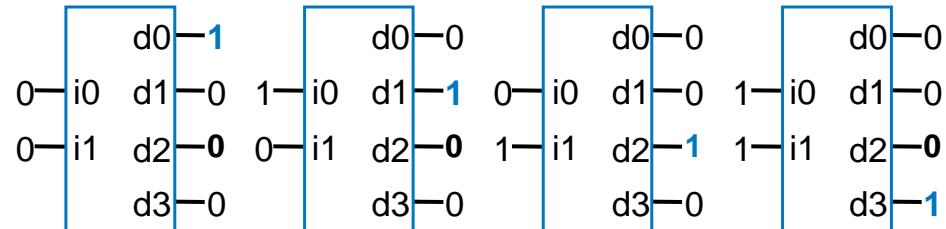
Instructors of courses requiring Vahid's *Digital Design* textbook (published by John Wiley and Sons) have permission to modify and use these slides for customary course-related activities, subject to keeping this copyright notice in place and unmodified. These slides may be posted as unanimated pdf versions on publicly-accessible course websites.. PowerPoint source (or pdf with animations) may not be posted to publicly-accessible websites, but may be posted for students on internal protected sites or distributed directly to students by other electronic means. Instructors may make printouts of the slides available to students for a reasonable photocopying charge, without incurring royalties. Any other use requires explicit permission. Instructors may obtain PowerPoint source or obtain special use permissions from Wiley – see <http://www.ddvahid.com> for information.

Decoder

- A decoder is a logic device that has **m** selection inputs and **n** outputs, where $n \leq 2^m$. A single output is asserted for each unique combination of input values. A decoder is basically a minterm generator. There may also be an enable input that controls whether the output is asserted.
- **Applications**
 - BCD to decimal – one of ten outputs is asserted that corresponds to the specific BCD code value.
 - Memory address decoder – the address bits are selection inputs to the decoder, thus enabling a certain block of memory
 - Minterm generator – the asserted output is the minterm corresponding to the selection input bits.

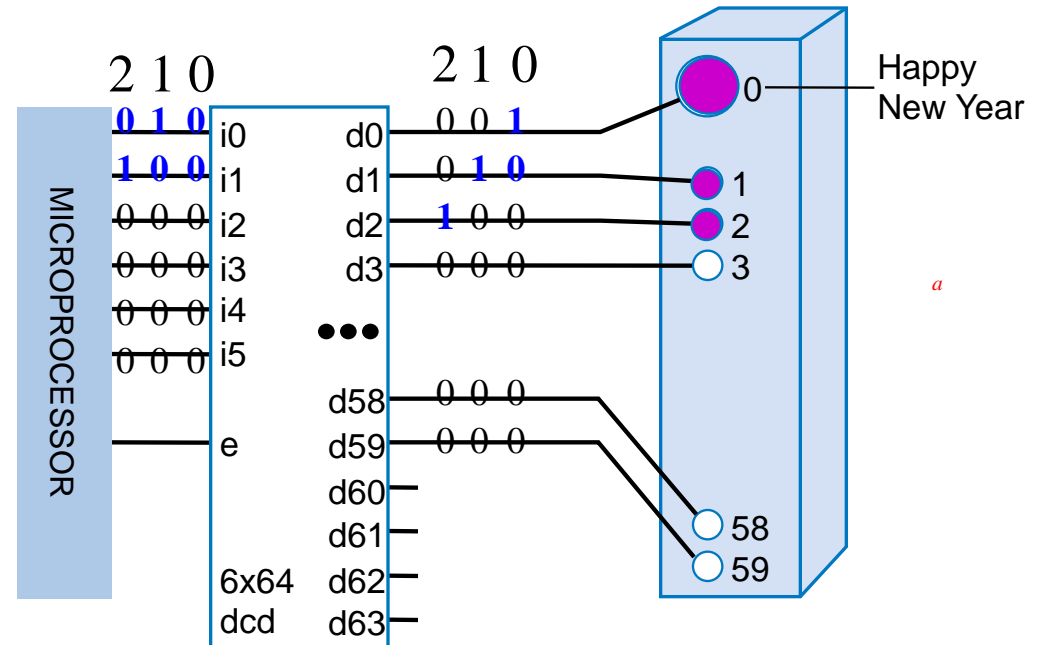
Decoders

- **Decoder:** Popular combinational logic building block, in addition to logic gates
 - Converts input binary number to one high output
- 2-input decoder: four possible input binary numbers
 - So has four outputs, one for each possible input binary number
- Internal design
 - AND gate for each output to detect input combination
- Decoder with enable e
 - Outputs all 0 if $e=0$
 - Regular behavior if $e=1$
- n -input decoder: 2^n outputs

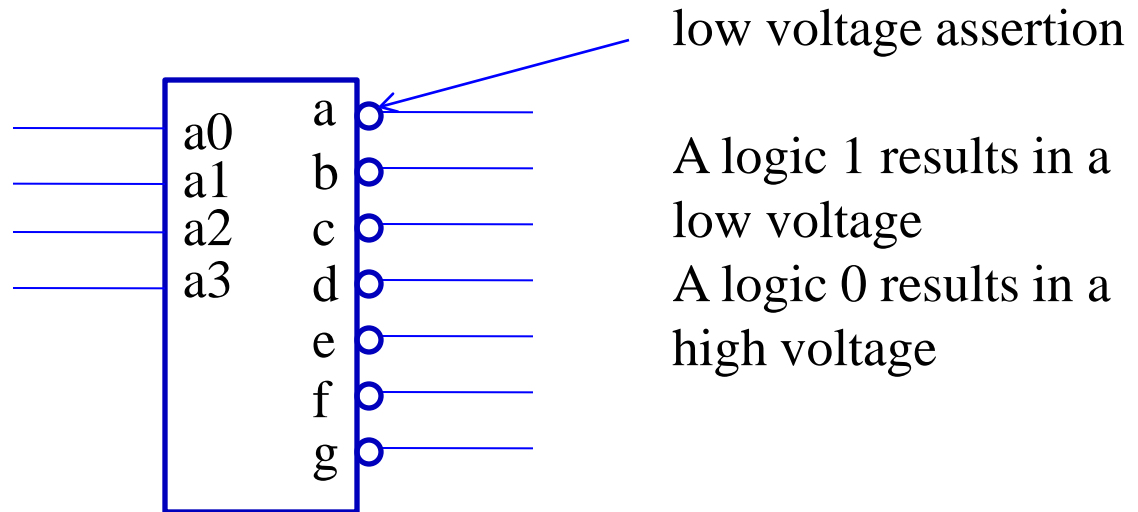


Decoder Example

- New Year's Eve Countdown Display
 - Microprocessor counts from 59 down to 0 in binary on 6-bit output
 - Want illuminate one of 60 lights for each binary number
 - Use 6x64 decoder
 - 4 outputs unused



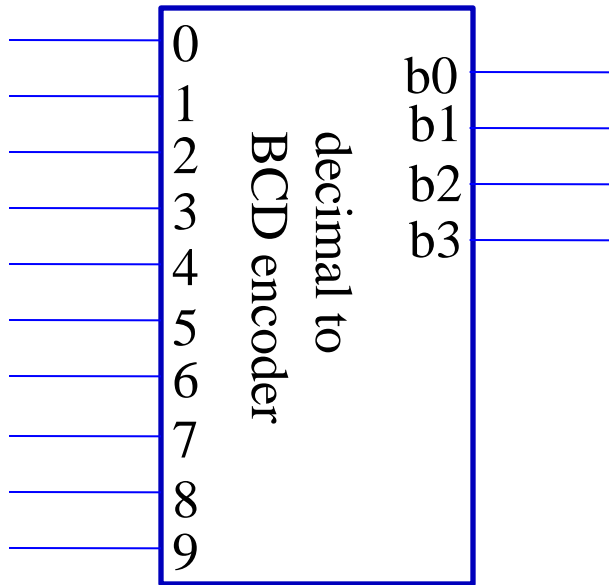
BCD to Seven-Segment Decoder



Encoder

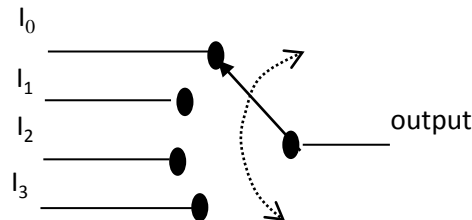
- A logic encoder is a device that translates either a single or a set of Boolean variables into a coded representation. An encoder with **m** Boolean inputs requires **n** outputs, where $m \leq 2^n$.
- **Encoder Examples**
 - Keyboard to ASCII – 128 keys could be represented with a 7-bit output code.
 - Decimal to BCD – 10 inputs are expressed in BCD code with 4 bits.
 - Rotary positional encoder – an 8-bit output code can represent an angle resolution of $360^\circ / 256$.

Decimal to BCD Encoder



Multiplexer

- A multiplexer is a device that has **m** selection control signals, 2^m inputs, and one output. The output is equal to one of the inputs, depending on the value of the selection control signals. For example, if $n = 2$, and given the control signals S_1 and S_0 , then the output O is expressed by the Boolean equation $O = S_1'S_0'I_0 + S_1'S_0I_1 + S_1S_0'I_2 + S_1S_0I_3$. The signals I_m are the inputs. A multiplexer can be modeled as an m -position rotary switch.



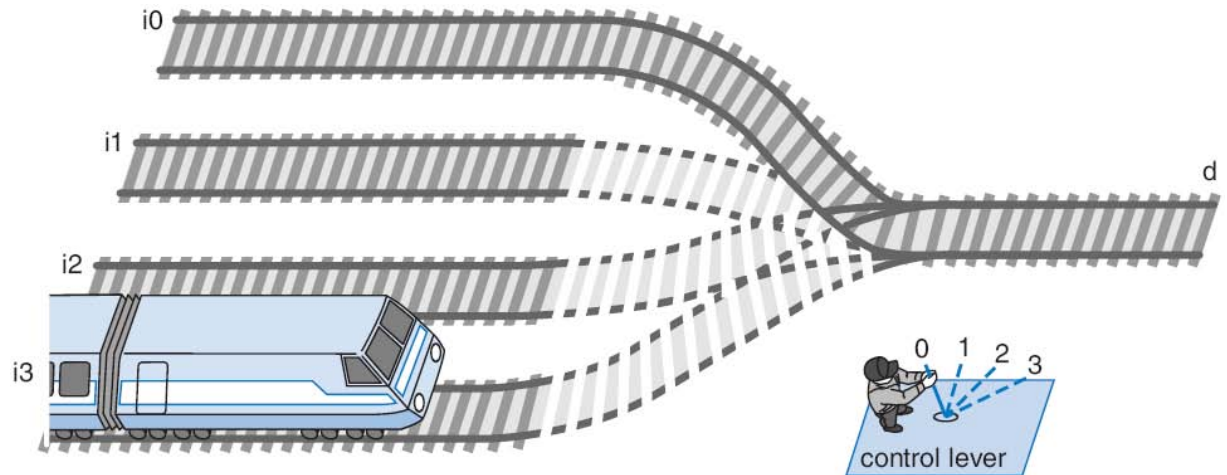
Multiplexer

- **Applications**

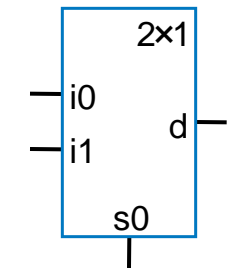
- Parallel-to-serial conversion – modems, local area network, and printers.
- Input monitoring – router, multiple computers with one printer, and burglar alarm.
- Boolean equation generator – allows selection of asserted minterms.

Multiplexer (Mux)

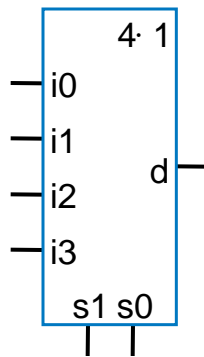
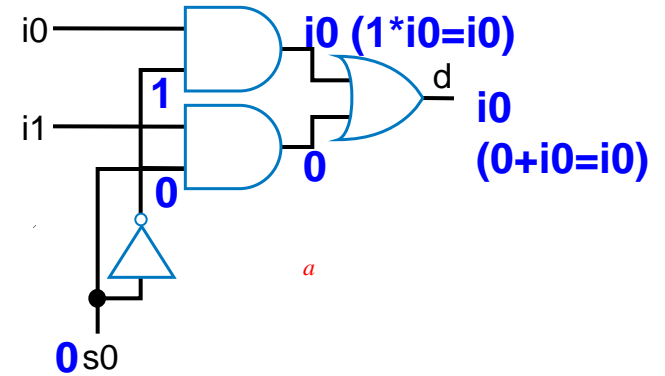
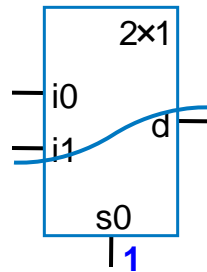
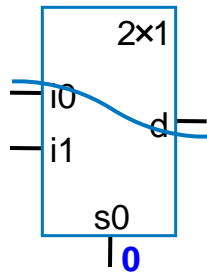
- Mux: Another popular combinational building block
 - Routes one of its N data inputs to its one output, based on binary value of select inputs
 - 4 input mux \rightarrow needs 2 select inputs to indicate which input to route through
 - 8 input mux \rightarrow 3 select inputs
 - N inputs $\rightarrow \log_2(N)$ selects
 - Like a railyard switch



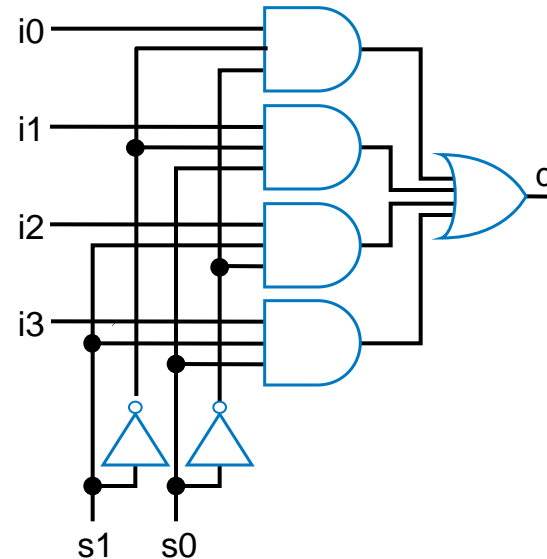
Mux Internal Design



2x1 mux

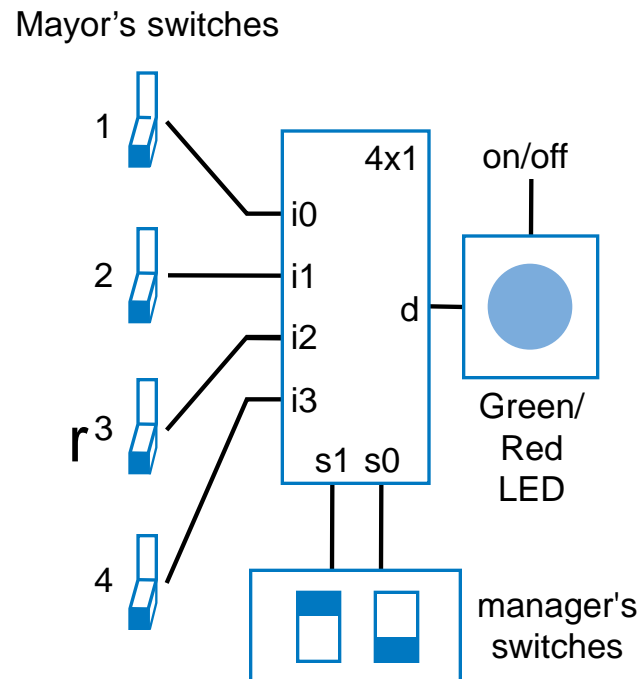


4x1 mux

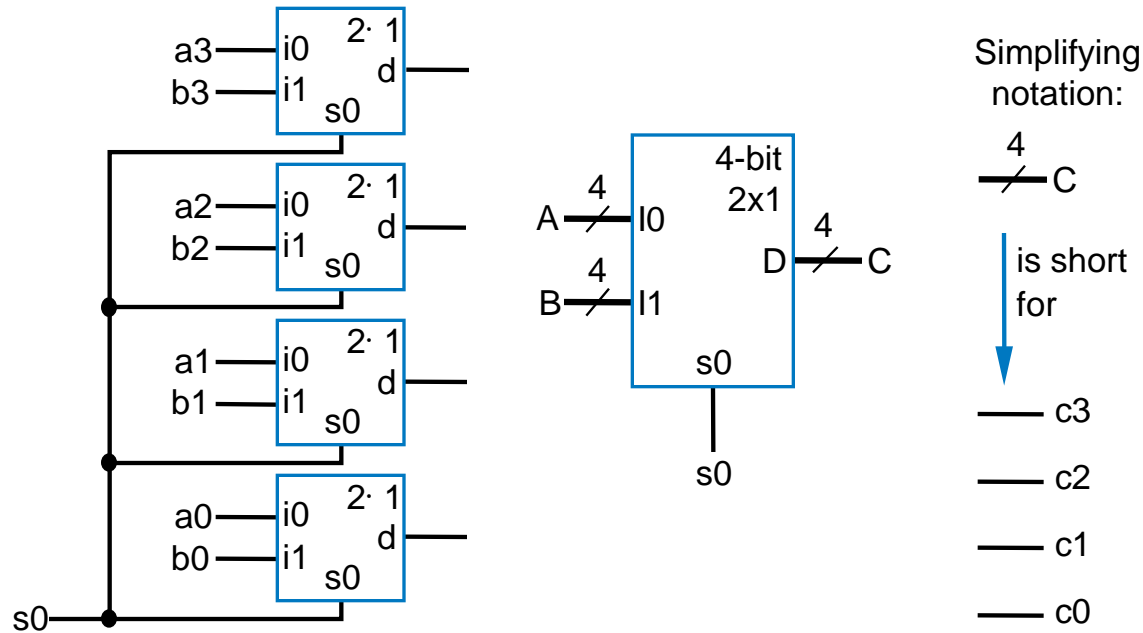


Mux Example

- City mayor can set four switches up or down, representing his/her vote on each of four proposals, numbered 0, 1, 2, 3
- City manager can display any such vote on large green/red LED (light) by setting two switches to represent binary 0, 1, 2, or 3
- Use 4x1 mux



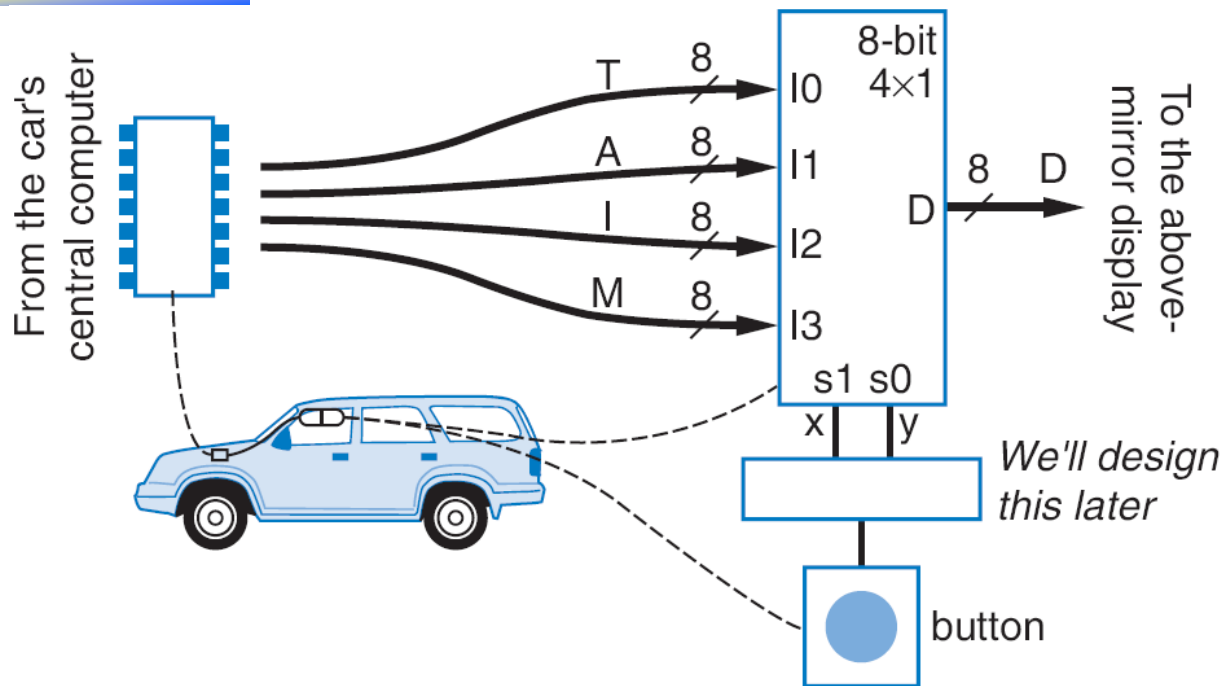
Muxes Combined Together -- N-bit Mux



- Ex: Two 4-bit inputs, A (a3 a2 a1 a0), and B (b3 b2 b1 b0)
 - 4-bit 2x1 mux (just four 2x1 muxes sharing a select line) can select between A or B



N-bit Mux Example

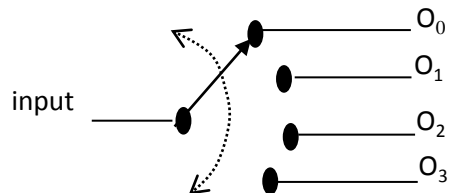


- Four possible display items
 - Temperature (T), Average miles-per-gallon (A), Instantaneous mpg (I), and Miles remaining (M) -- each is 8-bits wide
 - Choose which to display using two inputs x and y
 - Use 8-bit 4x1 mux



Demultiplexer

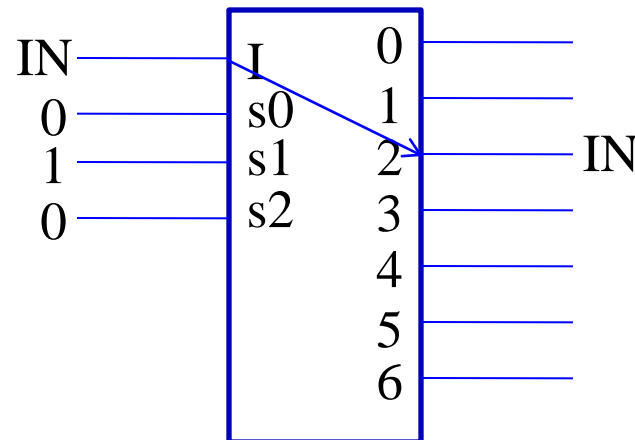
- A demultiplexer is a device that has one input, m selection control signals, and 2^m outputs. A demultiplexer performs the inverse operation of a multiplexer: a single input is directed to one of the outputs based on the value of the selection word. The Boolean equations that describe a 1-to-4 demultiplexer with input I , are: $O_3 = S_1 S_0 I$, $O_2 = S_1 S_0' I$, $O_1 = S_1' S_0 I$, $O_0 = S_1' S_0' I$. A demultiplexer can be modeled as an m -position rotary switch.



Demultiplexer

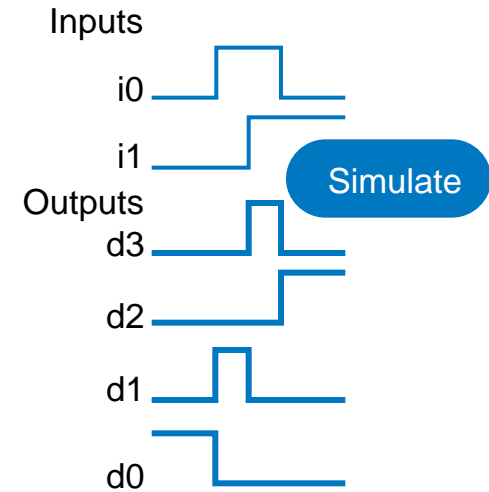
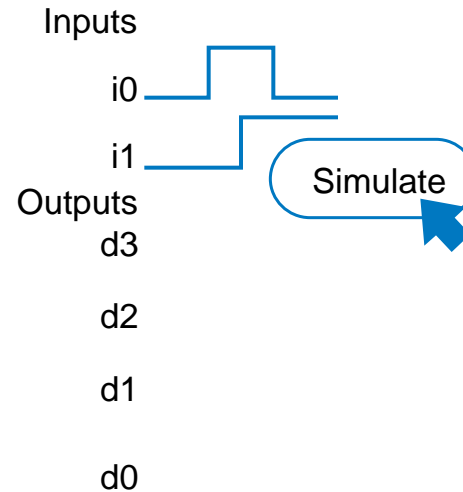
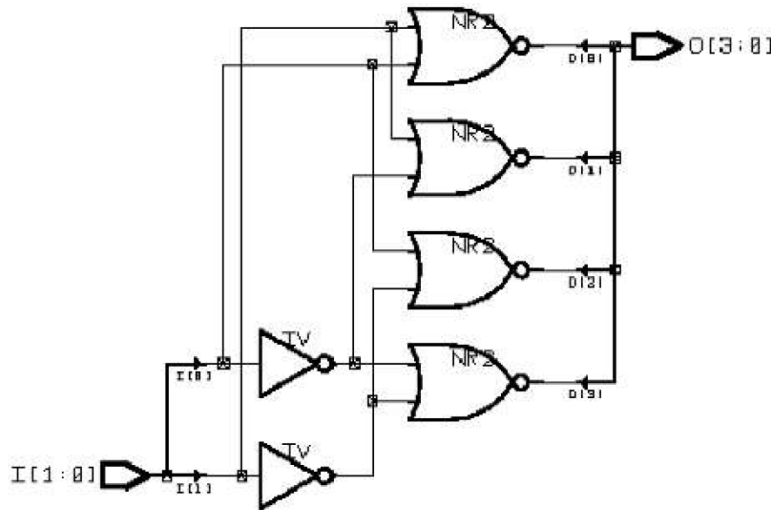
- **Applications**

- Serial-to-Parallel conversion – modems, local area network, and printers.
- Boolean equation generator – allows selection of asserted minterms.



Additional Considerations

Schematic Capture and Simulation



- **Schematic capture**

- Computer tool for user to capture logic circuit graphically

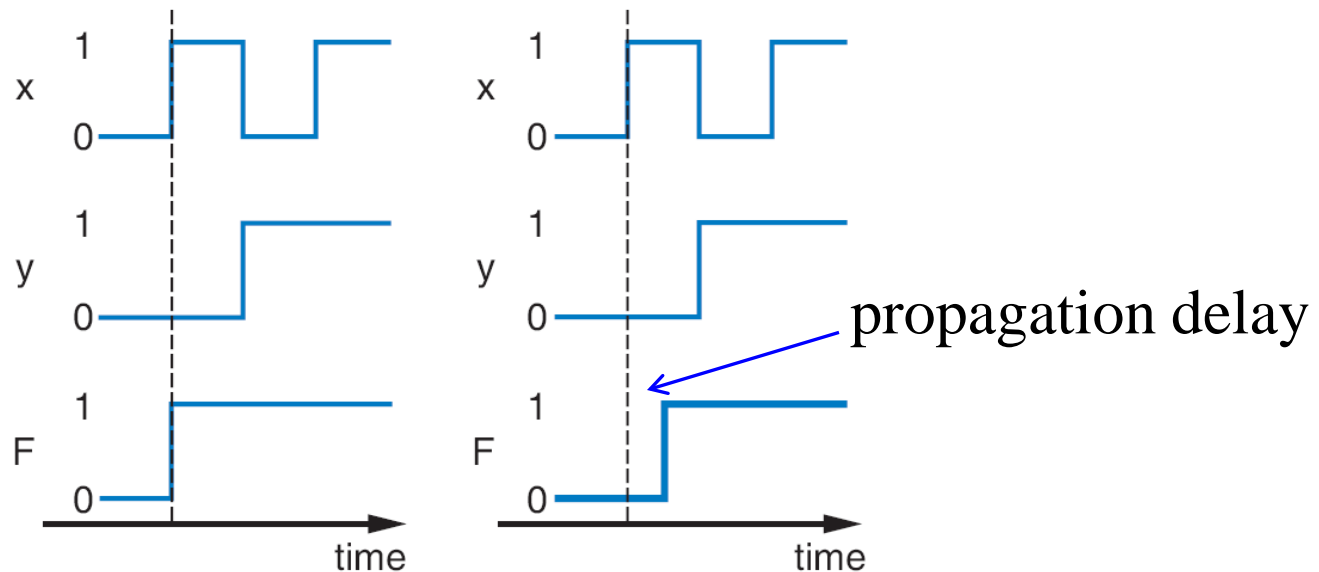
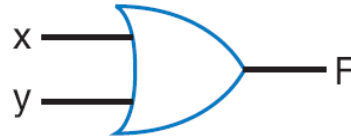
- **Simulator**

- Computer tool to show what circuit outputs would be for given inputs
 - Outputs commonly displayed as **waveform**



Additional Considerations

Non-Ideal Gate Behavior -- Delay



- Real gates have some delay
 - Outputs don't change immediately after inputs change



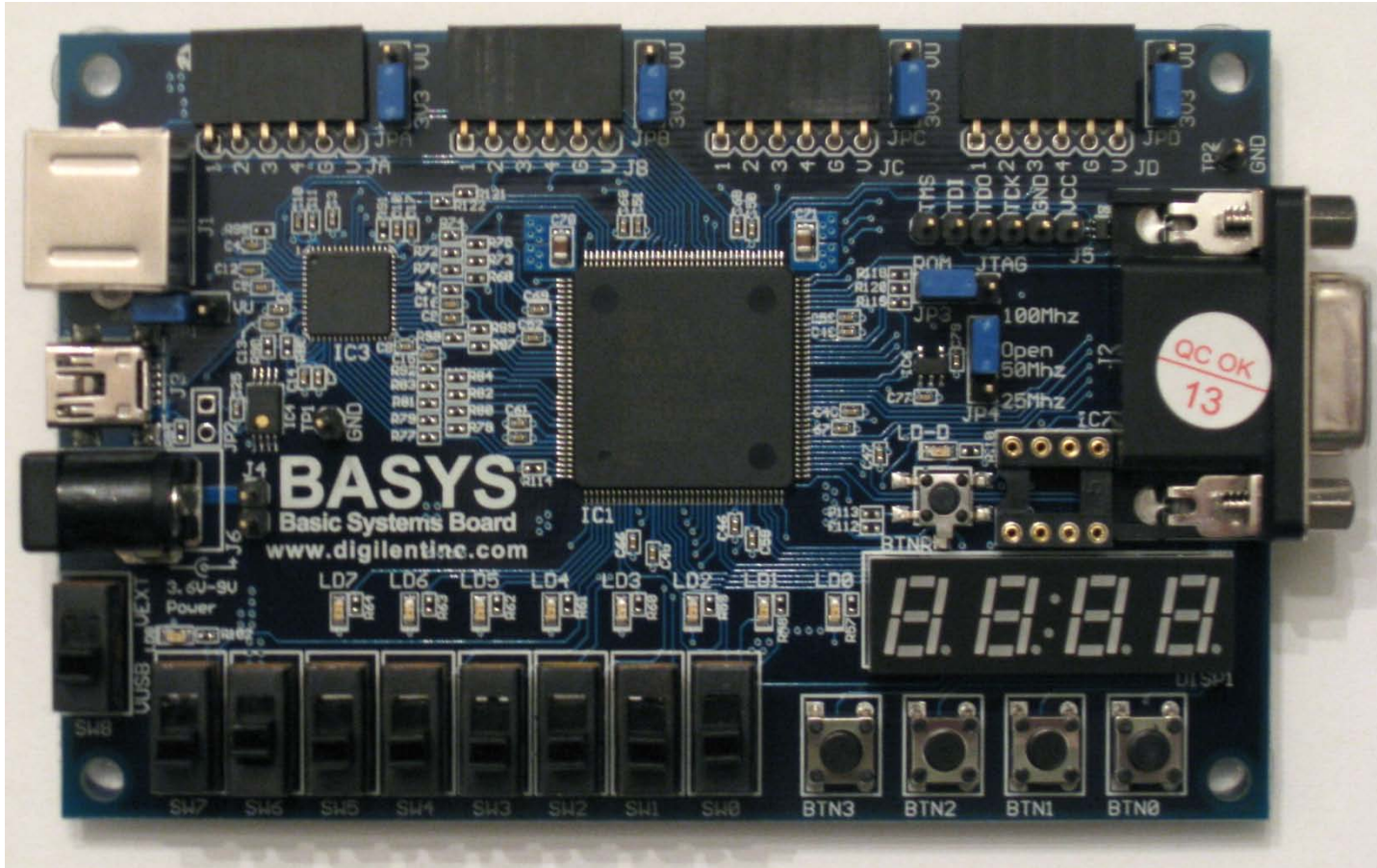
Homework

- Chapter 2: 69, 72, 76, 80
- Homework is due on Thursday, February 4
- Refer to next slides for information on next lab

Notes on Next Lab

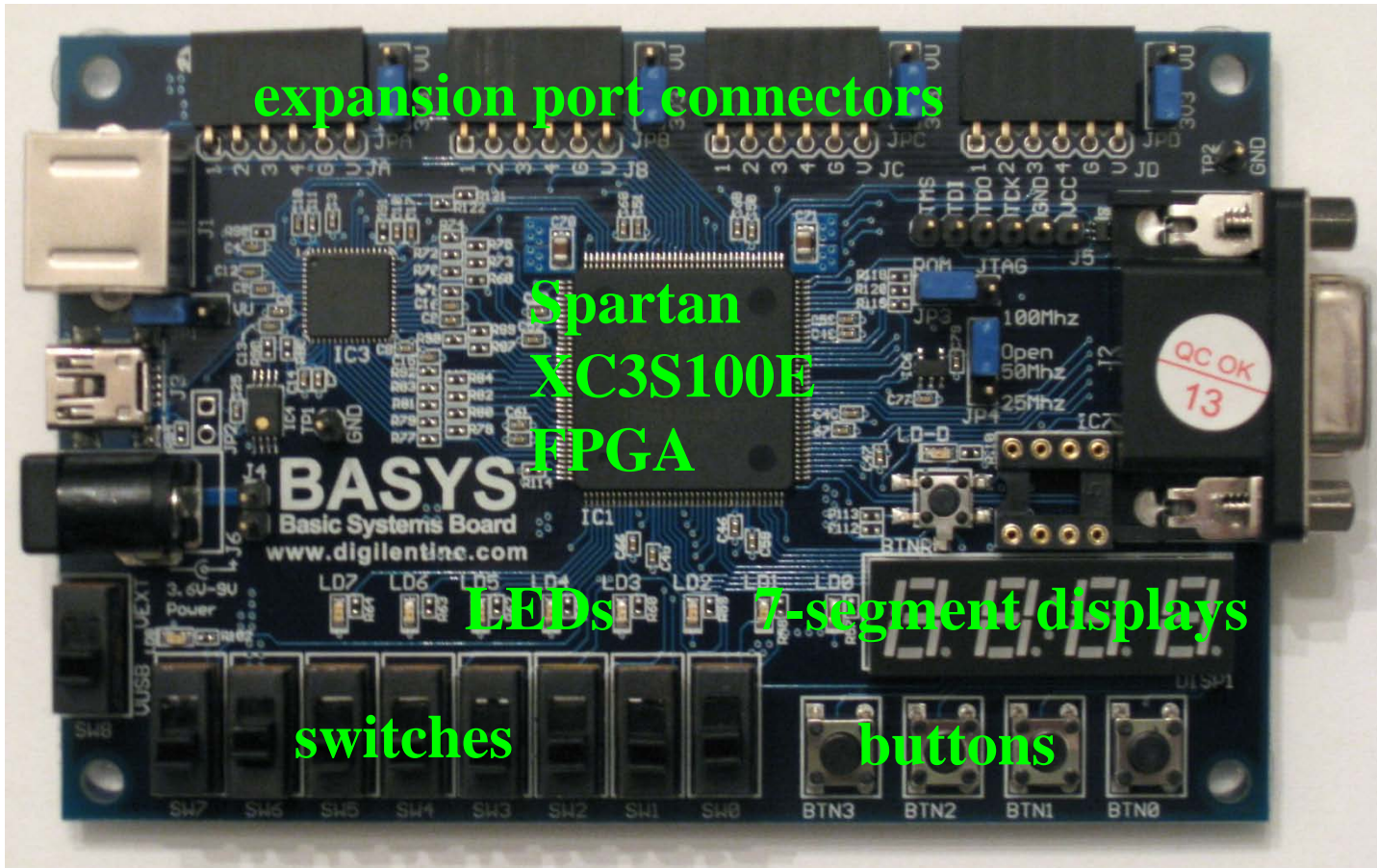
- The objective of the lab is to learn how to implement a logic circuit using the Digilent Basys board.
- The Digilent board allows for quick prototyping
- The Digilent board contains a field-programmable gate array (FPGA) that can be easily configured to implement logic expressions.

Digilent Basys Board, USB Version



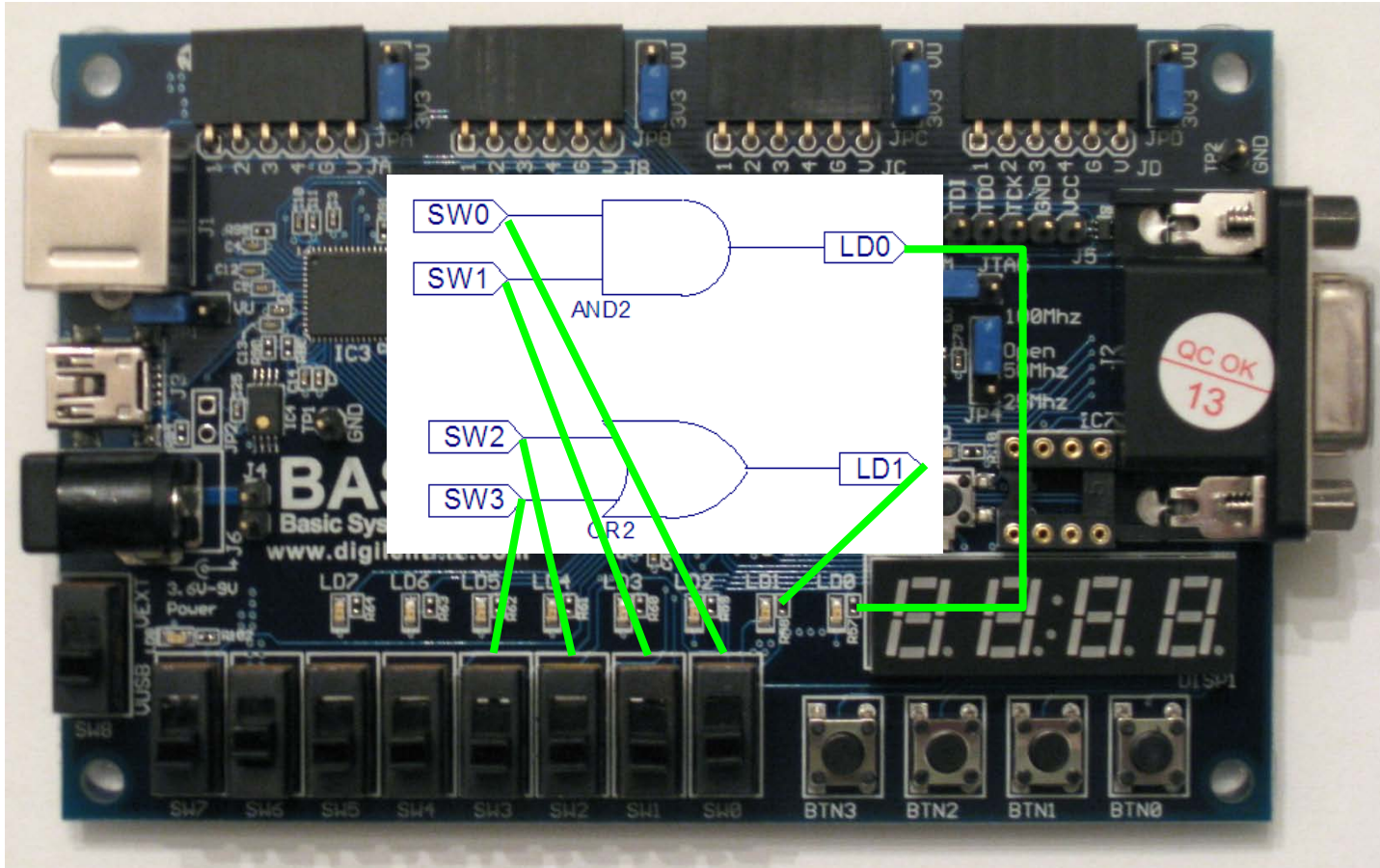
www.digilentinc.com/Data/Products/BASYS/BASYS_E_RM.pdf

Digilent Basys Board, USB Version



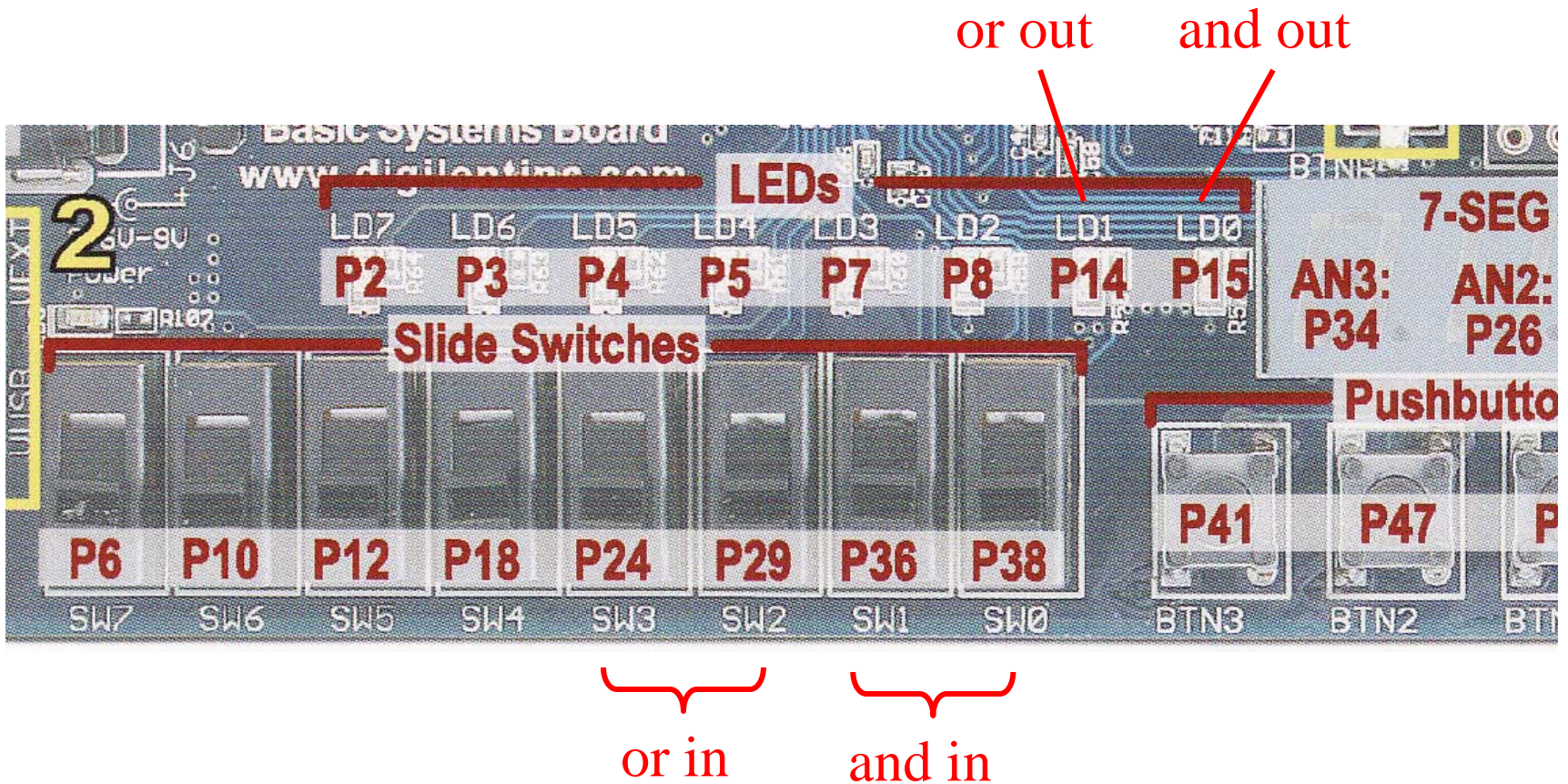
www.digilentinc.com/Data/Products/BASYS/BASYS_E_RM.pdf

Digilent Basys Board, USB Version



www.digilentinc.com/Data/Products/BASYS/BASYS_E_RM.pdf

Digilent Basys Board, USB Version



www.digilentinc.com/Data/Products/BASYS/BASYS_E_RM.pdf