Lecture #5: Combinational Logic Idioms

Paul Hartke
Phartke@stanford.edu
Stanford EE121
January 22, 2001

Book Reading

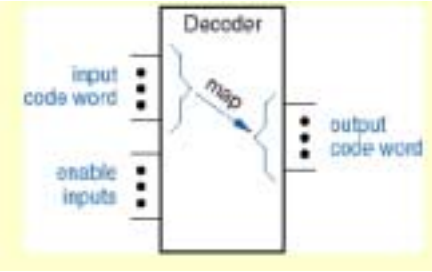
- Chapter 1
- Chapter 2.1-2.6
- Chapter 3.1.-3.7
- Chapter 4.1-4.3, 4.5
- Chapter 5.3, 5.4.1, 5.4.8, 5.5.1, 5.7.0, 5.8.1-5.8.4, 5.9.1-5.9.3, 5.10.1-5.10.7, 5.11.1
  - Know the architectures involved but don't need the TTL equivalents or ABEL/VHDL
- Chapter 10.5-10.6
  - Difference between CPLD and FPGA architecture

Combinational Logic

- Could implement all logic with K-maps directly from specifications.
  - Tedious and error prone
- Just as in programming use existing blocks that raise the level of abstraction.
- Some CL has a special purpose that is often used.

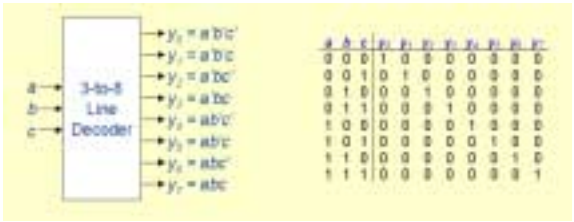
5.4 Decoders

- General Decoder Structure
  - Typically n-inputs, 2^n ouputs
  - 2-4, 3-8, 4-16 etc.



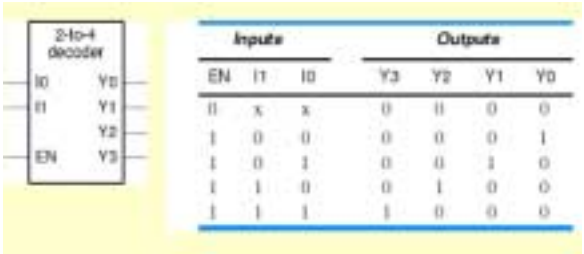
General Structure

- One of these will be needed for Lab 3



Binary 2-to-4 Decoder

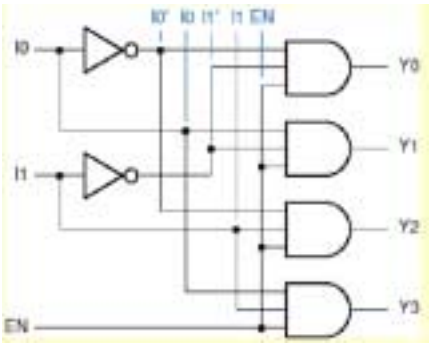
- What is the difference between this and a 2-1 mux?
  - Note: “x” means don’t care



Inputs			Outputs			
EN	I1	I0	Y3	Y2	Y1	Y0
0	x	x	0	0	0	0
1	0	0	0	0	0	1
1	0	1	0	0	1	0
1	1	0	0	1	0	0
1	1	1	1	0	0	0

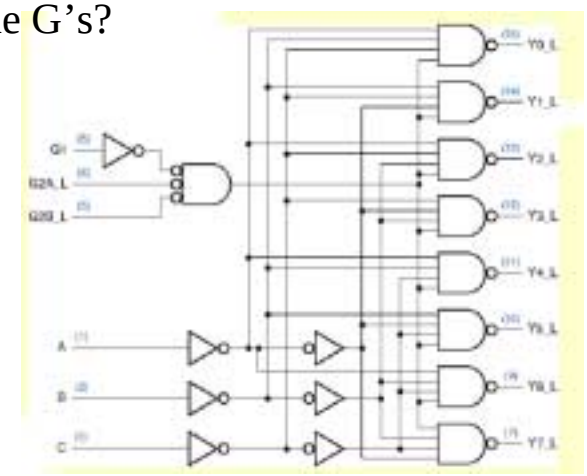
Binary 2-to-4 Decoder Implementation

- Any difference?



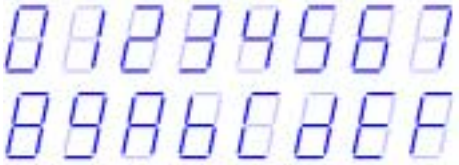
3-to-8 Decoder

- What are the G's?



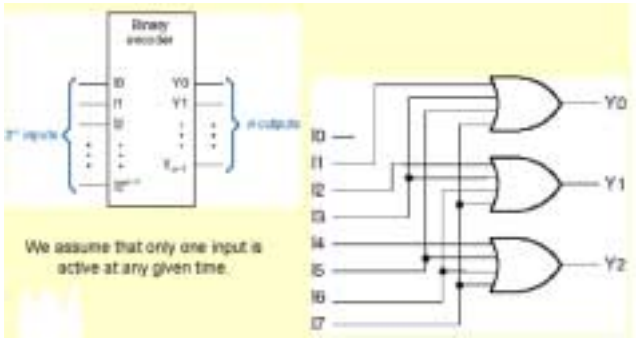
Seven-Segment LED Driver

- Just a special decoder
- How many Karnaugh maps will it take to design the logic for a seven-segment display driver?
  - Create a Karnaugh map for output e of the seven-segment display driver.



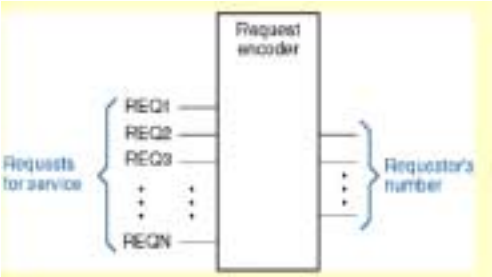
5.5 Binary Encoder

- What if more than one output is active?
  - Not necessarily bad.



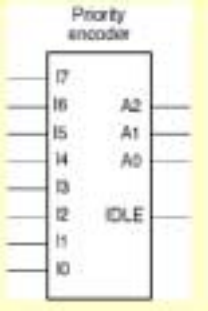
Need Priority in Most Applications

- Report input with highest priority among potentially many requests.



8 input priority encoder

- Also need to report when there are no matches.



Priority-Encoder Logic Equations

First we define eight intermediate variables H0-H7:

$H7 = I7$

$H6 = I6 \cdot I7'$

$H5 = I5 \cdot I6 \cdot I7'$

...

$H0 = I0 \cdot I1' \cdot I2' \cdot I3' \cdot I4' \cdot I5' \cdot I6' \cdot I7'$

Using these signals, we can define the equations for A2-A0:

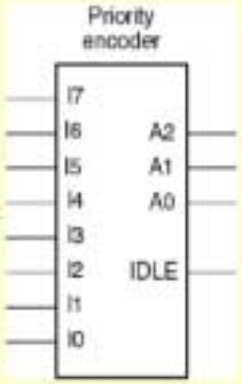
$A2 = H4 + H5 + H6 + H7$

$A1 = H2 + H3 + H6 + H7$

$A0 = H1 + H3 + H5 + H7$

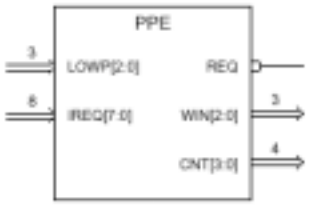
Finally:

$IDLE = I0' \cdot I1' \cdot I2' \cdot I3' \cdot I4' \cdot I5' \cdot I6' \cdot I7'$



Programmable Priority Encoder

- That's nice but it would be cooler if we could change the priority so a specified input is the highest/lowest.
- Lab 3! ☺



5.8 2-input XOR Gate

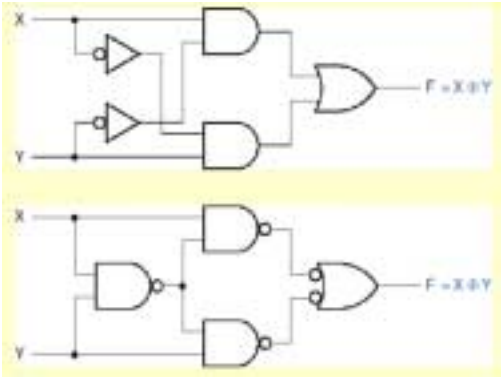
- Like an OR gate, but excludes the case where both inputs are 1.

X	Y	X⊕Y	(X⊕Y)'
0	0	0	1
0	1	1	0
1	0	1	0
1	1	0	1

- XNOR: Complement of XOR

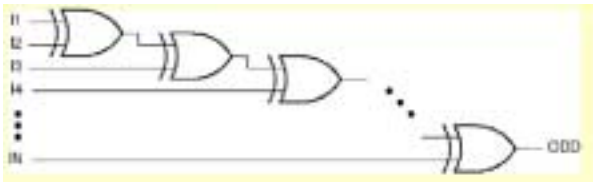
Gate Level XOR Implementation

- Note more expensive in CMOS than AND/OR but not more expensive in LUT FPGA.



Multi-Input XOR

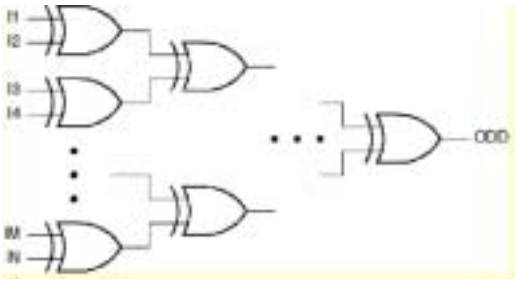
- Sum Modulo 2
- Parity Computation



- Used to generate and check parity bits in computer systems.
  - Detects any single-bit error.
    - What about 2, 3, 4, etc?

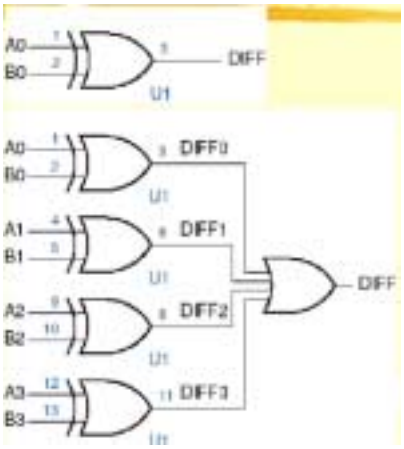
Balanced Trees are Faster

- Why?
  - Useful for many architectures…



5.9 Comparators

- 1 bit and 4 bit

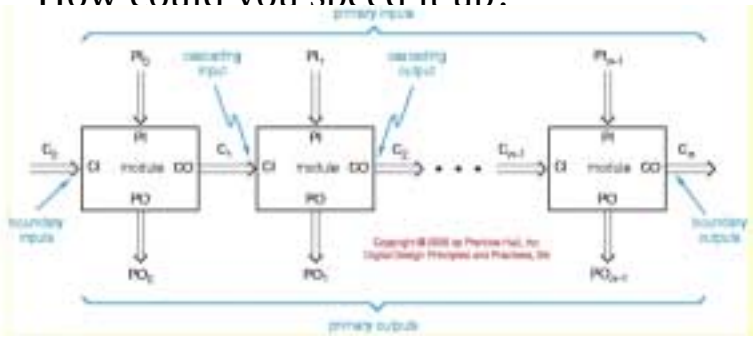


Inequality Comparator?

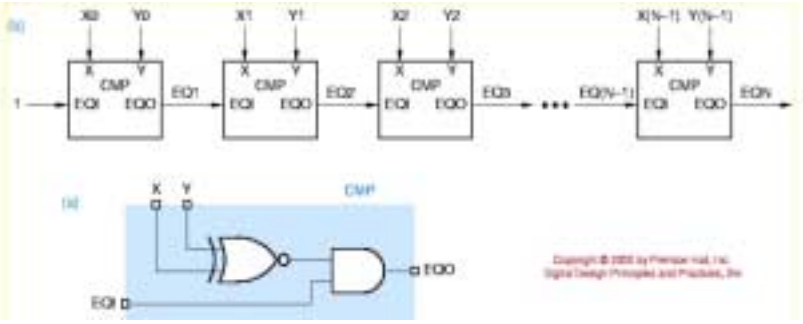
- What is the algorithm you use to compare two numbers?
  - Ie: 12345 vs 12645
- Transfer this to logic gates

Iterative Combinational Circuit

- Very General structure
  - Size?, speed?
- How could you speed it up?



Iterative Comparator Circuit



Next Time

- Arithmetic
  - Adders and Multipliers