Bitwise Operators

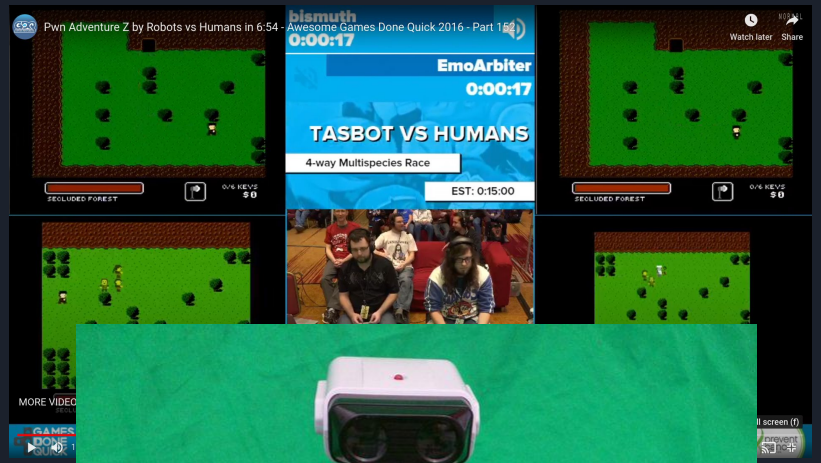
Level 0x03: XOR

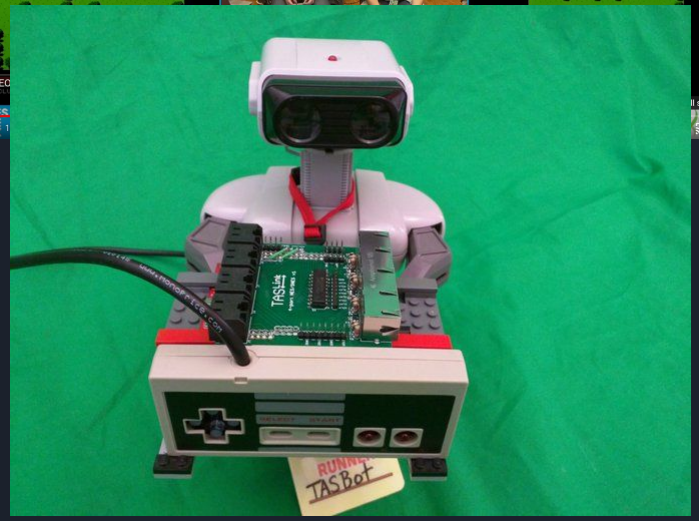
Quick Overview

- Fun Stuff
- Bitwise Operators

Cybersecurity and Video Game Speedrunning

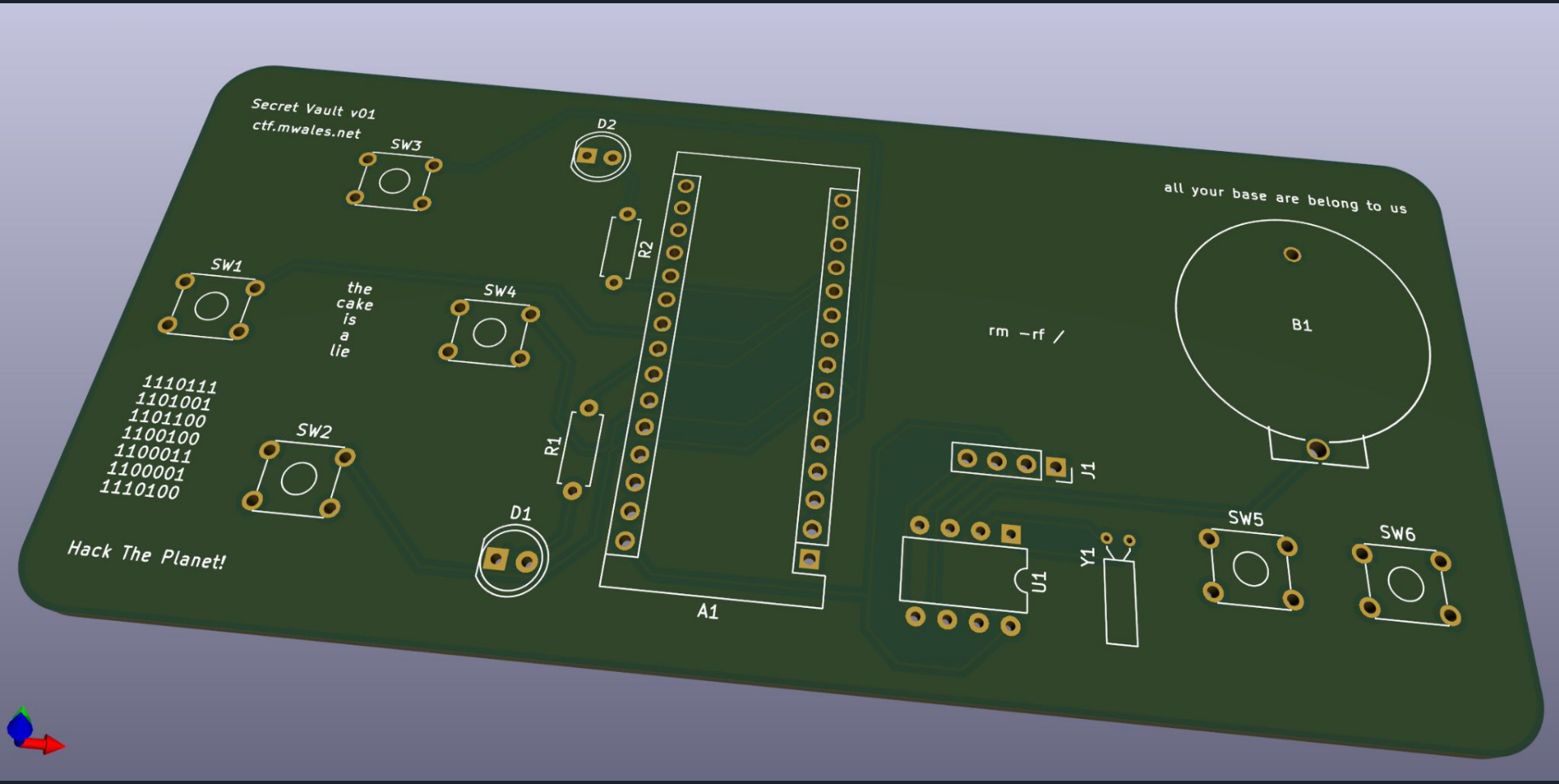
- West Shore Career Fair Expo
- Presenters
  - Jordan Wiens
  - Michael Wales
- Discussing the overlap of video game speedrunning and cyber security research
- Talk about our work in cyber security





Jeri Ellsworth - Maker / Hacker

- Commodore 64 Kid
- HAM radio (built her own walkie talkies, blew way past FCC limits…)
- Race car driver / chassis builder / rule ~~breaker~~ extender
- Entrepreneur (made her own computer store / chain)
- Made her own transistors
- Made her own 1-chip custom chip C64 in a controller in 1 year
- Worked at Valve (VR / AR / everything research)
- Commodore 64 bass guitar
- Tilt Five - AR Board Game


Secret Vault v01
ctf.mwales.net
SW3
D2
R2
SW1
the
cake
is
a
lie
SW4
1110111
1101001
1101100
1100100
1100011
1100001
1110100
SW2
R1
D1
Hack The Planet!
A1
all your base are belong to us
rm -rf /
B1
J1
U1
Y1
SW5
SW6

AND

2 - input AND gate

A	B	Output
0	0	0
0	1	0
1	0	0
1	1	1

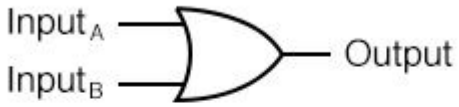
Bitwise AND & [edit]

bit a	bit b	a & b (a AND b)
0	0	0
0	1	0
1	0	0
1	1	1

```
  11001000
& 10111000
  --------
= 10001000
```

OR

2 - input OR gate



A	B	Output
0	0	0
0	1	1
1	0	1
1	1	1

Bitwise OR | [edit]

bit a	bit b	a | b (a OR b)
0	0	0
0	1	1
1	0	1
1	1	1

```
  11001000
| 10111000
  --------
= 11111000
```

XOR - Exclusive OR

Exclusive-OR gate

A	B	Output
0	0	0
0	1	1
1	0	1
1	1	0

Bitwise XOR ^ [edit]

bit a	bit b	a ^ b (a XOR b)
0	0	0
0	1	1
1	0	1
1	1	0

```
  11001000
^ 10111000
  --------
= 01110000
```

Fun with XOR

$$A \oplus 0 = A,$$

$$A \oplus A = 0,$$

$$A \oplus B = B \oplus A,$$

$$(A \oplus B) \oplus C = A \oplus (B \oplus C),$$

$$(B \oplus A) \oplus A = B \oplus 0 = B,$$

Simple XOR Encryption

Example [edit]

The string "Wiki" (`01010111 01101001 01101011 01101001` in 8-bit ASCII) as follows:

```
  01010111 01101001 01101011 01101001
⊕ 11110011 11110011 11110011 11110011
  ----------------------------------
= 10100100 10011010 10011000 10011010
```

And conversely, for decryption:

```
  10100100 10011010 10011000 10011010
⊕ 11110011 11110011 11110011 11110011
  ----------------------------------
= 01010111 01101001 01101011 01101001
```

Compilation Optimization

```
// C function, returns 0

int square()

{

        int a = 0;

        return a;

}
```

x86-64 gcc 8.4 (Editor #1)

x86-64 gcc 8.4 | -m32 -O1

```
        square():
        b8 00 00 00 00
0       mov     eax,0x0
        c3
5       ret
```

x86-64 gcc 8.4 (Editor #1)

x86-64 gcc 8.4 | -m32 -O2

```
        square():
        31 c0
0       xor     eax,eax
        c3
2       ret
```

Attributions

- https://computerengineeringforbabies.com/blogs/engineering/xor-gate
- https://www.allaboutcircuits.com/textbook/digital/chpt-3/multiple-input-gates/
- https://en.wikipedia.org/wiki/Bitwise_operations_in_C
- https://godbolt.org/