



University of
Massachusetts
Amherst

Engin112 – Lecture 23

Modular Logic Design

Maciej Ciesielski
Department of Electrical and Computer Engineering
10/31/2011

Recap from last lectures

- Standard arithmetic components (combinational)
 - Adders/subtractors
 - Multipliers
 - Comparators
 - Decoders, Encoders
 - Multiplexers (MUX), deMultiplexers

Today's lecture:

- Designing combinational logic
 - Standard procedure (review)
 - » Truth table, K-maps, gate-level network
 - Modular design
 - » Implementing combinational logic with decoders and MUXes

Custom Approach to Logic Design

- Standard approach to logic design
 - Define the problem: problem specification
 - Create truth table (or write Boolean equations directly)
 - Simplify logic expressions
 - » Karnaugh map
 - » Logic minimization software
- This approach is used for unstructured (custom) logic
 - Control logic
 - “Glue” logic
 - “Random” logic

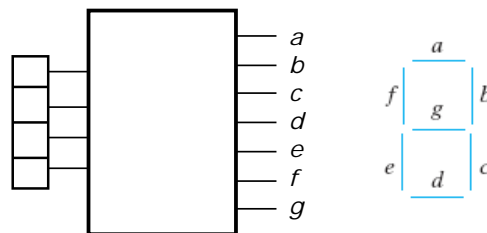
10/31/2011

Engin 112 - Intro to ECE

3

BCD to 7-segment Display

- Problem (4.9). Design a BCD-to7segment display decoder.



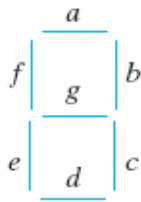
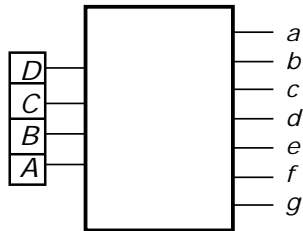
10/31/2011

Engin 112 - Intro to ECE

4

BCD to 7-segment Display

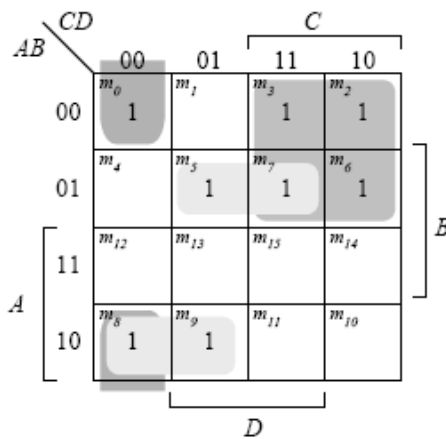
- Step 1. Create Boolean expressions for each segment



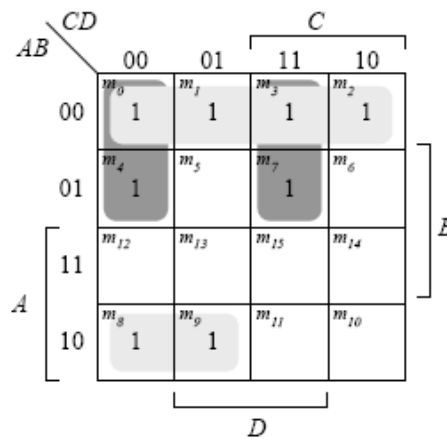
ABCD	a	b	c	d	e	f	g
0000	1	1	1	1	1	1	0
0001	0	1	1	0	0	0	0
0010	1	1	0	1	1	0	1
0011	1	1	1	1	0	0	1
0100	0	1	1	0	0	1	1
0101	1	0	1	1	0	1	1
0110	1	0	1	1	1	1	1
0111	1	1	1	0	0	0	0
1000	1	1	1	1	1	1	1
1001	1	1	1	1	0	1	1

BCD to 7-segment Display

- Step 2. Minimize Boolean expressions



$$a = A'C + A'BD + B'C'D' + AB'C'$$



$$b = A'B' + A'C'D' + A'CD + AB'C'$$

etc.

- Step 3. Create a gate-level network (4 inputs, 7 outputs)

Modular Approach to Logic Design

- Alternative approach: modular design
 - Using Decoders
 - » Generate minterms
 - » Need to combine selected minterms to generate function
 - Using Multiplexer
 - » Output of multiplexer is function
 - » Choose right inputs to generate the function

10/31/2011

Engin 112 - Intro to ECE

7

Combinational Logic with Decoders

- What are the outputs of the decoder?

- Minterms

x	y	z	D0	D1	D2	D3	D4	D5	D6	D7
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	1	0	0
1	1	0	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	1

- Combinational function:

- $F = \sum m_i = \sum D_i$
- Collect the required decoder outputs (minterms)

- What do we need to add to the decoder to implement an *arbitrary* combinatorial function?

- an OR gate to collect the minterms

10/31/2011

Engin 112 - Intro to ECE

8

Full Adder using Decoder

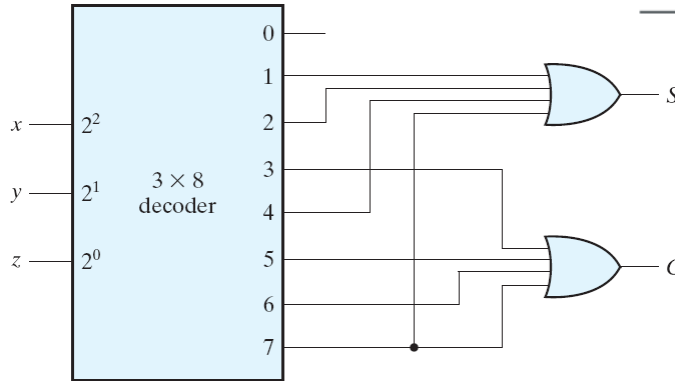
- Functions of Full Adder:

- $S(x,y,z) = \Sigma(1,2,4,7)$
- $C(x,y,z) = \Sigma(3,5,6,7)$

Full Adder

x	y	z	C	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

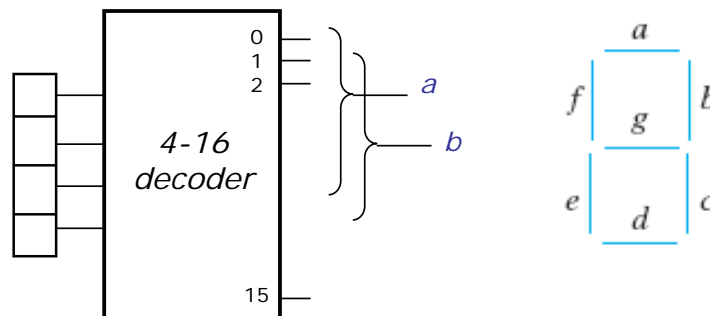
- Implementation (using decoder):



BCD to 7-segment Display

- Design using decoders – try it yourself !

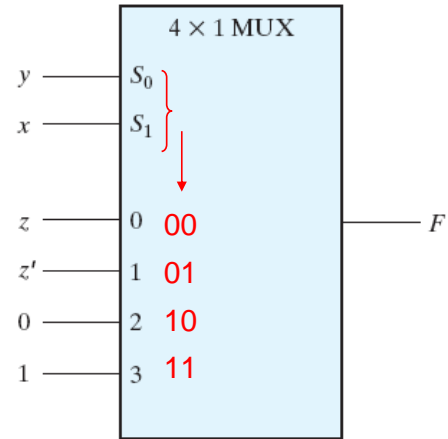
- $a = \Sigma(0,2,3,5,6,7,8,9)$
Is there a better way to do it ? (try $a' = \dots$)
- $b = \Sigma(0,1,2,3,4)$
- etc.



Combinational Logic with Multiplexers

- Can be done efficiently using MUX
 - 2^{n-1} -to-1 multiplexer
- Idea:
 - Wire up $n-1$ variables to selection bits
 - Function might depend on n^{th} variable
 - » Use for 2^{n-1} inputs
- Example:

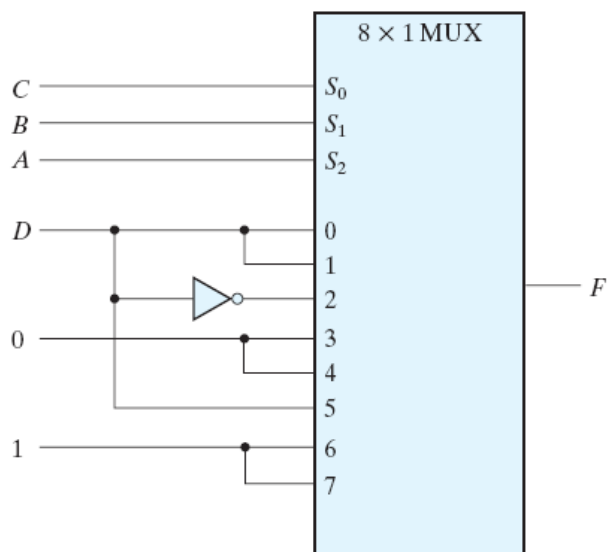
x	y	z	F
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1



Combinational Logic with Multiplexers

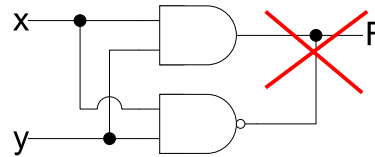
- Truth table:

A	B	C	D	F
0	0	0	0	0
0	0	0	1	1
0	0	1	0	0
0	0	1	1	1
0	1	0	0	1
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1



Three-State (Tri-state) Gates

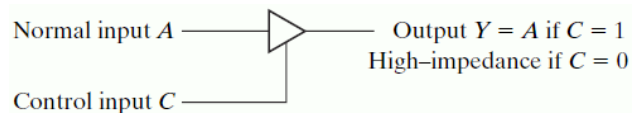
- What is the truth table for F ?



- The two gates will try to drive F at the same time
 - Not a good idea to wire their outputs
- Sometimes it is necessary to “disconnect” a gate

- Three state:

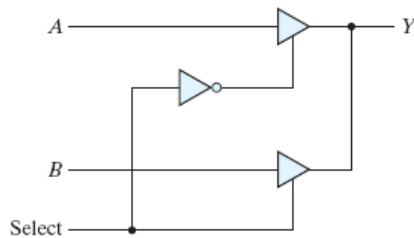
- 0 or 1 Boolean value
- “High impedance”, Z state



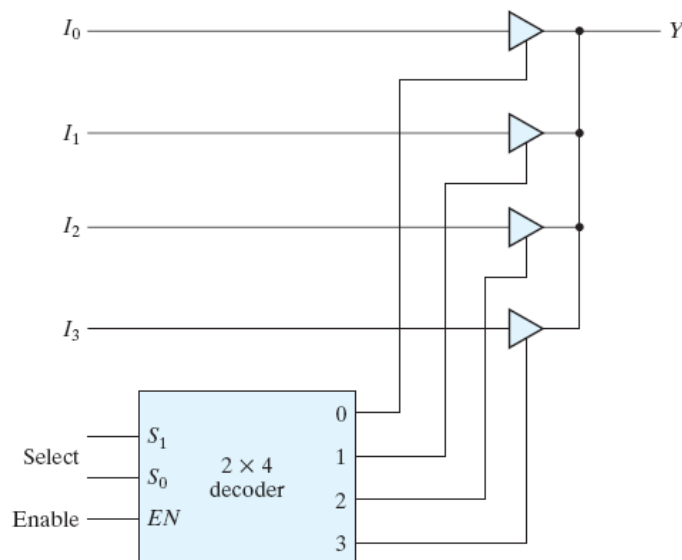
- High impedance acts as if gate were disconnected

Multiplexer with Tri-state Gates

- 2-to-1 MUX:



- 4-to-1 MUX with Enable:



Reading Assignment

- Starting sequential logic (the real stuff !)
 - Mano 5.1 - 5.3 (sequential circuits and Latches).