
Digital Design: A Systems Approach

Lecture 5: Combinational Review

Review

- Lecture 1 – Digital abstraction
- Lecture 2 – Combinational logic design
- Lecture 3 – Combinational building blocks
- Lecture 4 – Numbers and arithmetic
- Lecture 5 – Quiz 1 Review

Digital Signals

- Noise margins
 - $V_{OL} = 0.1V$, $V_{OH} = 0.9V$, $V_{IL} = 0.5V$, $V_{IH} = 0.7V$
 - What are the high and low noise margins?
 - How much variation in power supply (with GNDs equal) can be tolerated?
 - How would you adjust these settings?
 - Assume $V_{IH} - V_{IL} \geq 0.2V$.

Digital Signals

- Representation
 - A single signal can represent a _____
 - Represent a set with a _____
 - Can be either _____ or _____
- Example
 - Represent a chessboard

Boolean Algebra

- Simplify the expressions
 - $f = (x \wedge y \wedge z) \vee (x \wedge y) \vee (x' \wedge y \wedge z)$
 - $f = (x \wedge y \wedge z) \vee (x \wedge y') \vee (x \wedge z')$

Duals and Complements

- Given a function $f(a,b,c)$
- Its complement is $f'(a,b,c)$
- Its dual is $f'(a',b',c')$

- Example - $f = \text{AND}$, complement is _____, dual is _____

- Example - $f = 3\text{-bit prime number function}$
 - f is one at 1,2,3,5,7
 - f' is one at _____
 - f_D is one at _____

Manual Combinational Design

- Start with a specification
- Write K-map
- Find all prime implicants
- Pick a minimal set of prime implicants that *covers* the function
- Example,
 - 4-bit Fibonacci circuit

Combinational Logic Design – Karnaugh maps are one method

Consider a 4-input Fibonacci circuit

True when input = 1,2,3,5,8, or 13

ba \ a	00	01	11	10
0	0 ₀	1 ₁	1 ₃	1 ₂
1	0 ₄	1 ₅	0 ₇	0 ₆
2	0 ₁₂	1 ₁₃	0 ₁₅	0 ₁₄
3	1 ₈	0 ₉	0 ₁₁	0 ₁₀

Implicants (dcba)

1000

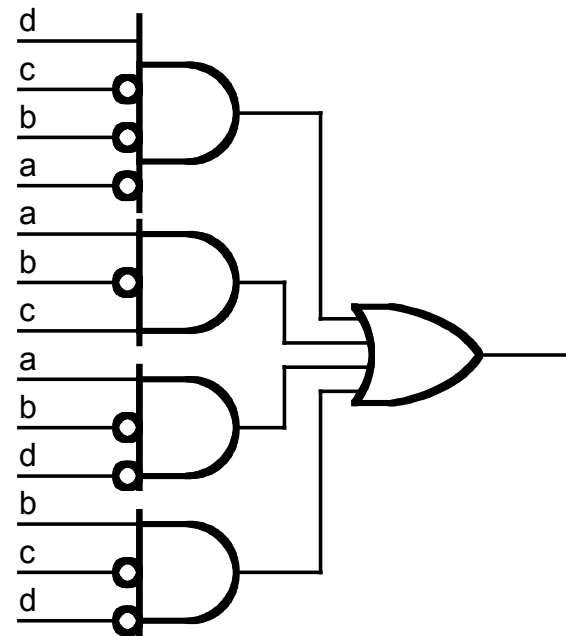
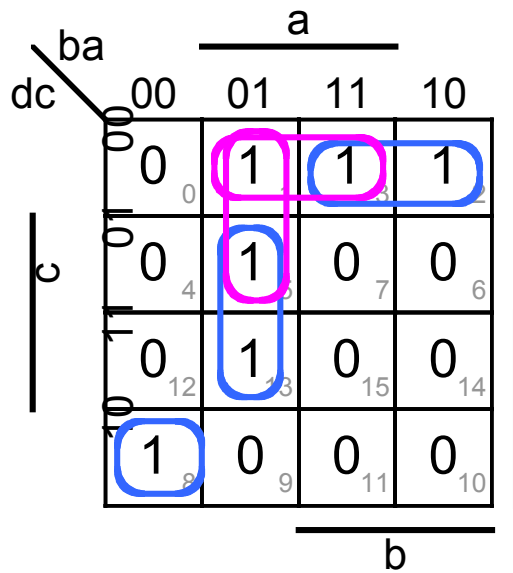
x101

0x01

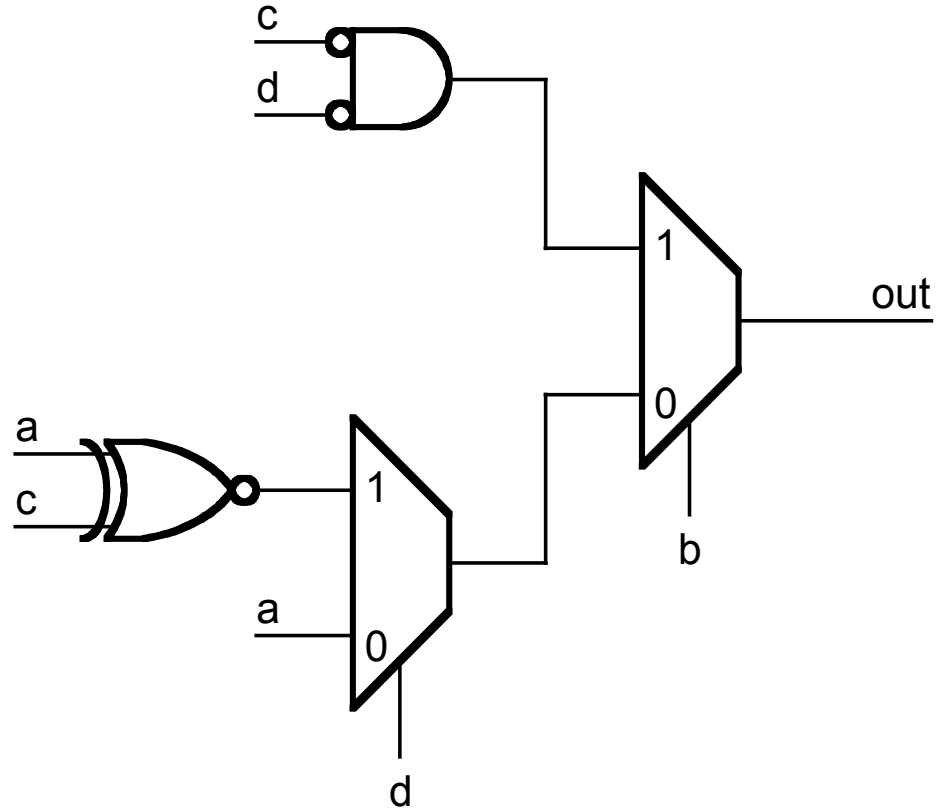
00x1

001x

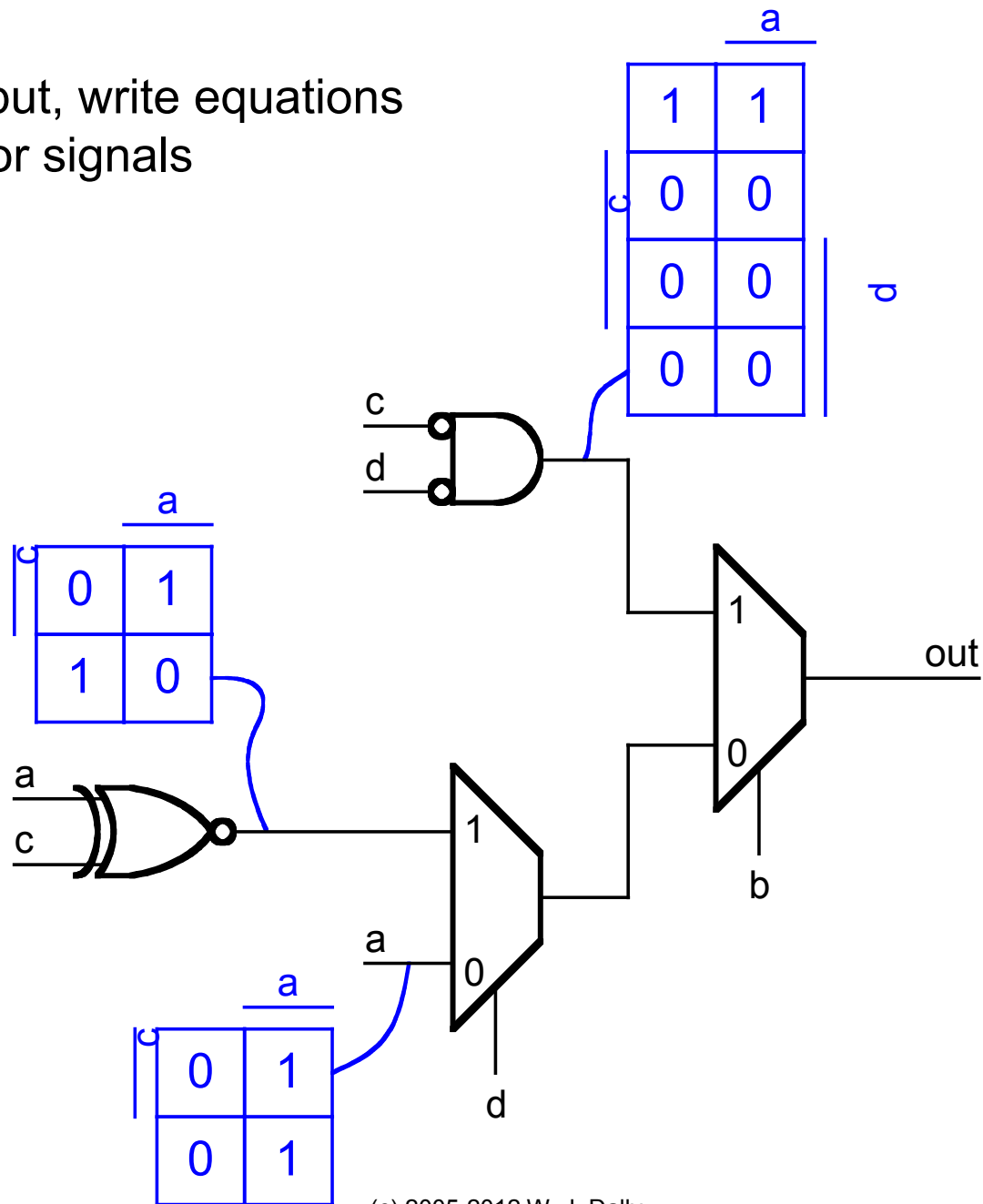
Three are essential

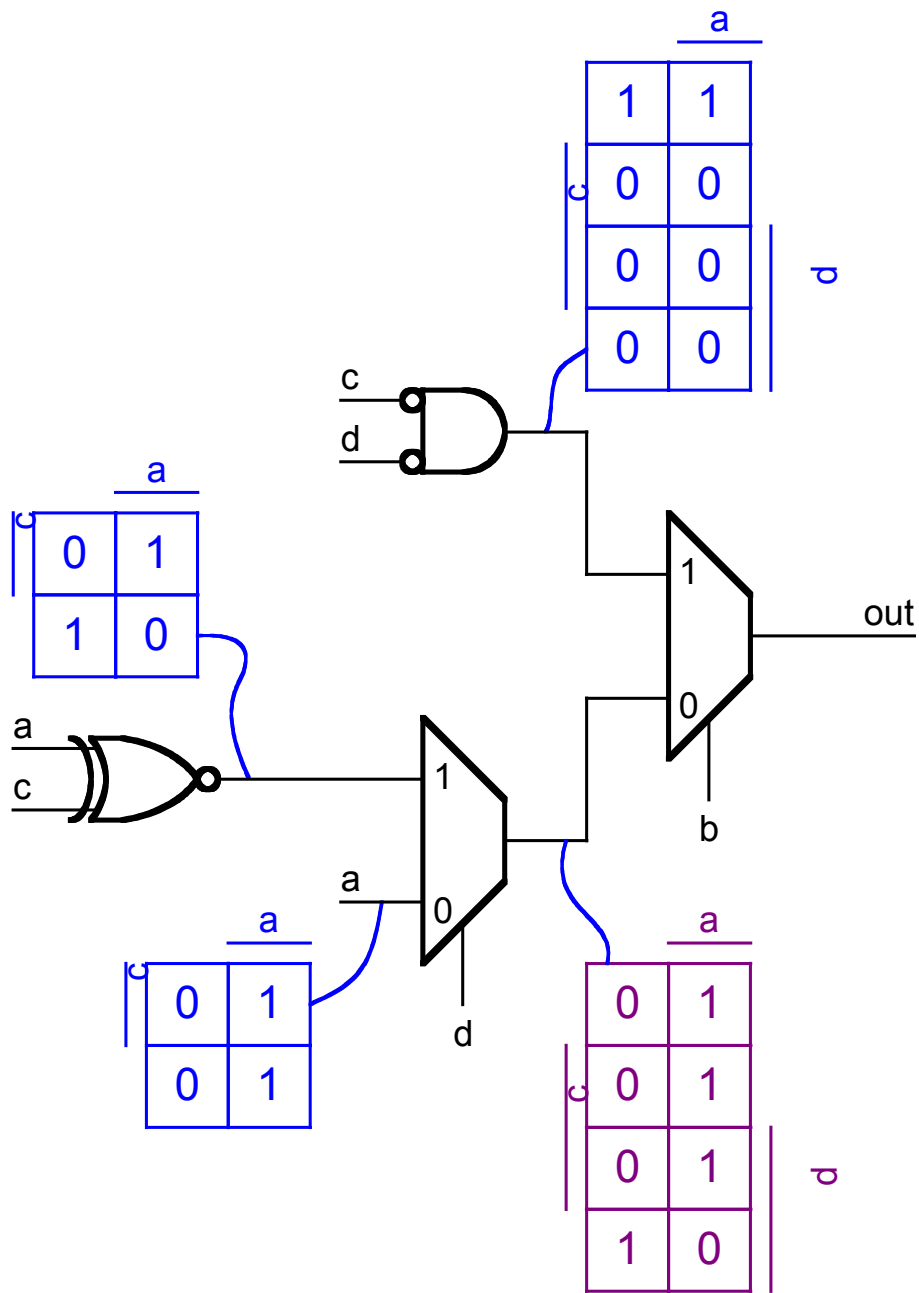


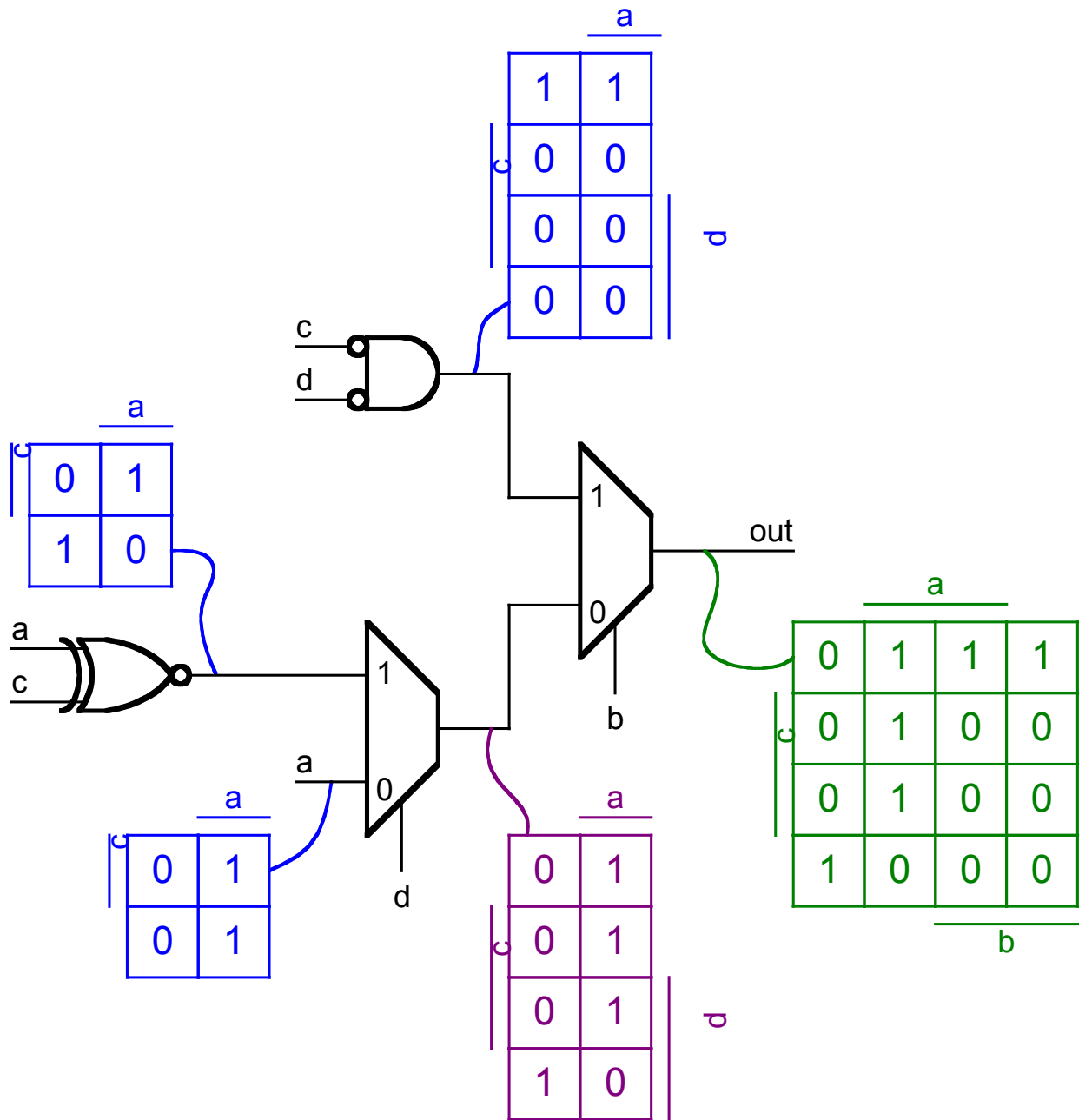
What function does this realize?



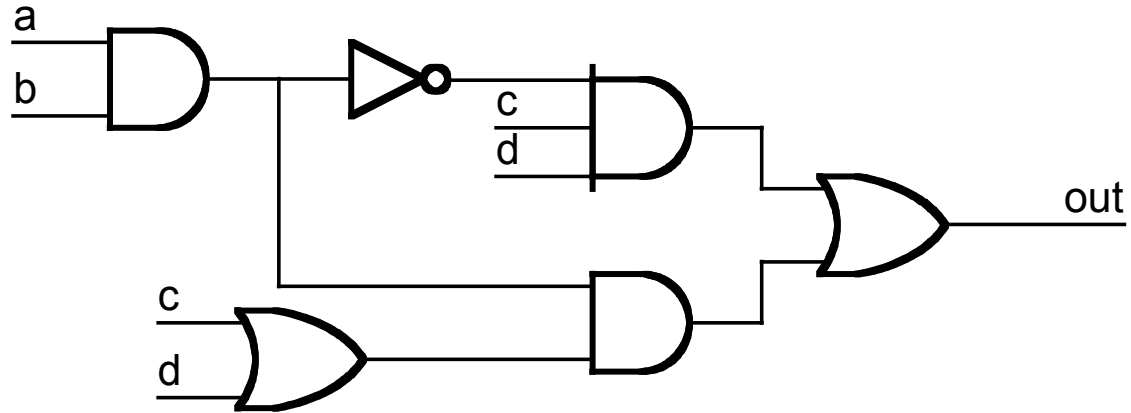
Starting at input, write equations
(or k-maps) for signals

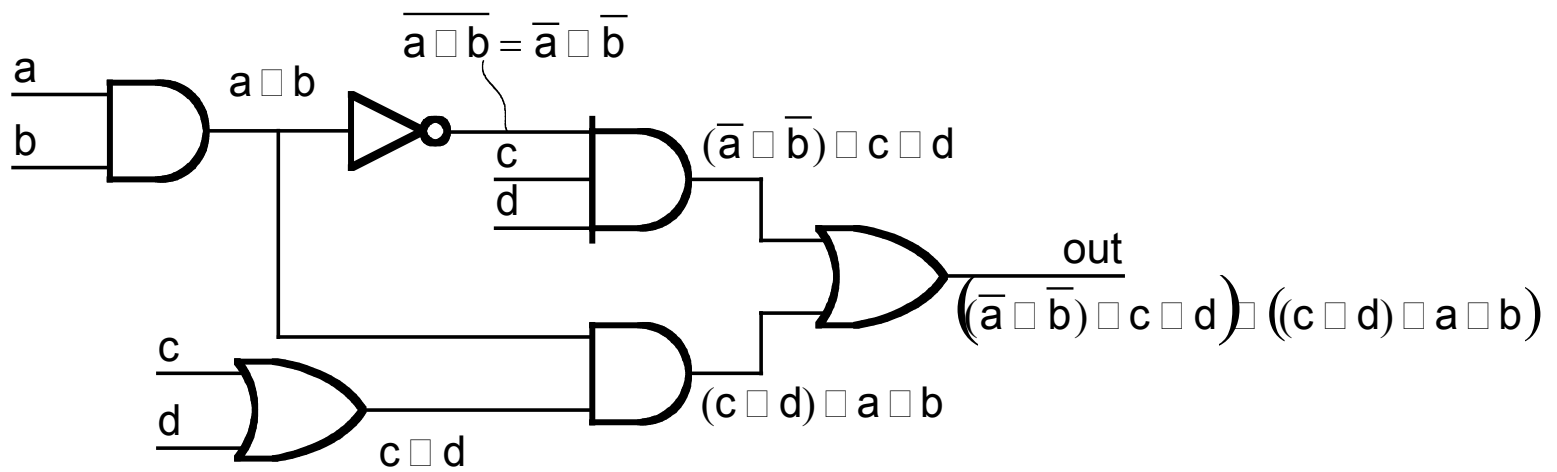


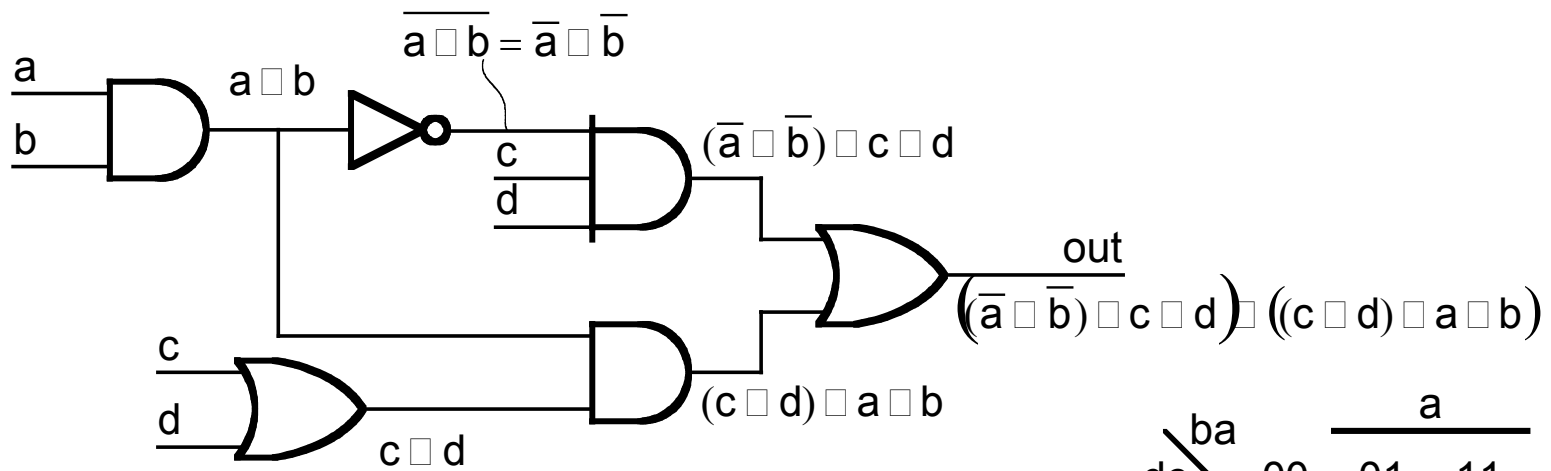




What function does this realize?





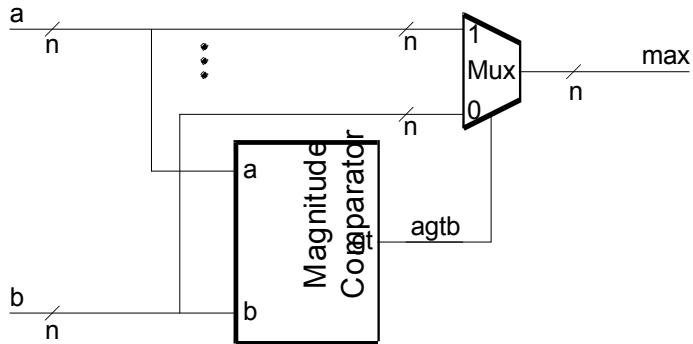


		a			
		00	01	11	10
dc	00	0 ₀	0 ₁	0 ₃	0 ₂
	01	0 ₄	0 ₅	1 ₇	0 ₆
	11	1 ₁₂	1 ₁₃	1 ₁₅	1 ₁₄
	10	0 ₈	0 ₉	1 ₁₁	0 ₁₀
c		b			

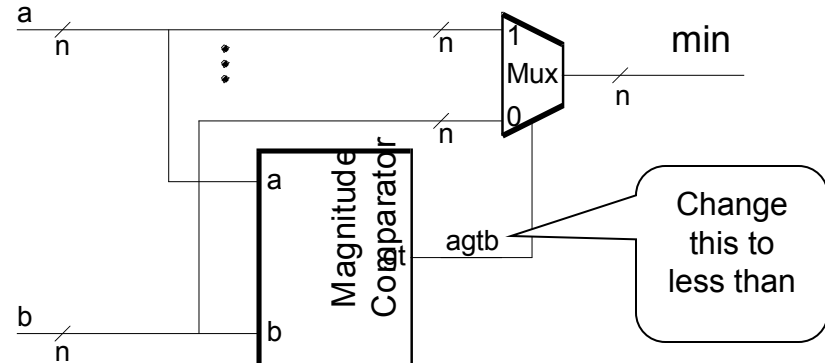
Building Block Combinational Logic

- Raise the abstraction layer of design using building blocks rather than directly with K-maps
 - Assemble combinational circuits from pre-defined building blocks
 - Decoder – converts codes (e.g., binary to one-hot)
 - Encoder – encodes one-hot to binary
 - Multiplexer – select an input (one-hot select)
 - Arbiter – pick first true bit
 - Comparators – equality and magnitude
 - ROMs
- Divide and conquer to build large units from small units
 - Decoder, encoder, multiplexer
- Logic with multiplexers or decoders (basis of LookUp Table in FGPAs)
- Bit-slice coding style

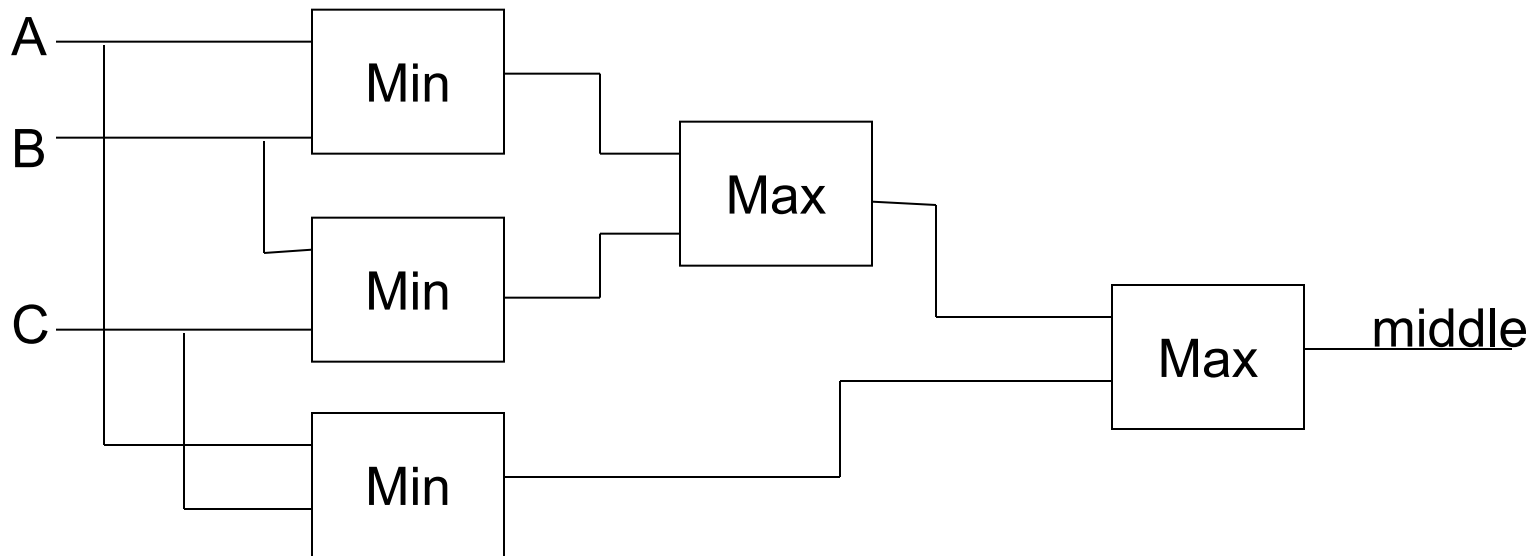
Maximum unit



How to change this to a minimum unit?



How to determine the middle of three numbers?



Example building block problems

- Circuit accepts 4 4-bit one-hot numbers, output the number with a one in the most significant position.

Circuit accepts 4 4-bit numbers (of any kind) output the number with the smallest number of 1s

Numbers and Arithmetic

- Binary numbers – bit i weighted 2^i
- 2's complement – MSB weighted -2^i
 - $-x$ represented by $2^{n-1} - x \pmod{2^{n-1}}$
 - makes subtraction easy
 - To negate, invert and add one
 - The add one can use the carry in of the least significant bit
- Multiplication accomplished by adding shifted versions of the multiplier
 - The shift is an implicit multiply by a power of 2
 - Example, 7×5

$$\begin{array}{r} 111 \\ x 101 \\ \hline 111 \\ 000 \\ 111 \\ \hline 100011 \end{array}$$

- Number representations do not have infinite ability to express all possible values
- accuracy and resolution help quantify the error
- Fixed versus floating point

Examples

- Design a circuit that computes $-2a$ given input a
 - Shift a left 1-bit, ($2x$) and negate – complement and add 1
- Convert 35.6 to 8.3 floating point
 - First convert to binary 100101.10011...
 - Then take 8 MSBs as mantissa .10010110
 - Need to shift binary point six places .10010110e110
 - With an implied leading 1 1.00101100e101

Next Lectures – Sequential Logic