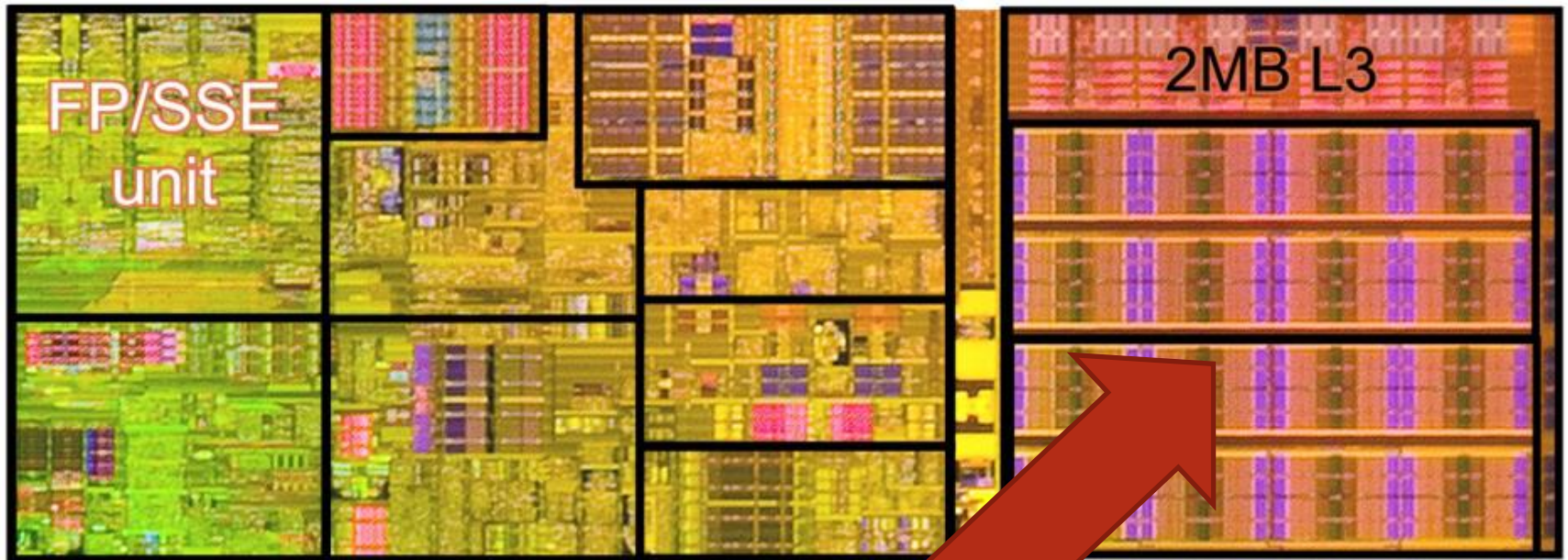


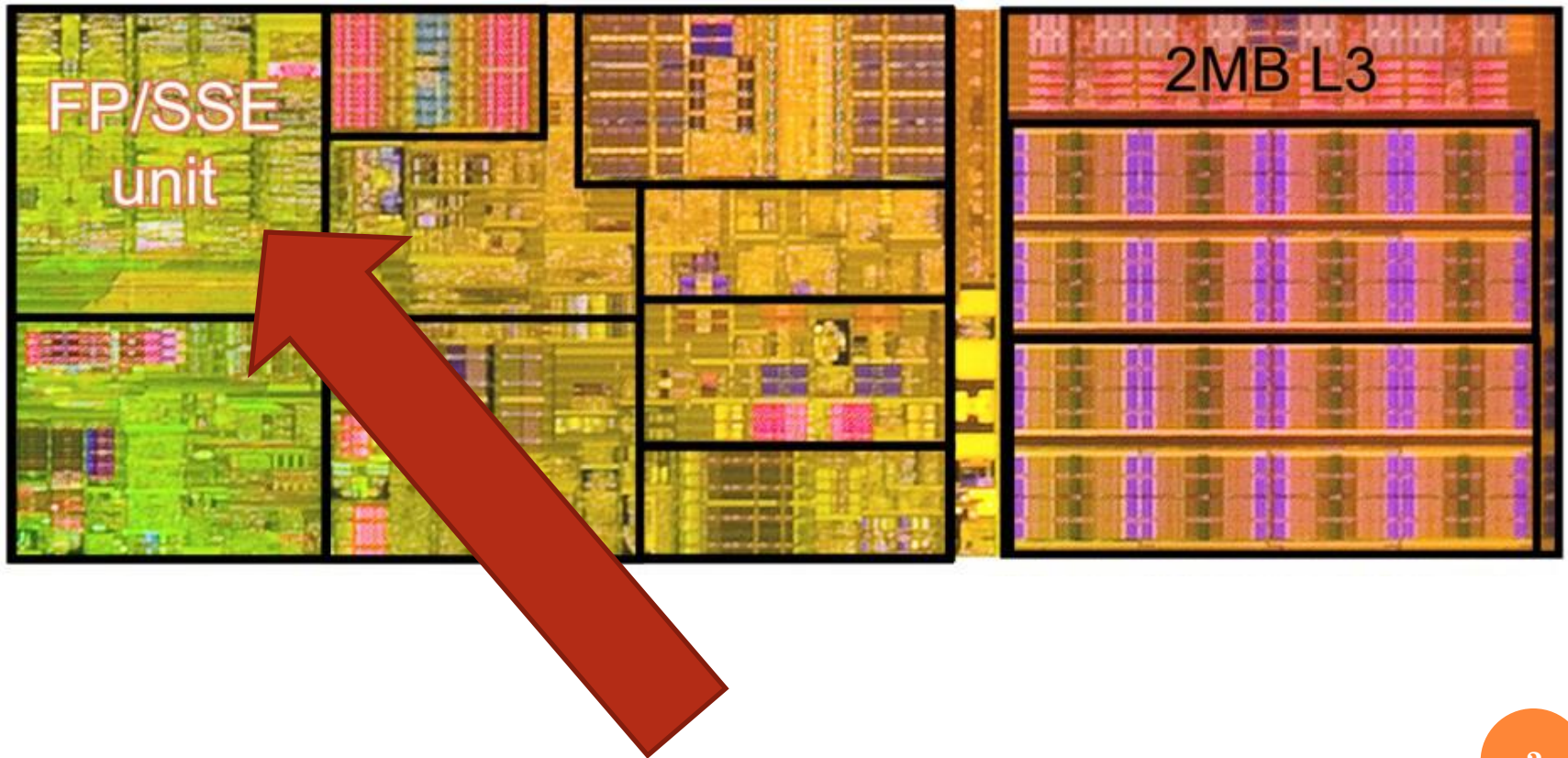


ON SUBNORMAL FLOATING POINT AND ABNORMAL TIMING

Marc Andryscio, **David Kohlbrenner**, Keaton Mowery,
Ranjit Jhala, Sorin Lerner, and Hovav Shacham

UC San Diego





LETS RUN SOME CODE

Normal Floating Point

```
#include <stdio.h>
#include <stdint.h>

void main(int argc, char* argv[]){

    double x = 1.0;
    double z,y = 1.0;
    uint32_t i;
    for(i=0; i<1000000000; i++){
        z = y*x;
    }
}
```

Subnormal Floating Point

```
#include <stdio.h>
#include <stdint.h>

void main(int argc, char* argv[]){

    double x = 1.0e-323;
    double z,y = 1.0;
    uint32_t i;
    for(i=0; i<1000000000; i++){
        z = y*x;
    }
}
```

LETS RUN SOME CODE

Normal Floating Point

```
#include <stdio.h>
#include <stdint.h>

void main(int argc, char* argv[]){

    double x = 1.0;
    double z,y = 1.0;
    uint32_t i;
    for(i=0; i<1000000000; i++){
        z = y*x;
    }
}
```

0.204s

Subnormal Floating Point

```
#include <stdio.h>
#include <stdint.h>

void main(int argc, char* argv[]){

    double x = 1.0e-323;
    double z,y = 1.0;
    uint32_t i;
    for(i=0; i<1000000000; i++){
        z = y*x;
    }
}
```

4.332s

20 TIMES SLOWER?

- Who knew?
 - Numerical analysts
 - CPU designers
 - Game engine authors

20 TIMES SLOWER?

- Who knew?
 - Numerical analysts
 - CPU designers
 - Game engine authors
- Who should know?
 - “What Every Computer Scientist Should Know About Floating-Point Arithmetic” – Goldberg '91

20 TIMES SLOWER?

- Who knew?
 - Numerical analysts
 - CPU designers
 - Game engine authors
- Who should know?
 - “What Every Computer Scientist Should Know About Floating-Point Arithmetic” – Goldberg ’91
- Academic researchers claim to “effectively close[s] all known remotely exploitable channels”
 - Specifically referring to timing side channels!

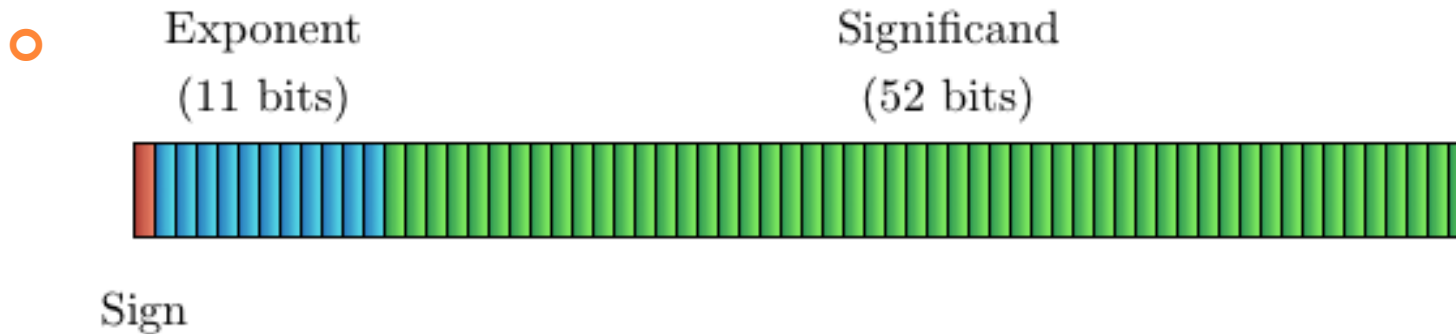


FLOATING POINT AND TIMING

WHAT HAPPENED?

- IEEE 754 specifies *subnormal* floating point values

FLOATING POINT NORMAL AND SUBNORMAL



- Value = $(-1)^{\text{sign}} * \text{significand} * 2^{(\text{exponent} - \text{bias})}$
 - The **exponent** is non-zero
 - Normal values have an *implicit* leading 1-bit on the **significand**
- A *subnormal* value is a special encoding
 - The **exponent** is all zeroes
 - The **significand** has an *implicit* leading 0-bit

SUBNORMAL DETAILS

- Subnormal ranges (double)
 - Minimum: $\sim 4.9 \times 10^{-324}$
 - Maximum: $\sim 2.23 \times 10^{-308}$
 - Planck length: $1.6 \times 10^{-35} \text{ m}$
- Why?
 - Extend the range of floating point
 - Graceful underflow

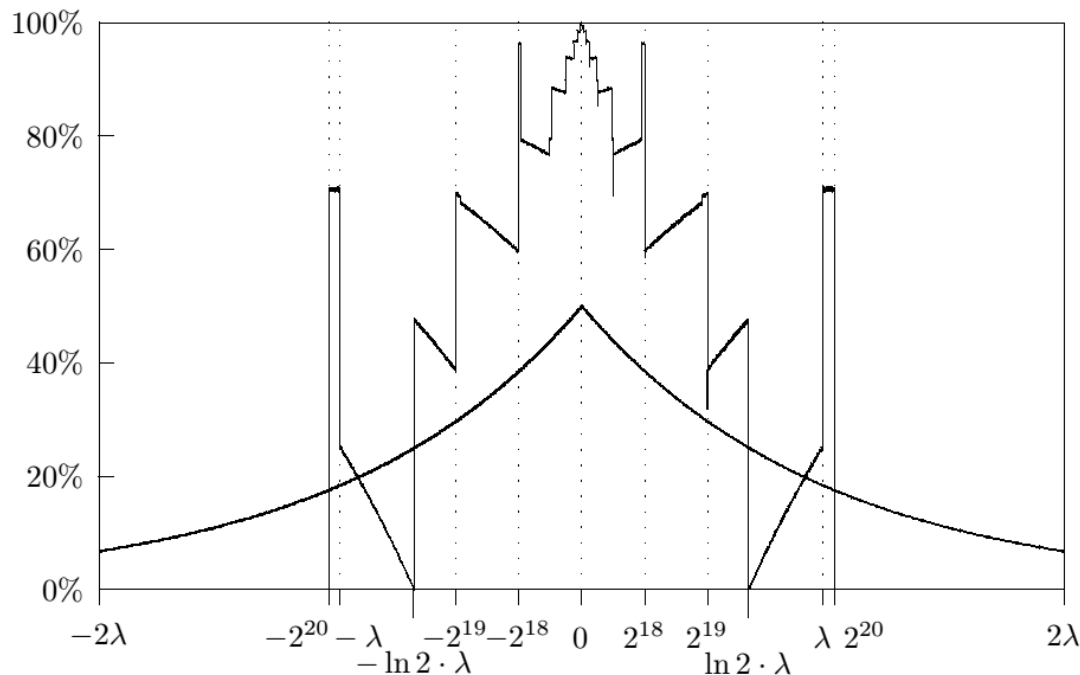
```
if(a != b)
    x = c / (a-b);
```

WHAT HAPPENED?

- IEEE 754 specifies *subnormal* floating point values
- FPUs are optimized for pure speed
 - Subnormals are *not* the common case
 - So let's pretend they don't matter!
- Subnormals are a hardware slowpath
 - The Alpha trapped to kernel for subnormals!
 - Most GPUs don't support them

FLOATING POINT IS A SECURITY ISSUE

- Ilya Mironov on Laplacian noise generation



- Lack of dependable results
 - gcc -O1 vs gcc -O3

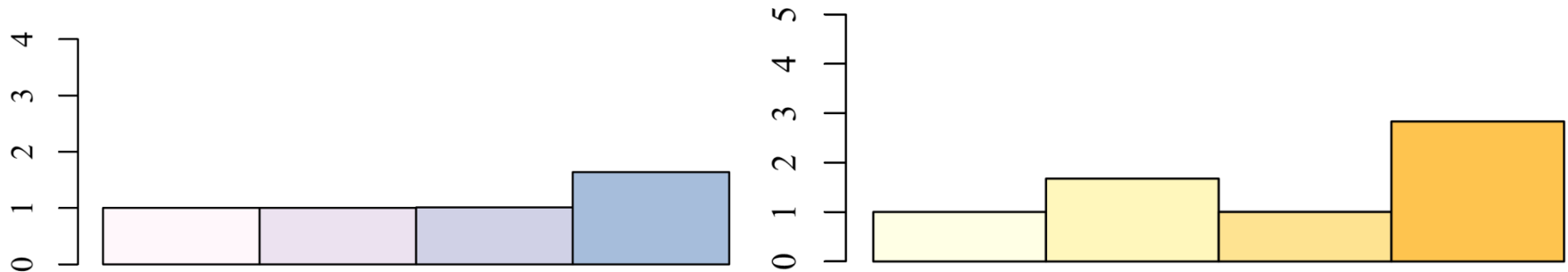


LEVERAGING SUBNORMAL FLOATING POINT INTO ATTACKS

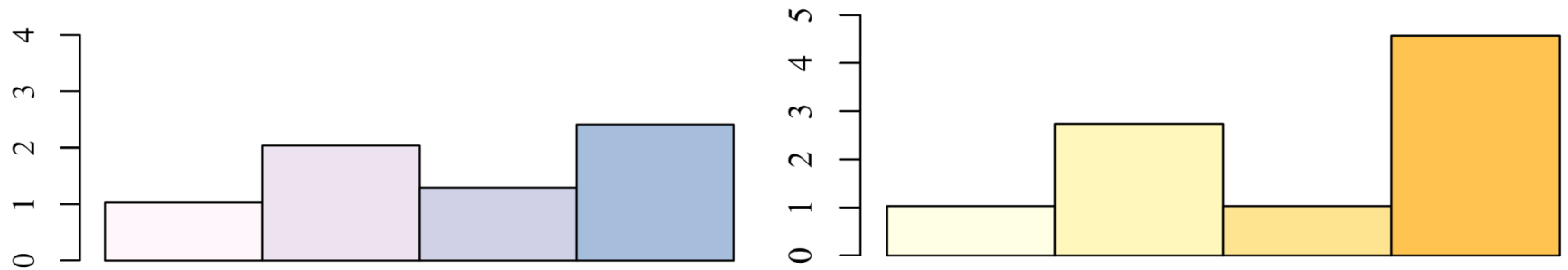
FLOATING POINT AS A SIDE-CHANNEL

- Code that operates on *secret* and *attacker* values can result in timing side channels
 - From instruction traces
 - Or memory access patterns
 - Or IO usage
 - Etc.
- We present the first *instruction data* based timing side channel attack on a commodity desktop processor
 - Proposed by Kocher 20 years ago!

FLOATING POINT HARDWARE DATA



Core i7-3667U SSE and x87



Atom D2550 SSE and x87

addsd	normal,normal
addsd	normal,subnormal
divsd	normal,normal
divsd	normal,subnormal

fadd	normal,normal
fadd	normal,subnormal
fmul	normal,normal
fmul	normal,subnormal

AMPLIFYING TIMING DIFFERENCES

- Even a 100 cycle difference is hard to spot
 - Especially with a loaded system

```
#include <stdio.h>
#include <stdint.h>

void main(int argc, char* argv[]){

    double x = 1.0e-323;
    double z,y = 1.0;
    uint32_t i;
    for(i=0; i<1000000000; i++){
        z = y*x;
    }
}
```

AMPLIFYING TIMING DIFFERENCES

- Even a 100 cycle difference is hard to spot
 - Especially with a loaded system
- We need an *amplifier*
- Remember our sample code?
 - We need tight math loops

```
#include <stdio.h>
#include <stdint.h>

void main(int argc, char* argv[]){

    double x = 1.0e-323;
    double z,y = 1.0;
    uint32_t i;
    for(i=0; i<1000000000; i++){
        z = y*x;
    }
}
```



DETOUR TIME!

Firefox SVG Filters and Previous Attacks

FIREFOX SVG FILTERS

○ Turn this



The screenshot shows the Wikipedia homepage in a Firefox browser window. The address bar displays the URL `csweb.ucsd.edu/~dkohlbre/noblur.html`. The page features the Wikipedia logo and the text "WIKIPEDIA The Free Encyclopedia". The left sidebar contains navigation links such as "Main page", "Contents", "Featured content", "Current events", "Random article", "Donate to Wikipedia", and "Wikipedia store". The main content area includes a "Welcome to Wikipedia" message, a "From today's featured article" section highlighting the film *Blackrock*, and an "In the news" section with a list of recent events, including a gun attack in Karachi and the sentencing of Hosni Mubarak. The bottom of the page shows language options and a "Print/export" section.

csweb.ucsd.edu/~dkohlbre/noblur.html

Create account Log in

Main Page Talk Read View source View history Search

Welcome to **Wikipedia**,
the free encyclopedia that anyone can edit.
4,868,887 articles in English

- Arts
- Biography
- Geography
- History
- Mathematics
- Science
- Society
- Technology
- All portals

From today's featured article

Blackrock is a 1997 Australian drama film directed by **Steven Vidler** and written by **Nick Enright**. In *Blackrock*, a fictional beachside working-class suburb, a young surfer witnesses his friends raping a girl. When she is found murdered the next day, he is torn between revealing what he saw and protecting his friends. Filming locations included **Stockton, New South Wales**, where a girl named **Leigh Leigh was murdered** in 1989. While the film was never marketed as the story of her death, many viewers incorrectly believed it to be a factual account of the crime. Her family objected to what they saw as a fictionalisation of her murder, and residents of Stockton opposed the decision to shoot scenes there. The film received generally positive critical reviews in Australia, where it was nominated for five **AACTA Awards** and won two **AWGIE Awards**, though it received mixed reviews elsewhere. Adapted from Enright's *play of the same name*, the film stars **Laurence Breuls**, **Simon Lyndon** and **Linda Cropper**, and features the first credited film performance of **Heath Ledger**. (**Full article...**)

In the news

- At least forty-five people are killed in a **gun attack on a bus** in Karachi, Pakistan.
- A **7.3-magnitude aftershock** of the recent earthquake in Nepal kills more than seventy people and injures more than two thousand.
- Former Egyptian President **Hosni Mubarak** (*pictured*) is sentenced to three years in prison for corruption.
- The World Health Organization declares Liberia **Ebola-free**.
- A **prison break** in Iraq results in more than fifty prisoners escaping, as well as the deaths of an estimated fifty prisoners and twelve police officers.
- Scientists announce the discovery of **Lokiarchaeota**, a transitional form between Archaea and Eukaryotes.
- The UK's Conservative Party, led by **David Cameron**, **wins a majority of seats** in the House of Commons.

What links here Related changes Upload file Special pages Permanent link Page information Wikidata item

Print/export Create a book Download as PDF Printable version

Languages Simple English العربية Bahasa Indonesia Bahasa Melayu

FIREFOX SVG FILTERS

○ Into this!

```
<svg><filter>  
<feGaussianBlur  
stdDeviation="3"/>  
</filter></svg>
```



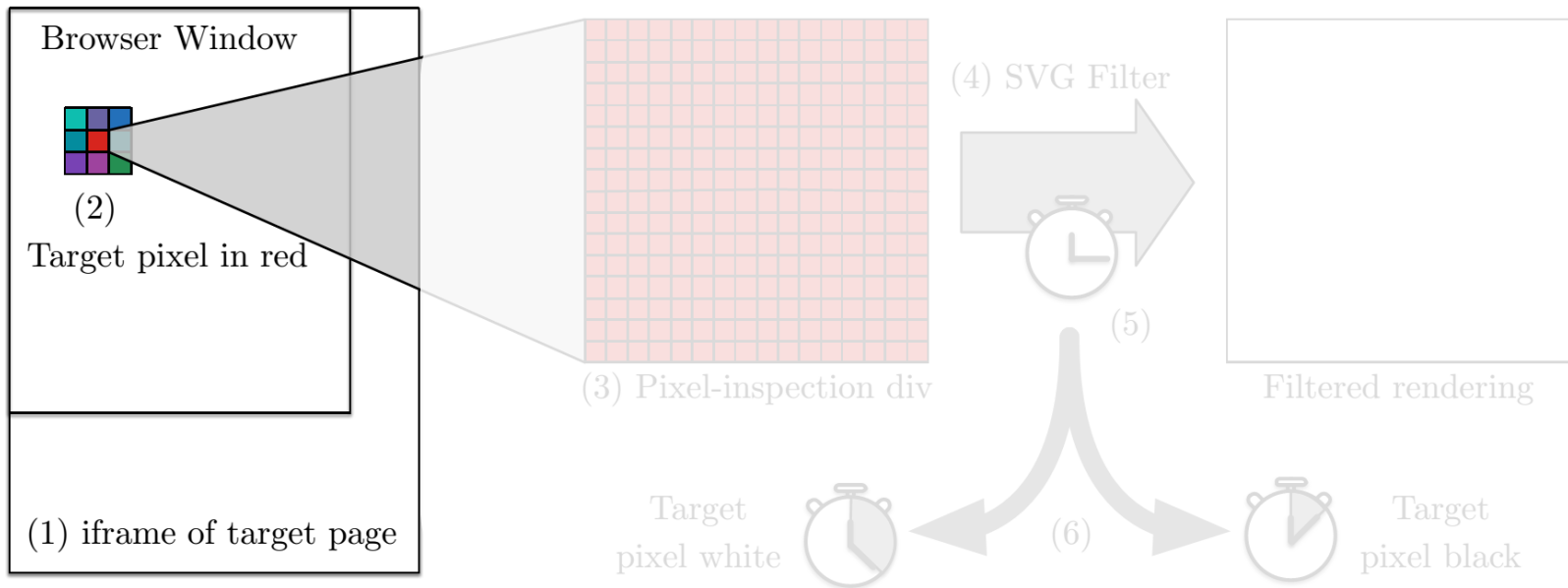
FIREFOX SVG FILTERS

- CSS defined filters
 - <div>
 - <iframe>
 - Really any element
- Run various functions
 - convolve
 - blur
 - skew
 - gradient
 - clipping
- Stackable!



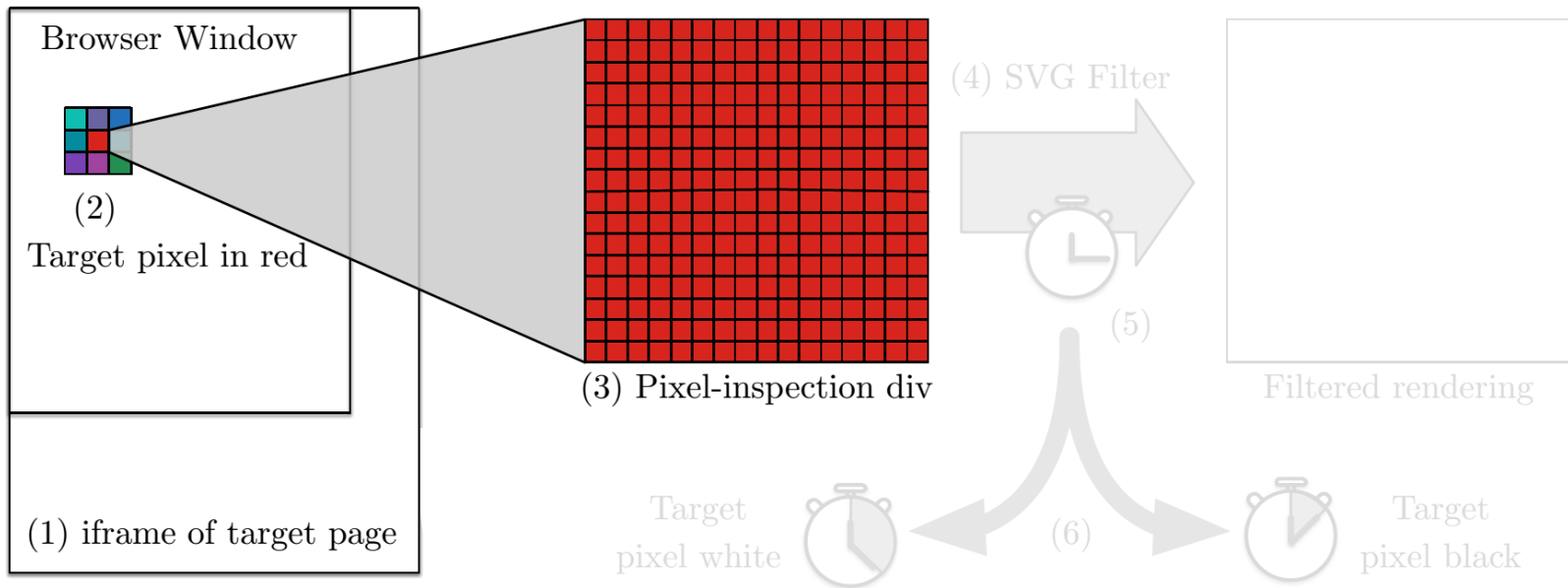
FIREFOX SVG FILTER TIMING ATTACK

- See Paul Stone's "Pixel Perfect Timing Attacks with HTML 5"



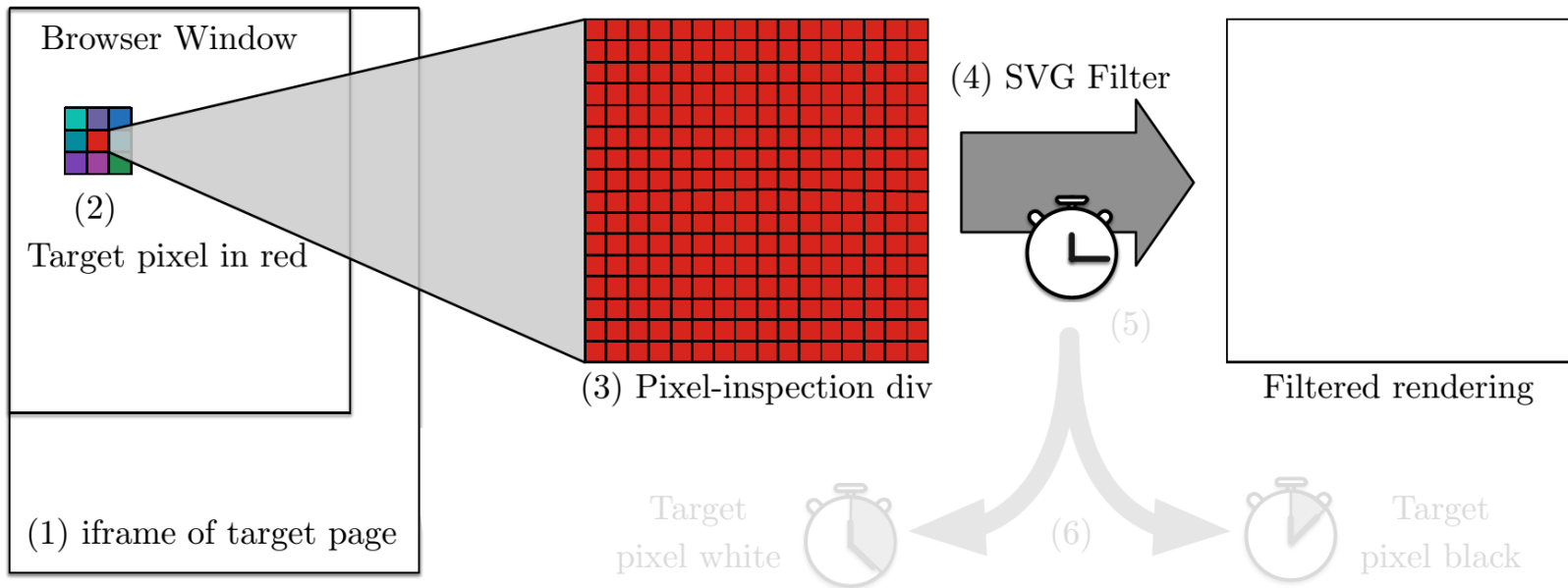
FIREFOX SVG FILTER TIMING ATTACK

- See Paul Stone's "Pixel Perfect Timing Attacks with HTML 5"



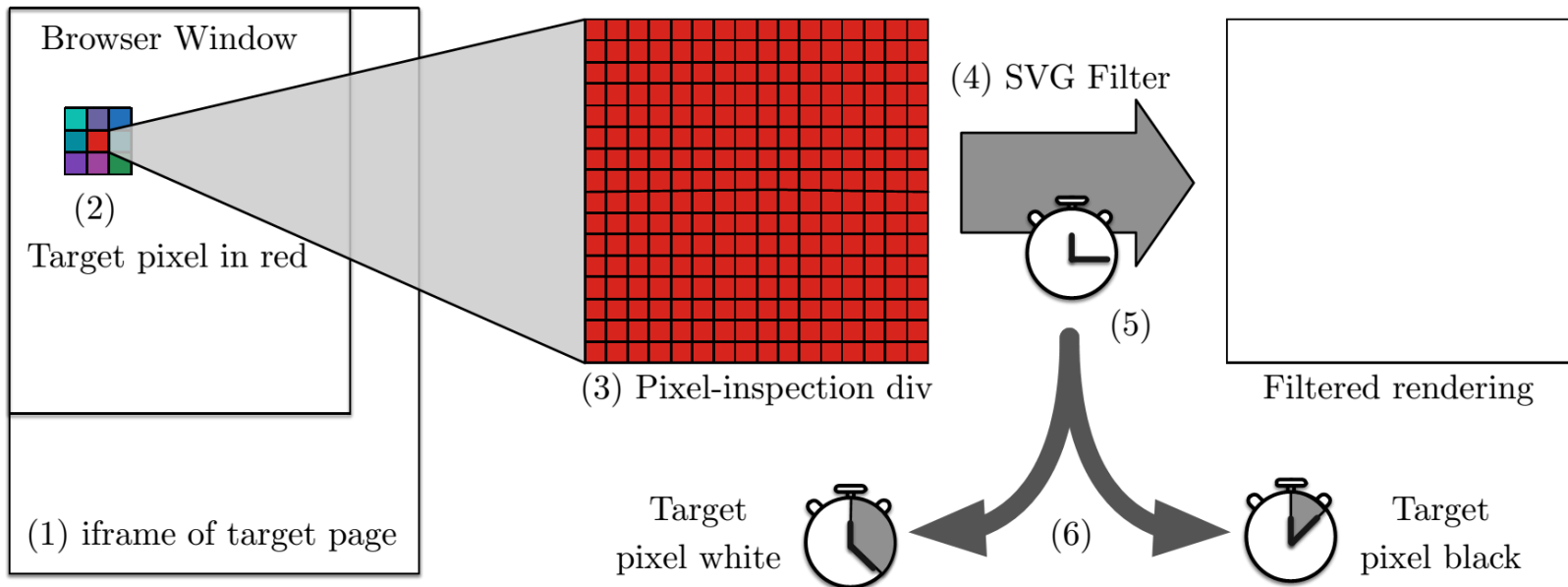
FIREFOX SVG FILTER TIMING ATTACK

- See Paul Stone's "Pixel Perfect Timing Attacks with HTML 5"



FIREFOX SVG FILTER TIMING ATTACK

- See Paul Stone's "Pixel Perfect Timing Attacks with HTML 5"



PAUL STONE'S SVG TIMING SIDE CHANNEL

- Relied on a *fast path optimization* in the femorphology SVG filter
 - In cases of a solid color image, filter ran much faster

```
if (x == rect.x || xExt[0] <= startX || xExt[1] <= startX ||  
    xExt[2] <= startX || xExt[3] <= startX){  
    [...]  
} else { // We only need to look at the newest column  
    for (PRUint32 y1 = startY; y1 <= endY; y1++) {  
        [...]    }
```

- Fix was to write constant time code!
 - Took ~2 years to land, and 150+ comment bug thread
 - “the problem boils down to: how to implement constant-time min(a, b) and max(a, b) in C++?” – Bugzilla thread



BACK TO THE PRESENT

NEW FIREFOX SVG FILTER ATTACK

- Firefox SVG Filters are still ‘vulnerable’ pending a timing difference
- We have a new timing side-channel source

NEW FIREFOX SVG FILTER ATTACK

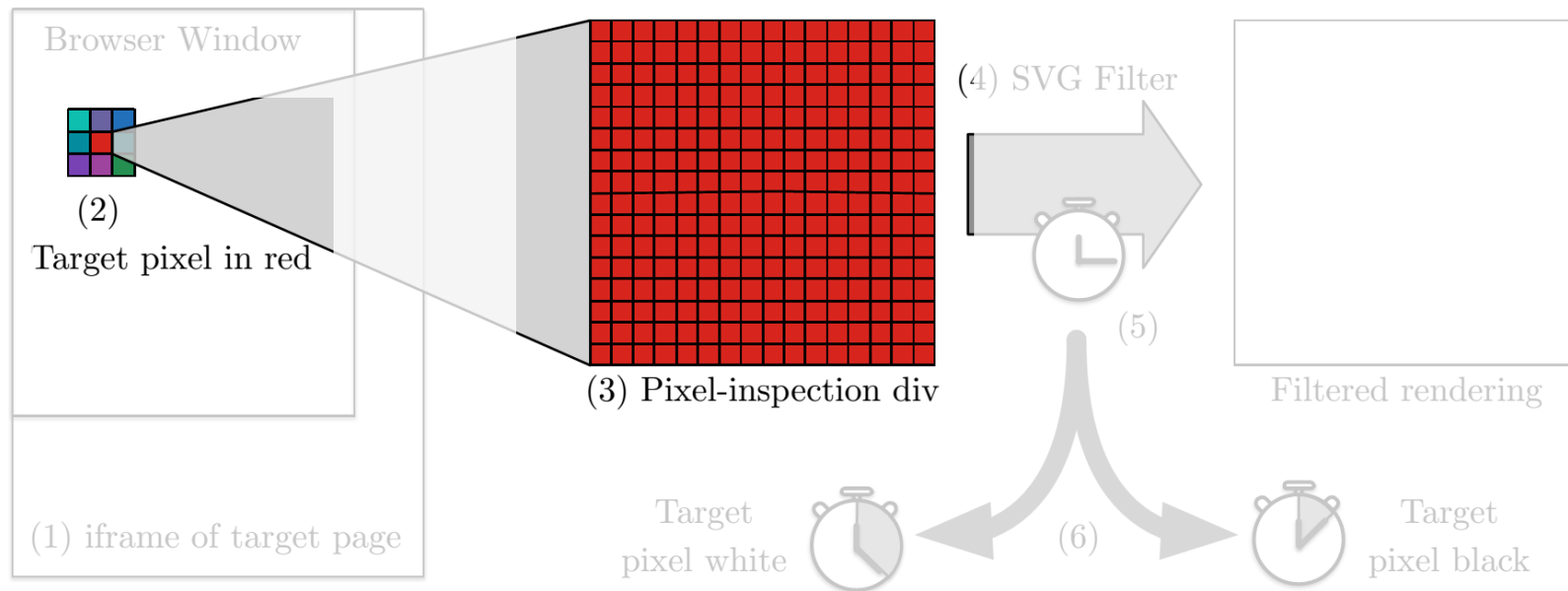
- Firefox SVG Filters are still ‘vulnerable’ pending a timing difference
- We have a new timing side-channel source
- SVG Filters run *floating point math!*

NEW FIREFOX SVG FILTER ATTACK

- Firefox SVG Filters are still ‘vulnerable’ pending a timing difference
- We have a new timing side-channel source
- SVG Filters run *floating point math!*
- We need an *amplifier*

NEW FIREFOX SVG FILTER ATTACK

- We need an *amplifier*



FIREFOX SVG FILTERS AND SUBNORMALS

```
def do_one_convolve(source_patch):  
    source_patch = [[ ?, ?, ? ]  
                    [ ?, ?, ? ]  
                    [ ?, ?, ? ]]  
  
    kernel = [[ 1e-42, 1e-42, 1e-42 ]  
              [ 1e-42, 1e-42, 1e-42 ]  
              [ 1e-42, 1e-42, 1e-42 ]]  
  
    for x,y in source_patch:  
        tmp[x][y] = source_patch[x][y] * kernel[x][y]  
  
    result = 0  
  
    for x,y in tmp:  
        result += tmp[x][y]  
  
    return result  
  
for x,y in source_image:  
    source_image[x][y] = do_one_convolve(swatch(x,y))
```

FIREFOX SVG FILTERS AND SUBNORMALS

```
def do_one_convolve(source_patch):  
    source_patch = [[ ?, ?, ? ]  
                    [ ?, ?, ? ]  
                    [ ?, ?, ? ]]  
  
    kernel = [[ 1e-42, 1e-42, 1e-42 ]  
              [ 1e-42, 1e-42, 1e-42 ]  
              [ 1e-42, 1e-42, 1e-42 ]]  
  
    for x,y in source_patch:  
        tmp[x][y] = source_patch[x][y] * kernel[x][y]  
  
    result = 0  
  
    for x,y in tmp:  
        result += tmp[x][y]  
  
    return result  
  
for x,y in source_image:  
    source_image[x][y] = do_one_convolve(swatch(x,y))
```

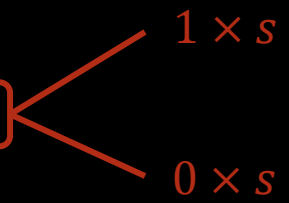
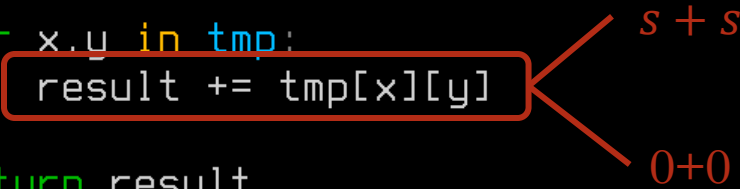


Diagram illustrating the operation: $1 \times s$ and $0 \times s$ are shown as results of the operation $\text{source_patch}[x][y] * \text{kernel}[x][y]$.

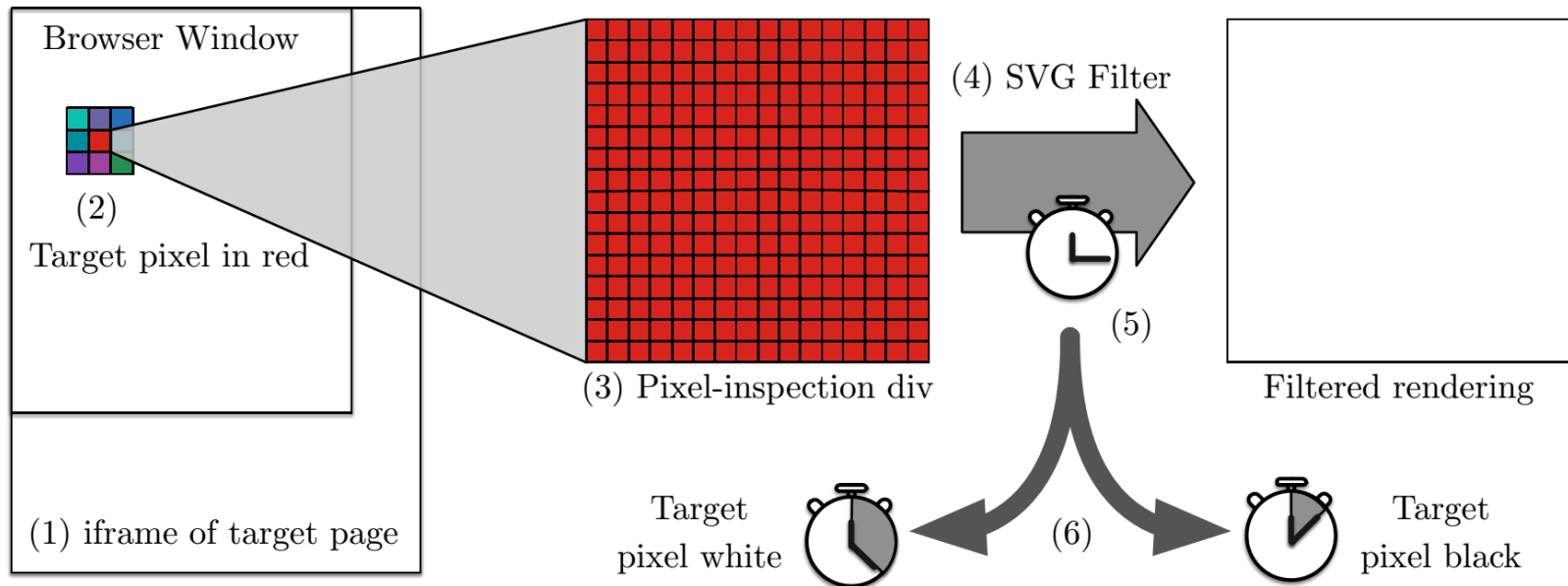
FIREFOX SVG FILTERS AND SUBNORMALS

```
def do_one_convolve(source_patch):  
    source_patch = [[ ?, ?, ? ]  
                    [ ?, ?, ? ]  
                    [ ?, ?, ? ]]  
  
    kernel = [[ 1e-42, 1e-42, 1e-42 ]  
              [ 1e-42, 1e-42, 1e-42 ]  
              [ 1e-42, 1e-42, 1e-42 ]]  
  
    for x,y in source_patch:  
        tmp[x][y] = source_patch[x][y] * kernel[x][y]  
  
    result = 0  
  
    for x,y in tmp:  
        result += tmp[x][y]  
  
    return result  
  
for x,y in source_image:  
    source_image[x][y] = do_one_convolve(swatch(x,y))
```



A diagram with two orange arrows pointing to the `result += tmp[x][y]` line. The top arrow is labeled $s + s$ and the bottom arrow is labeled $0 + 0$.

FIREFOX SVG FILTER TIMING ATTACK



FIREFOX SVG FILTERS ATTACK IMPACT



- Firefox does not consider running SVG filters over foreign pixels a violation of SOP
 - We disagree
 - Cross Origin Resource Sharing (CORS) is the obvious solution

READING PIXELS

- From other origins

- Reconstruct characters (OCR)
- Extract usernames, login status, user information, etc
- Blocked with frame options or CSP

```
TOP SECRET//SI//ORCON//NOFORN
Black: Avg:24.39002319999986
White: Avg:61.10947299999989
size: 200
midpt: 42.74974809999988
Total Runtime (M:S) 2:37
Pixels stolen: 3750
Pixels / second: 23.88535031847134
```

google
foobkjfd

- From *our* origin

- History sniffing



AVOIDING FLOATING POINT PROBLEMS

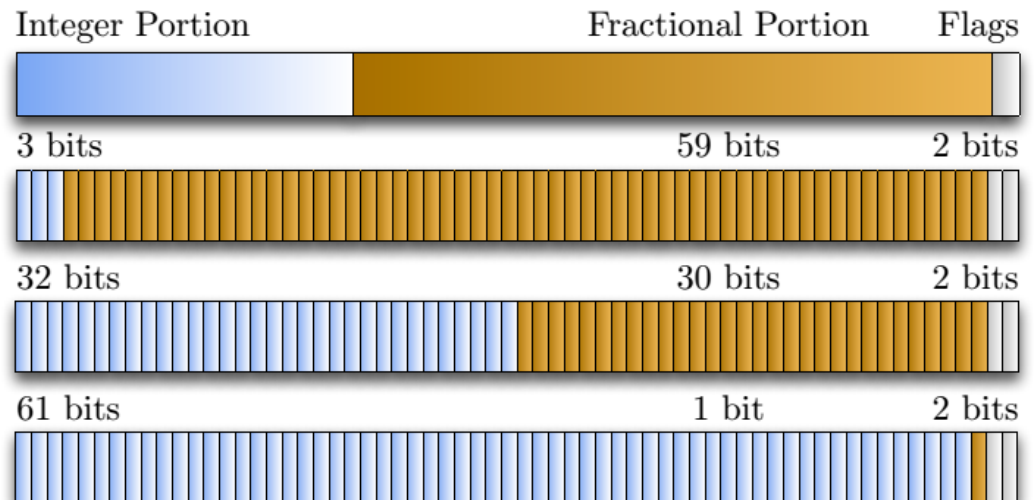
RECOMMENDATIONS

- Don't use floating point in security critical code
 - Unpredictable results
 - Large timing variations
 - Highly processor and build dependent
- Use Fixed Point if you need non-integer math

LIBFTFP – FIXED TIME FIXED POINT

- C library implementing most math operations
 - Add, divide, etc
 - Transcendentals
 - Exponents, logs, etc

- Variable Width



- Constant time! (Probably!)

BUILDING LIBFTFP

- Techniques
 - No data dependent jumps (&&, if, etc.)
 - No known variable time instructions (div, idiv, etc.)
 - No look-up tables (due to caching)
- We cannot be 100% sure of the constant-ness of our code
 - Intel doesn't release any information about instruction data dependency
 - We cannot exhaustively test processors and instruction arguments
- Writing constant time code is a battle against all future processors and compilers
- LibFTFP uses approximations

LIBFTFP STATISTICS

- Comparing to hardware slightly unfair
- Comparing to infinite precision software (MPFR) also slightly unfair

Function	FTFP	SSE	MPFR
neg	6	5	12-20
abs	9	4	10-17
cmp	21	5	10-15
add	15	4	15-58
sub	15	5	14-61
mul	43	5	16-76
div	381	7-15	15-170
floor	8	5	12-48
ceil	11	5	12-56
exp	1,460	7-16	37-13,330
ln	681	11-20	18-6,900
log2	679	9-20	19-24,000
log10	674	9-21	19-18,000
sqrt	7,870	7-16	9-154
pow	2,330	11-78	40-72,000
sin	1,998	–	11-33,000
cos	1,990	–	34-29,000
tan	2,380	–	13-37,000
print	443	350-600	210-230

TAKEAWAYS

- Security critical code should **omit floating point** or be extremely careful
- Writing **provably constant time code** is impossible
 - Intel? Some help here?
- Browsers should require **CORS/CSP for computing over all foreign data**
 - Like pixels

FUTURE WORK

- Firefox attack works on FF 23-27
 - Attack stopped working when filters changed to GPU
- GPU floating point implementations
 - “On NVIDIA GPUs starting with the Fermi architecture [...] multi-instruction sequences such as square root and [...] reciprocal square root, must do extra work and take a slower path for denormal values “
- Other math operation data side channels
 - `imul`, `div/idiv` cycle counts are data dependent
 - What can we break with that?



QUESTIONS?

dkohlbre@cs.ucsd.edu

LibFTFP: github.com/kmowery/libfixedtimefixedpoint