IOWA STATE UNIVERSITY
Department of Electrical and Computer Engineering

Iowa State Honors Poster Presentation
May 1st 2024

Jonathan Hess
Advised by Alexander Stoytchev

https://github.com/Jetsama/DigitalLogicProject

Digital Logic Lab Tool: Hands on Learning

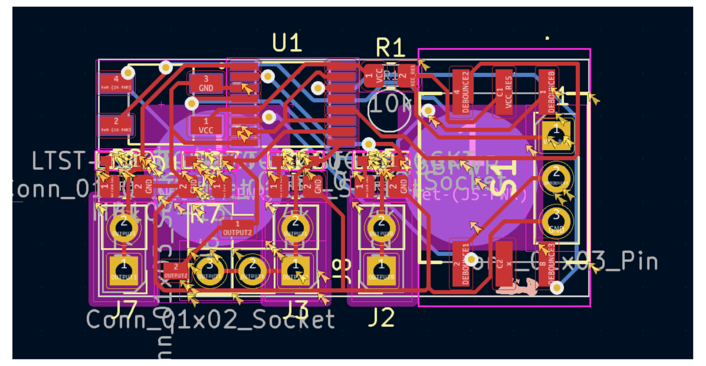
Introduction:
The project takes the abstract logic gates from computer engineering and creates physical representations that can be played and experimented with. This can make the learning of digital logic more fun and intuitive! Because of the nature of digital logic, students of all different knowledge levels can use these basic gates to create more interesting circuits.

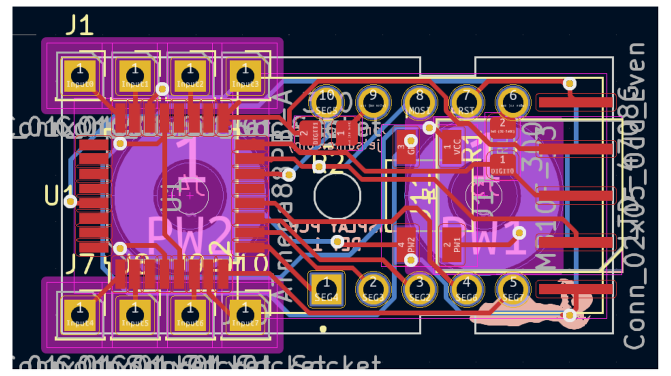
Background:
- Digital logic is the basis for all computer and digital systems
- Digital logic and the math that surrounds it could be understood by students in middle/high school
- Digital logic builds upon itself to create complexity by combing simple gates in more interesting ways
- Previous physical examples require power and wire management

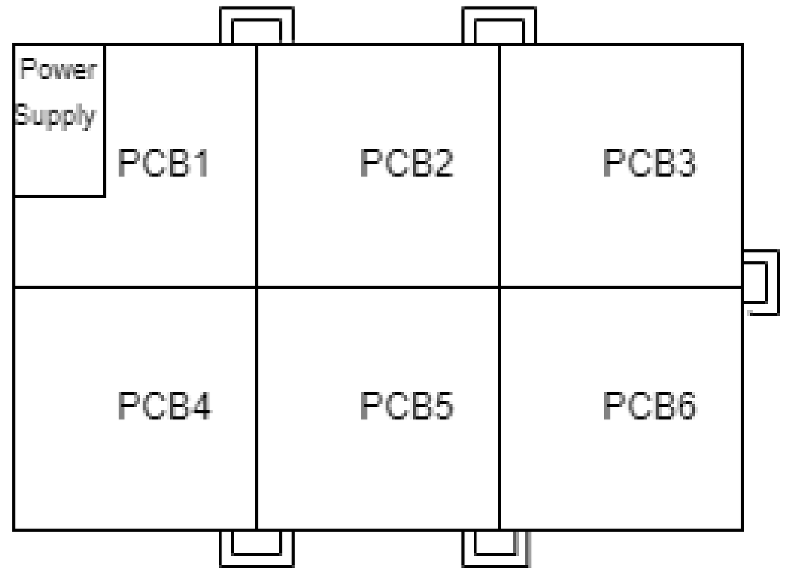
Each gate has the following:
- Lights showing the state of all Inputs and Outputs
- A 3D printed case that resembles the symbol used in circuits
- Pins for Inputs and Sockets for Outputs
- 2 Buttons that supply power to the PCB

Design:
- The logic gate's design needed to be small enough to allow for complex circuits but big enough to be handled easily.
- There is 10 logic gates of each type to allow for building of larger circuits.
- The board is expandable both to save on cost and for scalability.
- One PCB is used for all the basic logic gates by swapping the logic IC.

Goals for Project
- Use only open-source tools such as KiCad to allow teachers/students to edit the design in the future.
- Reduce costs to maximize the number of gates per kit.
- Make source code, PCBs, and BOM available for anyone to create, edit, or manufacture themselves.
- Create a booklet that teaches using puzzles and fun challenges.

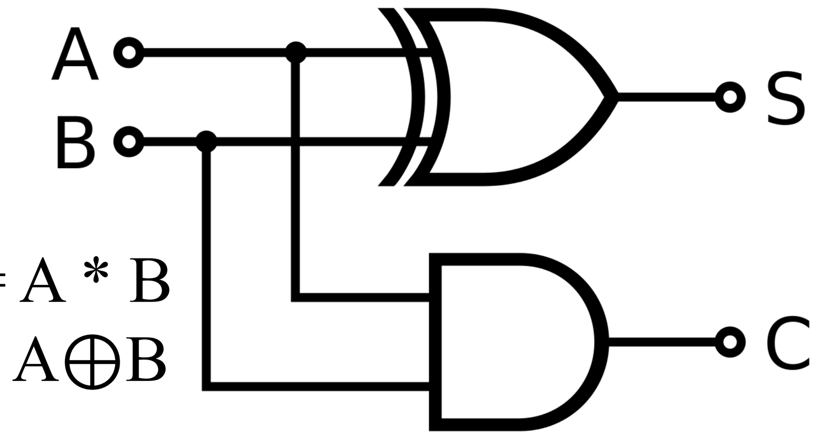

Rev 1 Inputs PCB


Rev 1 Outputs PCB



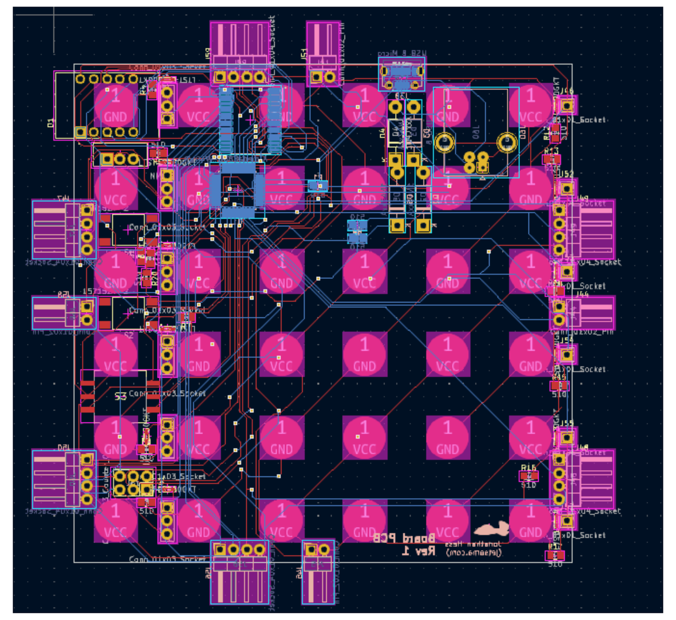
Scalability via Multiple Board PCB Layout

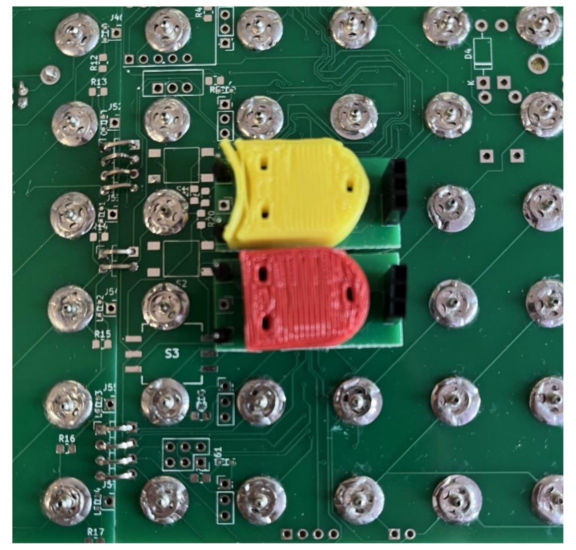
Half Adder Circuit

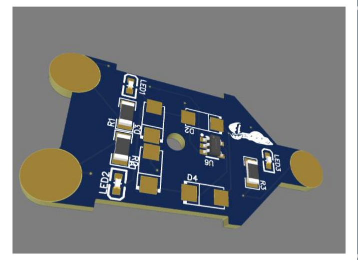


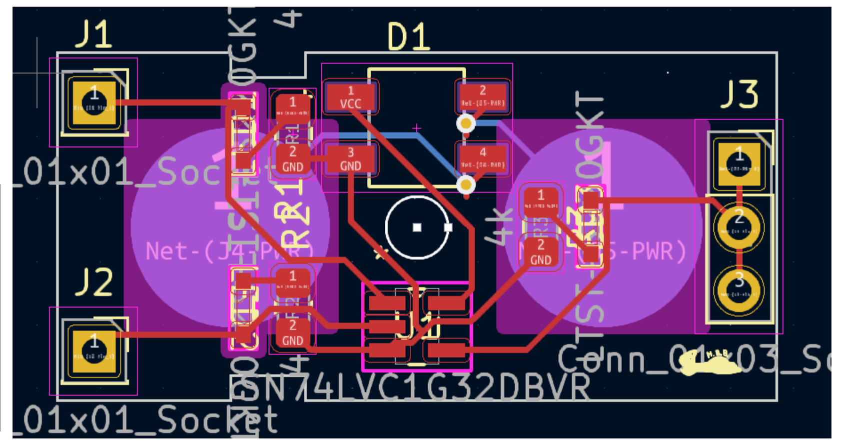
C = A * B
S = A⊕B

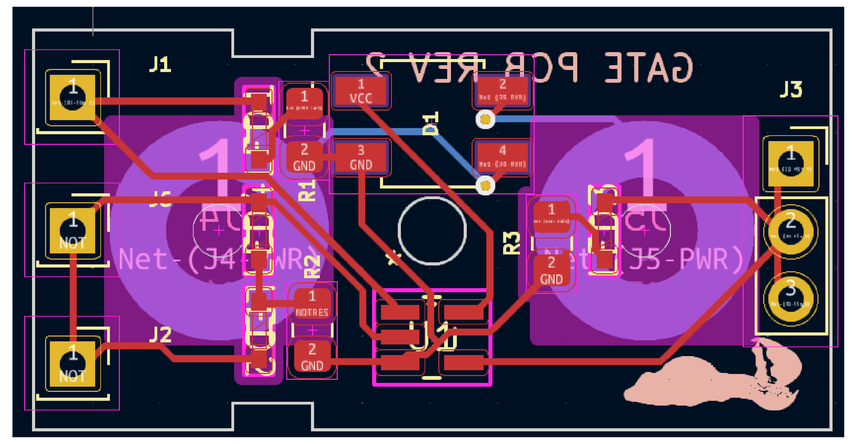
A	B	C	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0
		AND	XOR

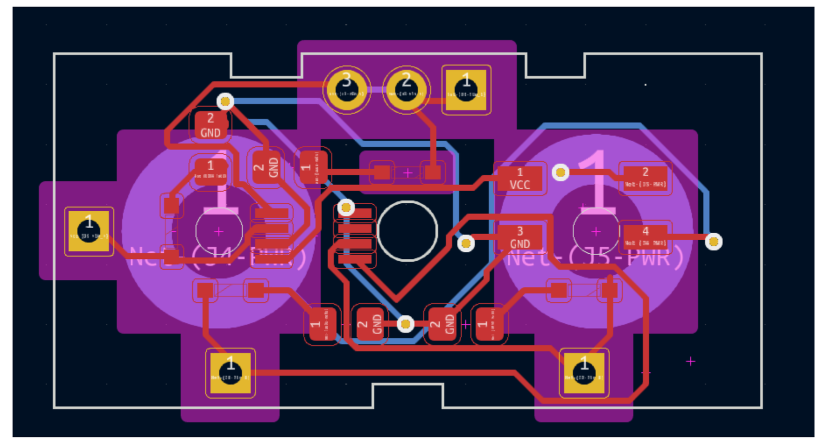

Rev 1 Board PCB


Board with XOR and AND Gates


Original PCB Design

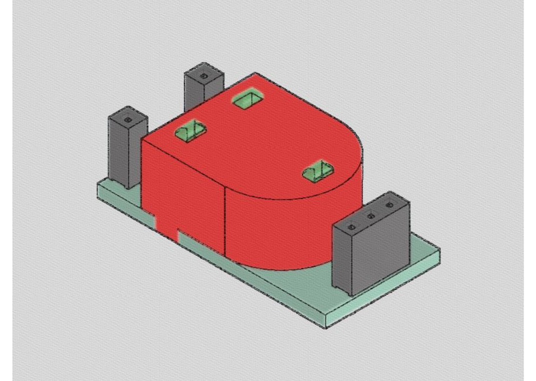

Rev 1 Gate PCB


Rev 2 Gate PCB

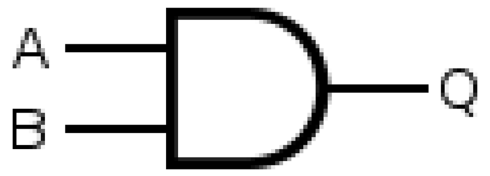

Rev 1 Mux PCB

Final Logic Gates Design:

AND

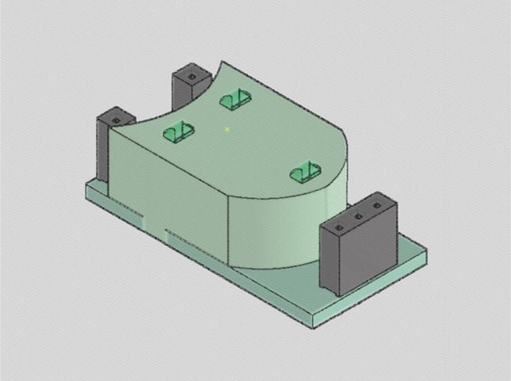


A	B	Q
0	0	0
0	1	0
1	0	0
1	1	1



Q = A * B

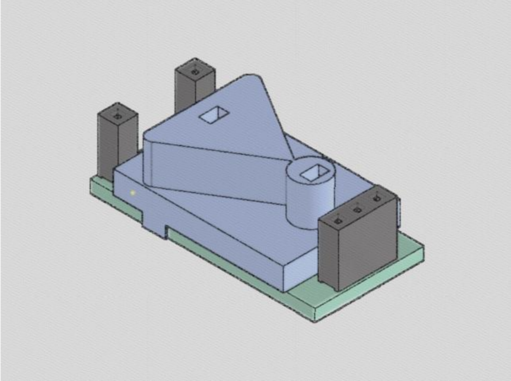
OR



A	B	Q
0	0	0
0	1	1
1	0	1
1	1	1

Q = A + B

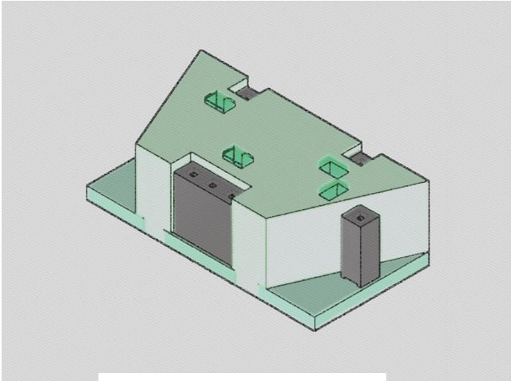
NOT



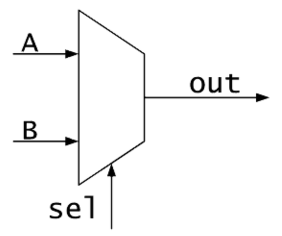
A	Q
0	1
1	0

$Q = \bar{A}$

MULTIPLEXER

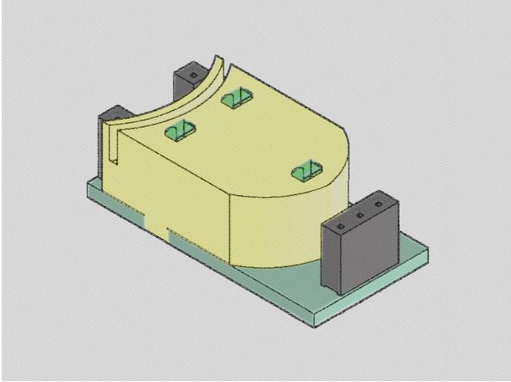


S	Q
0	A
1	B

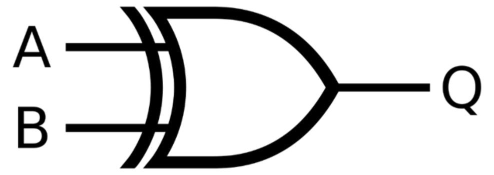


$Q = \overline{S}*A + S*B$

XOR

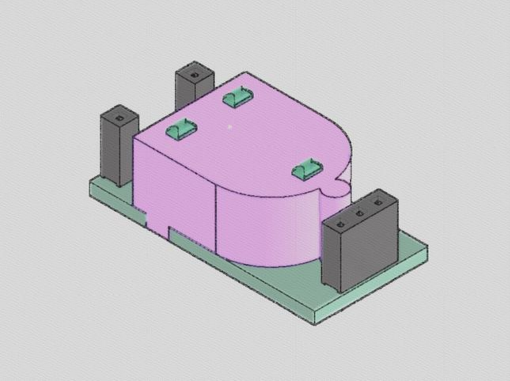


A	B	Q
0	0	0
0	1	1
1	0	1
1	1	0



Q = A ⊕ B

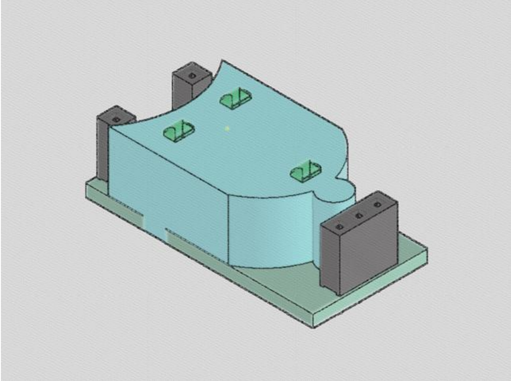
NAND



A	B	Q
0	0	1
0	1	1
1	0	1
1	1	0

$Q = \overline{(A*B)}$

NOR



A	B	Q
0	0	1
0	1	0
1	0	0
1	1	0

$Q = \overline{(A+B)}$

Special thanks to the ETG for all the help and consultation!