TOP SECRET

WITCHCRAFT SECRETS

CONFIDENTIAL
Witchcraft Secrets

...from a reverse-engineer

Alyssa Rosenzweig
Starswirl’s First Law

Magic can neither be created nor destroyed...
Starswirl’s First Law

Magic can neither be created nor destroyed…

….only transformed.
Starswirl’s First Law
Starswirl’s First Law
Starswirl’s First Law
Witchcraft
OPEN SORCERY
STEPS
Steps - trivial

- Write a test
- Trace baseline input
- Trace with single “interesting” change
- Diff the traces
- Single change? Lucky.
Spelling convention

\[ f(x) = y \]
Steps - nontrivial

- Write a test
- Trace many inputs
- Record results
- Find a pattern
- Deduce \( f \)
- Rewrite for \( f^{-1} \)
Steps - nontrivial

- Write a test
- **Trace many inputs**
- Record results
- Find a pattern
- Deduce $f$
- Rewrite for $f^{-1}$
TRACING
Steps - nontrivial

- Write a test
- Trace many inputs
- Record results
- **Find a pattern**
- **Deduce** \( f \)
- Rewrite for \( f^{-1} \)
TECHNIQUES
Waiting

- Cuss on IRC.
- Wait for a draconequus to whisper $f$ to you.

*Works 20% of the time*
Law of Parsimony

- Simple $f$ are more likely than complex $f$.
- Simple for the \textit{hardware}, not for you!
- Think like a hardware designer (gate count).
Properties

- Alignment?
- Monotonicity?
- Linear? *Almost* linear?
- Bitwise complements?
- Powers of two?
Information entropy

- Input entropy vs output bit count
- Equal entropy: just shuffling
- More in input: incomplete encoding (pigeonhole)
- More in output: incomplete input (Occam's Razor)
Graphs
Calculus

- Discrete derivatives (backwards differencing)
- Sometimes modeling $f'$ is easier than $f$.
- Integrate $f'$ to recover $f$ (summation)

*Useful for near-linear $f*
Classes of $f$

- What kind of function could satisfy the properties?
- Closed-form algebraic?
- Bitwise manipulation?
- Try some.
Purpose

• Every field has a reason for being.
• Your job: figure out why.
Purpose

```c
struct texture {
    unsigned width;
    unsigned unknown;
    unsigned depth;
...
```
Purpose

struct texture {
    unsigned width;
    unsigned height;
    unsigned depth;
    ...

Purpose

Proximate fields have proximate purposes.
Info drops

- Know the hardware, know the purpose.
- Conference slides.
- Vendor blogs.
- Code drops (kernel)
- Google is your friend.
XDC Hallway Track

[redacted]
If all else fails...

- Move on.
- Lots of seaponies in the sea.
EXAMPLE
if (vColor.x < 0.5) discard;

flt r31.w, r0.x, #0.5
br.discard.true
2 - AND

if (vColor.x < 0.5 && vColor.y < 0.75)
discard;

flt r31.w, r0.y, #0.75
flt r31.w, r0.x, #0.5
brx.discard.unk8888
if (vColor.x < 0.5 || vColor.y < 0.75) discard;

flt r31.w, r0.y, #0.75
flt r31.w, r0.x, #0.5
brx.discard.unkEEEE
if (!(vColor.x < 0.5 && vColor.y < 0.75))
discard;

flt r31.w, r0.y, #0.75
flt r31.w, r0.x, #0.5
brx.discard.unk1111
if (!(vColor.x < 0.5 || vColor.y < 0.75))
discard;

flt r31.w, r0.y, #0.75
flt r31.w, r0.x, #0.5
brx.discard.unk7777
<table>
<thead>
<tr>
<th>Expression</th>
<th>Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>(A &amp;&amp; B)</td>
<td>8888</td>
</tr>
<tr>
<td>(A</td>
<td></td>
</tr>
<tr>
<td>!(A &amp;&amp; B)</td>
<td>7777</td>
</tr>
<tr>
<td>!(A</td>
<td></td>
</tr>
</tbody>
</table>
if (vColor.x < 0.5 && vColor.y < 0.75 && vColor.z == 1.0) discard;

flt r31.x, r0.y, #0.75
feq r31.w, r0.z, #1
flt r31.w, r0.x, #0.5
brx.discard.unk8080
if (vColor.x < 0.5 && vColor.y < 0.75 && vColor.z == 1.0 && vColor.w == 0.0) discard;

feq r31.w, r0.z, #1
feq r31.w, r0.w, #0
flt r31.x, r0.y, #0.75
flt r31.w, r0.x, #0.5
brx.discard.unk8000
if (vColor.x < 0.5 && vColor.y < 0.75 && vColor.z == 1.0 && vColor.w == 0.0 && vColor.x > vColor.y) discard;

...  
iand r31.w, r0.z, r0.w  
brx.discard.unk8000
<table>
<thead>
<tr>
<th>Expression</th>
<th>Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>(A &amp;&amp; B)</td>
<td>8888</td>
</tr>
<tr>
<td>(A</td>
<td></td>
</tr>
<tr>
<td>!(A &amp;&amp; B)</td>
<td>7777</td>
</tr>
<tr>
<td>!(A</td>
<td></td>
</tr>
<tr>
<td>(A &amp;&amp; B) &amp;&amp; C</td>
<td>8080</td>
</tr>
<tr>
<td>(A &amp;&amp; B) &amp;&amp; C &amp;&amp; D</td>
<td>8000</td>
</tr>
</tbody>
</table>
Mathemagics

\[ f(x_1, x_2, x_3, x_4) = y \]
<table>
<thead>
<tr>
<th>Expression</th>
<th>Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>(A &amp;&amp; B)</td>
<td>8888</td>
</tr>
<tr>
<td>(A</td>
<td></td>
</tr>
<tr>
<td>!(A &amp;&amp; B)</td>
<td>7777</td>
</tr>
<tr>
<td>!(A</td>
<td></td>
</tr>
<tr>
<td>(A &amp;&amp; B) &amp;&amp; C</td>
<td>8080</td>
</tr>
<tr>
<td>(A &amp;&amp; B) &amp;&amp; C &amp;&amp; D</td>
<td>8000</td>
</tr>
<tr>
<td>Expression</td>
<td>Code</td>
</tr>
<tr>
<td>-----------------</td>
<td>------</td>
</tr>
<tr>
<td>(A &amp;&amp; B)</td>
<td>8888</td>
</tr>
<tr>
<td>!(A &amp;&amp; B)</td>
<td>7777</td>
</tr>
<tr>
<td>(A</td>
<td></td>
</tr>
<tr>
<td>!(A</td>
<td></td>
</tr>
<tr>
<td>Expression</td>
<td>Code</td>
</tr>
<tr>
<td>------------</td>
<td>------</td>
</tr>
<tr>
<td>(A &amp;&amp; B)</td>
<td>8888</td>
</tr>
<tr>
<td>!(A &amp;&amp; B)</td>
<td>7777</td>
</tr>
<tr>
<td>(A</td>
<td></td>
</tr>
<tr>
<td>!(A</td>
<td></td>
</tr>
<tr>
<td>Expression</td>
<td>Code</td>
</tr>
<tr>
<td>------------</td>
<td>------</td>
</tr>
<tr>
<td>((A &amp;&amp; B) &amp;&amp; C &amp;&amp; D)</td>
<td>8000</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Expression</td>
<td>Code</td>
</tr>
<tr>
<td>----------------------------</td>
<td>------</td>
</tr>
<tr>
<td>(A &amp;&amp; B)</td>
<td>8888</td>
</tr>
<tr>
<td>(A</td>
<td></td>
</tr>
<tr>
<td>!(A &amp;&amp; B)</td>
<td>7777</td>
</tr>
<tr>
<td>!(A</td>
<td></td>
</tr>
<tr>
<td>(A &amp;&amp; B) &amp;&amp; C</td>
<td>8080</td>
</tr>
<tr>
<td>(A &amp;&amp; B) &amp;&amp; C &amp;&amp; D</td>
<td>8000</td>
</tr>
</tbody>
</table>
LUT
Thank you!