hardwear.io

Hardware Security Conference and Training

Shaking trust in hardware: Attacks on hardware from software

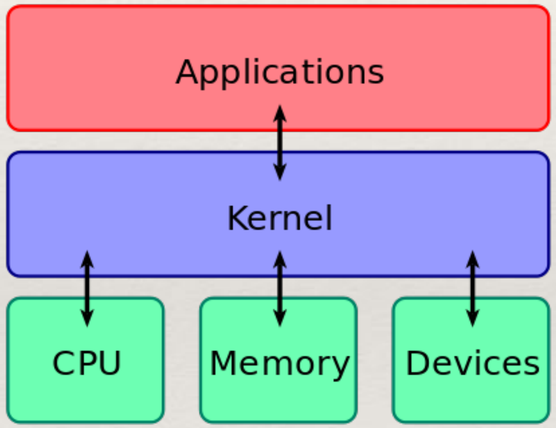
Ben Gras
Kaveh Razavi
Erik Bosman
Bart Preneel
Herbert Bos
Cristiano Giuffrida

We are ..

- VUSec systems security academic research group
- Defensive & offensive security projects using systems techniques
- VU University in Amsterdam
- `@vu5ec` on Twitter

Software depends on hardware

- Classic hierarchy



- And hardware is solid right?
- CPU and memory can't leak or corrupt data?

Software depends on hardware

Overview: two ways hardware can let you down

- **Breaking ASLR: an MMU side channel (AnC)**
- Compute virtual addresses of data & code
- With microarchitectural MMU side channel, not software
- Even from a JavaScript sandbox, 22 u-archs
- **This leaks secrets**

Overview: two ways hardware can let you down

- **Rogue Memory Write with Rowhammer (Flip Feng Shui)**
- Change a bit in a memory page of a victim
- Demonstrated with cross-VM memory write attack
- Cross-VM Break GPG (apt) and SSH RSA security
- **This changes victim's data**

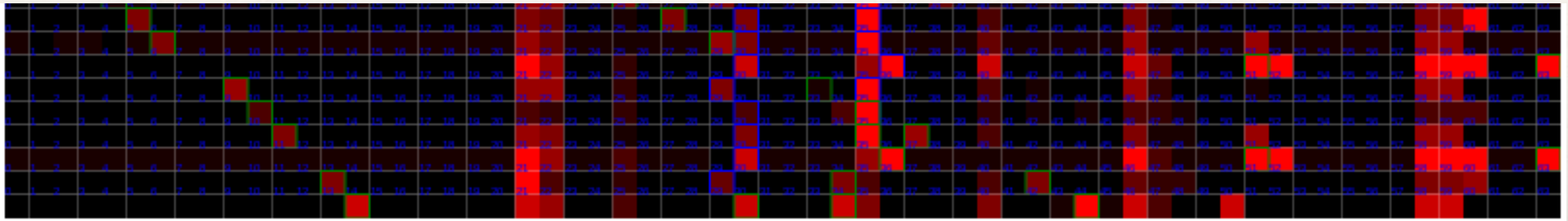
Overview: two ways hardware can let you down

- **Driven from Software**
- **We exploit Hardware Flaws**
- **No software bugs**

ANC: MMU SIDE CHANNEL

Teaser - AnC

- Visualization - JavaScript - and no software bug

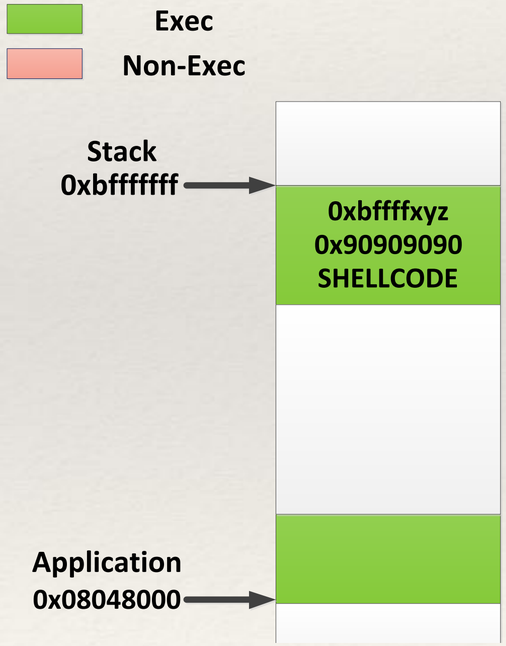


```
24.457 got level 4 - start slot 148, address 0x94000
24.993 got level 3 - start slot 295, address 0x24e94000
24.993 estimated remaining entropy 6 slot solutions: -1,-1,295,148
68.737 got level 4 - start slot 0, address 0x0
69.502 got level 3 - start slot 359, address 0x2ce00000
70.259 got level 2 - start slot 411, address 0x66ece00000
88.041 got level 1 - start slot 238, address 0x7766ece00000
88.041 estimated remaining entropy 0 slot solutions: 238,411,359,0
data: 0x7766ece00000, code slots: -1,-1,295,148, code: 0x7966e4e94000
```

- There will be a demo video

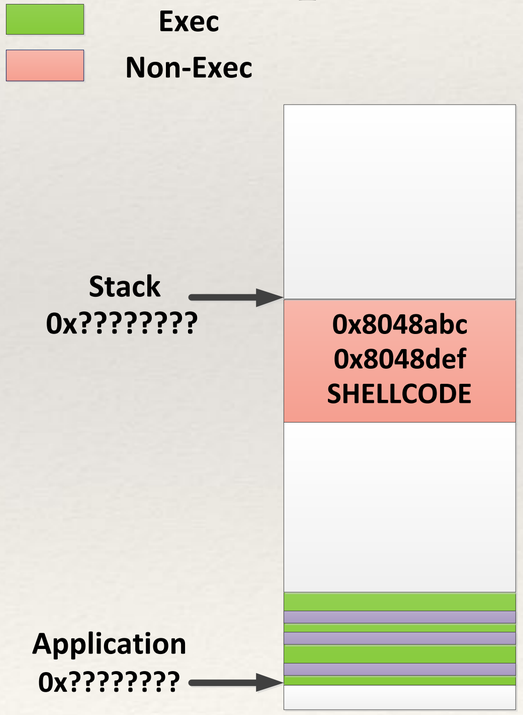
ASLR

- Main justification
- Response to exploitation in the 90s

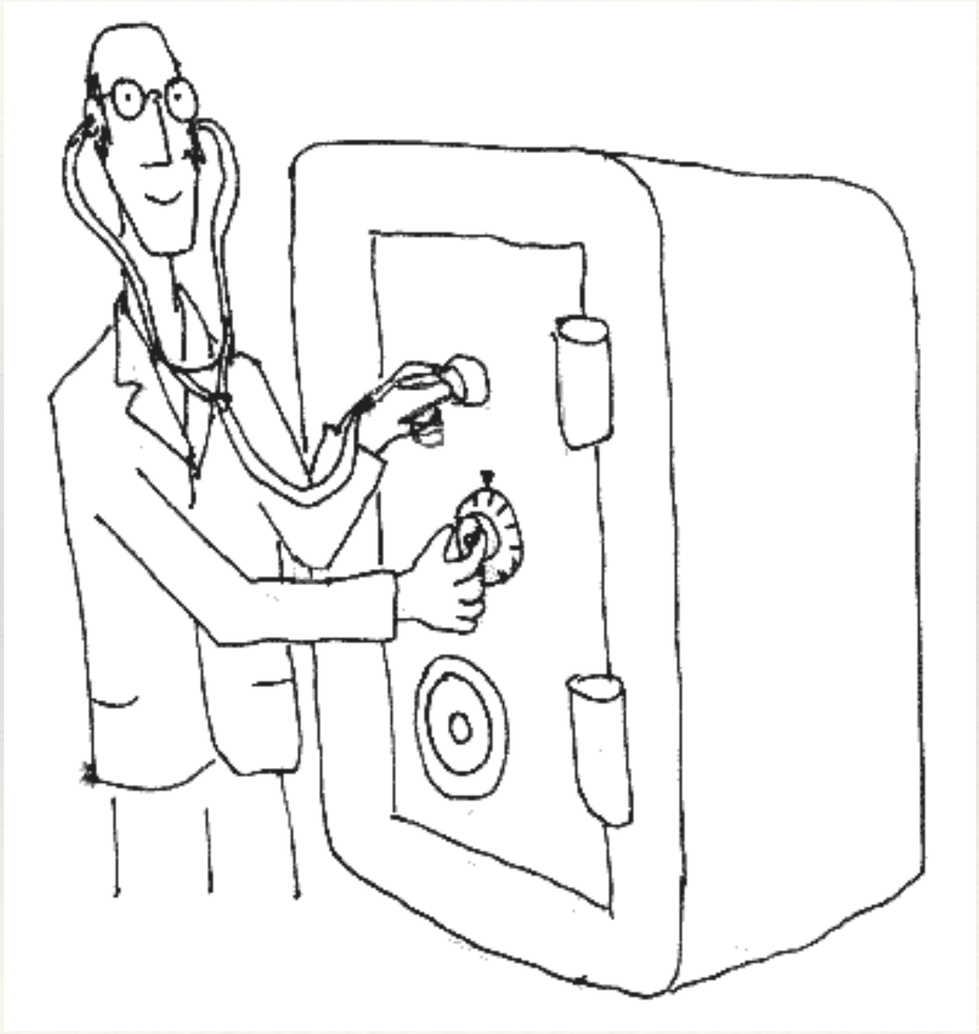


ASLR

- Let's randomize both areas: ASLR
- Also DEP. So exploitation requires ASLR leak and ROP

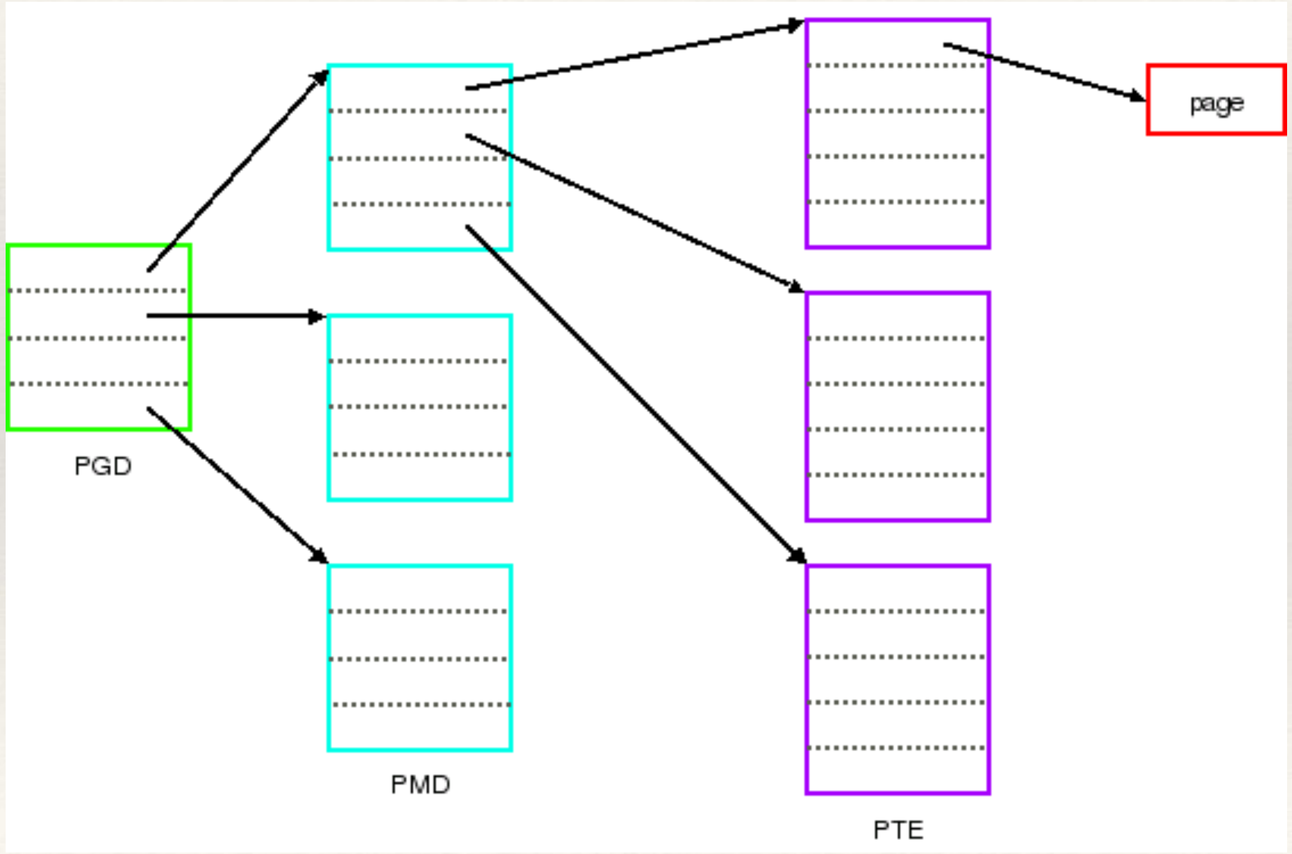


Side Channels



Pagetable Walks From DRAM

- Page tables point to the next step in a tree



Pagetable Walks From DRAM

0x644b321f4000

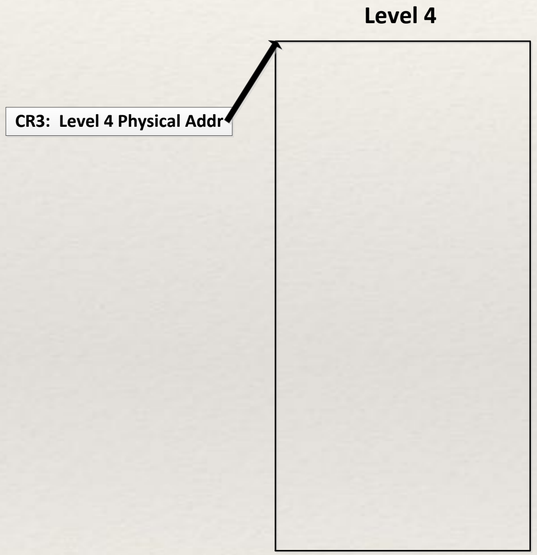
110010001001011001100100001111101000000000000000

CR3: Level 4 Physical Addr

Pagetable Walks From DRAM

0x644b321f4000

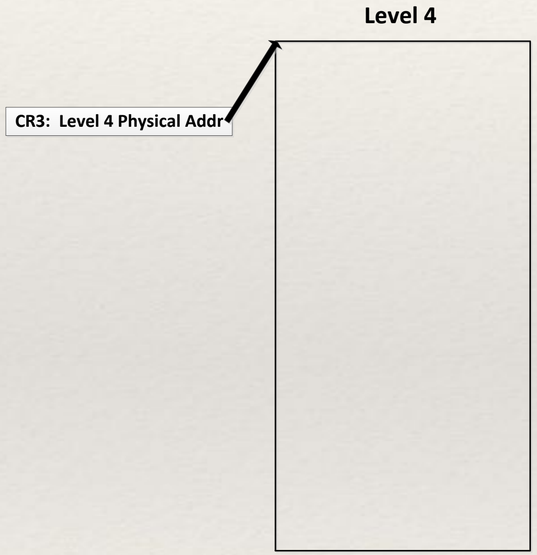
110010001001011001100100001111101000000000000000



Pagetable Walks From DRAM

0x644b321f4000

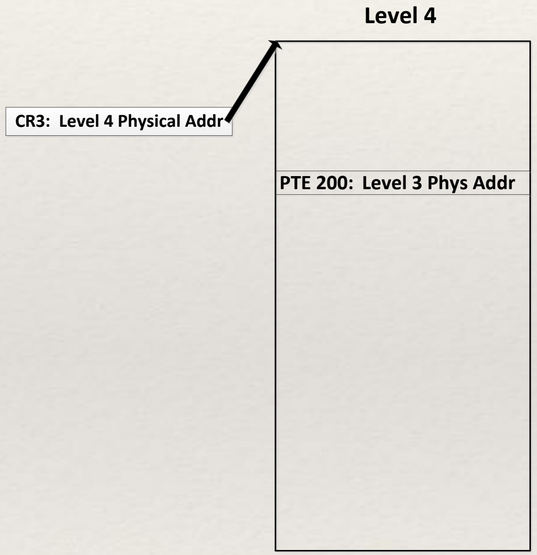
110010001001011001100100001111101000000000000000



Pagetable Walks From DRAM

0x644b321f4000

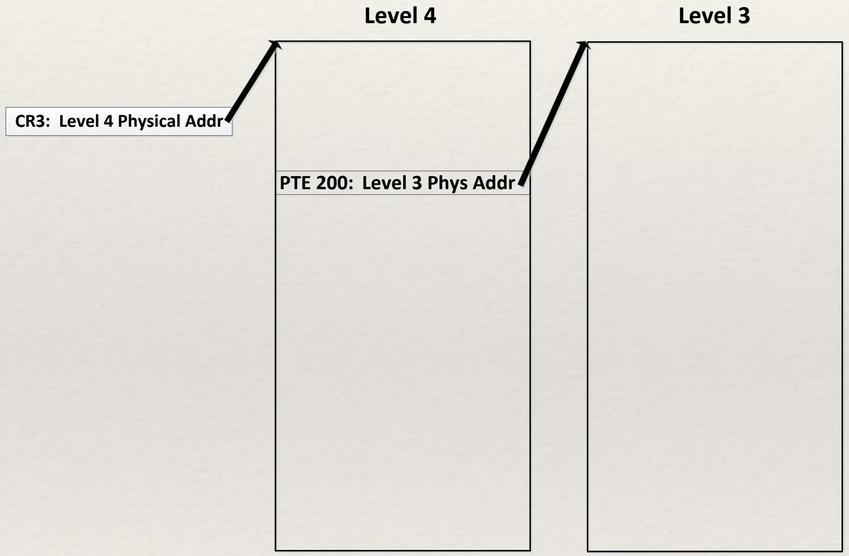
110010001001011001100100001111101000000000000000



Pagetable Walks From DRAM

0x644b321f4000

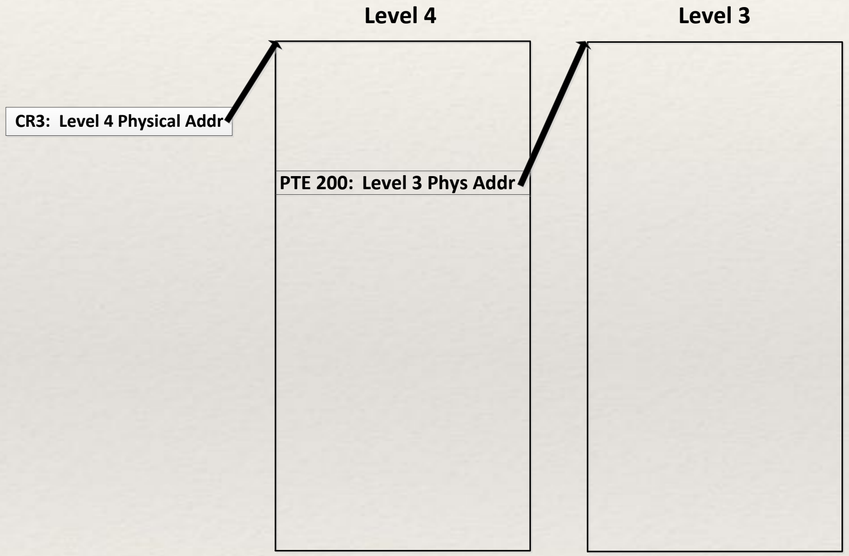
110010001001011001100100001111101000000000000000



Pagetable Walks From DRAM

0x644b321f4000

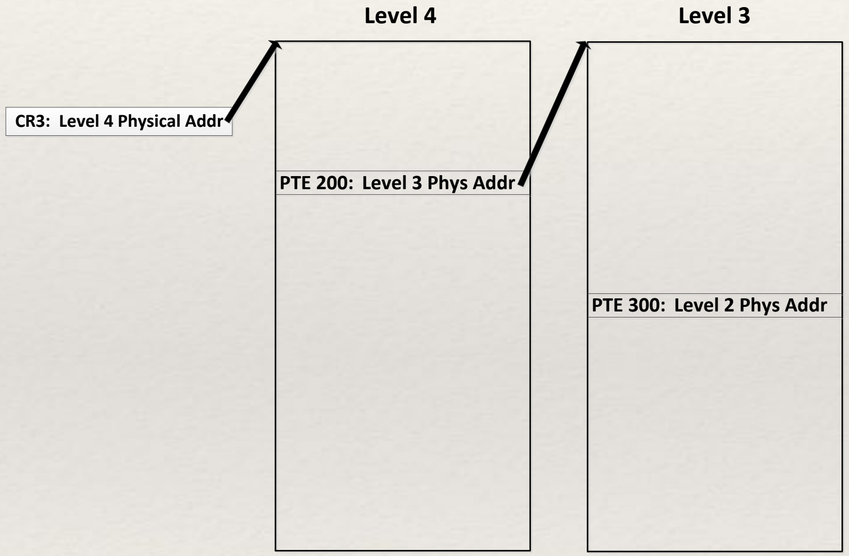
1100100010010110011001000011111010000000000000



Pagetable Walks From DRAM

0x644b321f4000

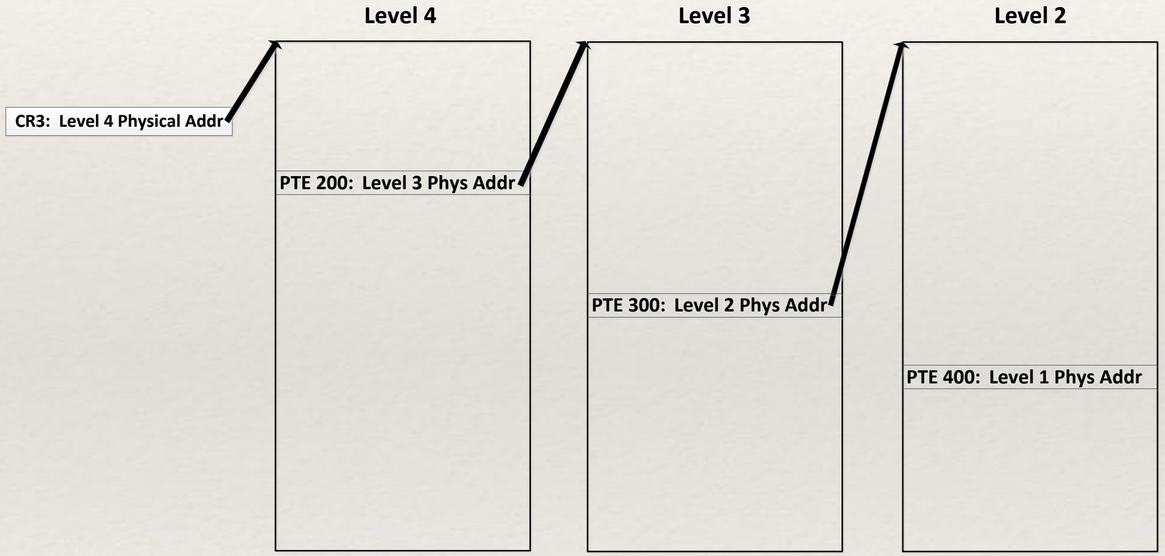
1100100010010110011001000011111010000000000000



Pagetable Walks From DRAM

0x644b321f4000

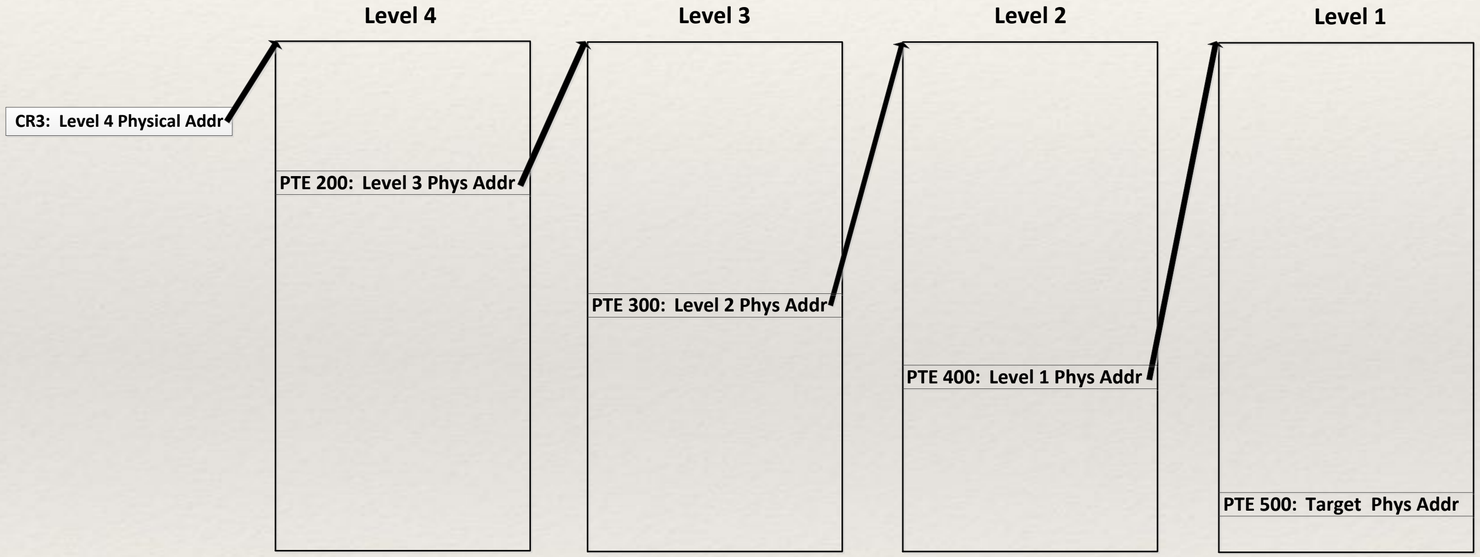
110010001001011001100100001111101000000000000000



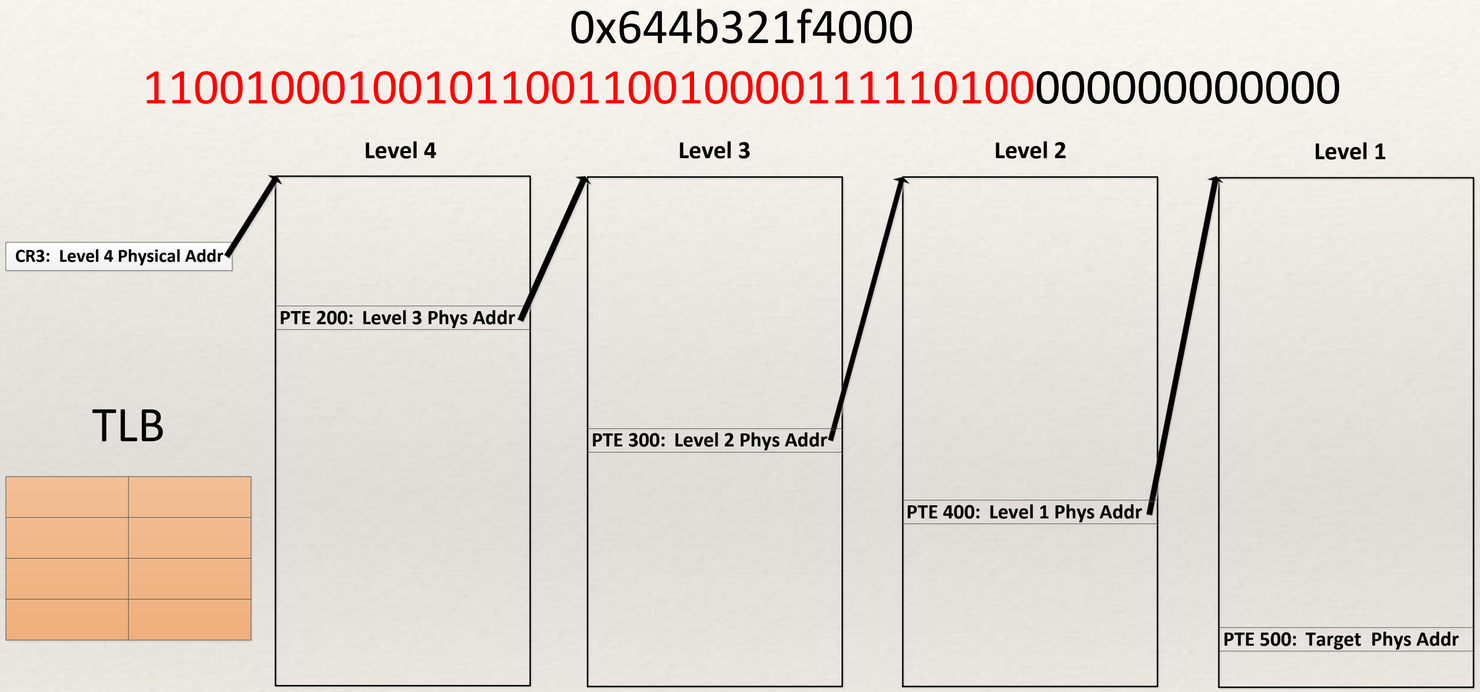
Pagetable Walks From DRAM

0x644b321f4000

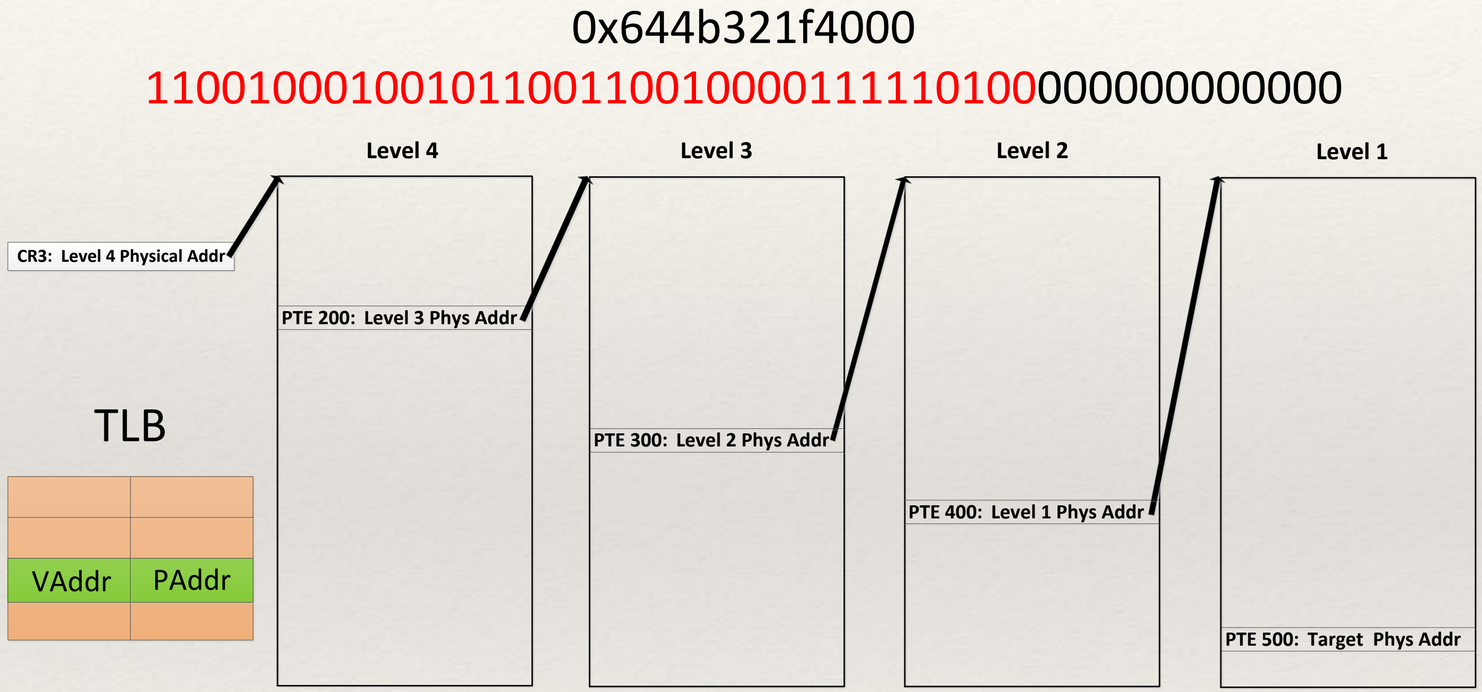
110010001001011001100100001111101000000000000



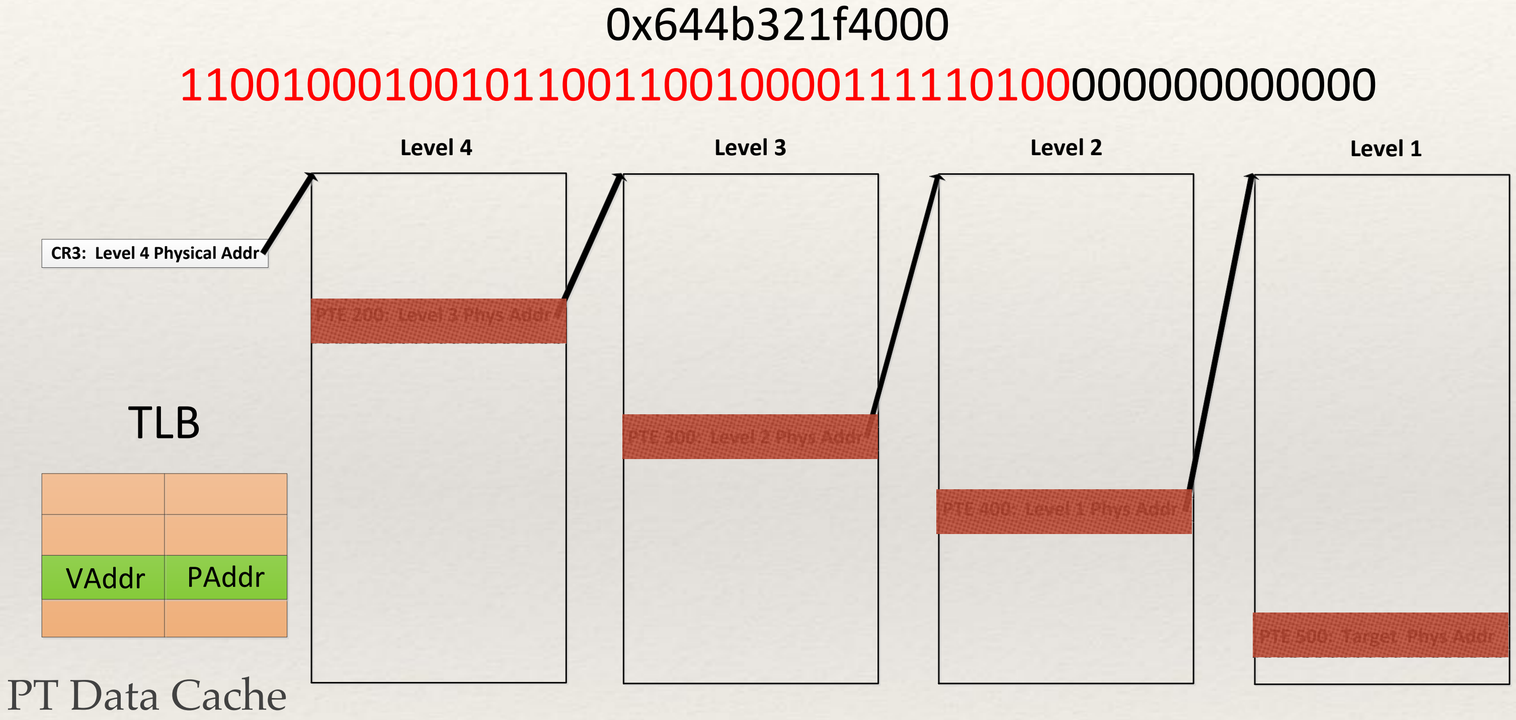
Pagetable Walks From DRAM



Pagetable Walks From DRAM

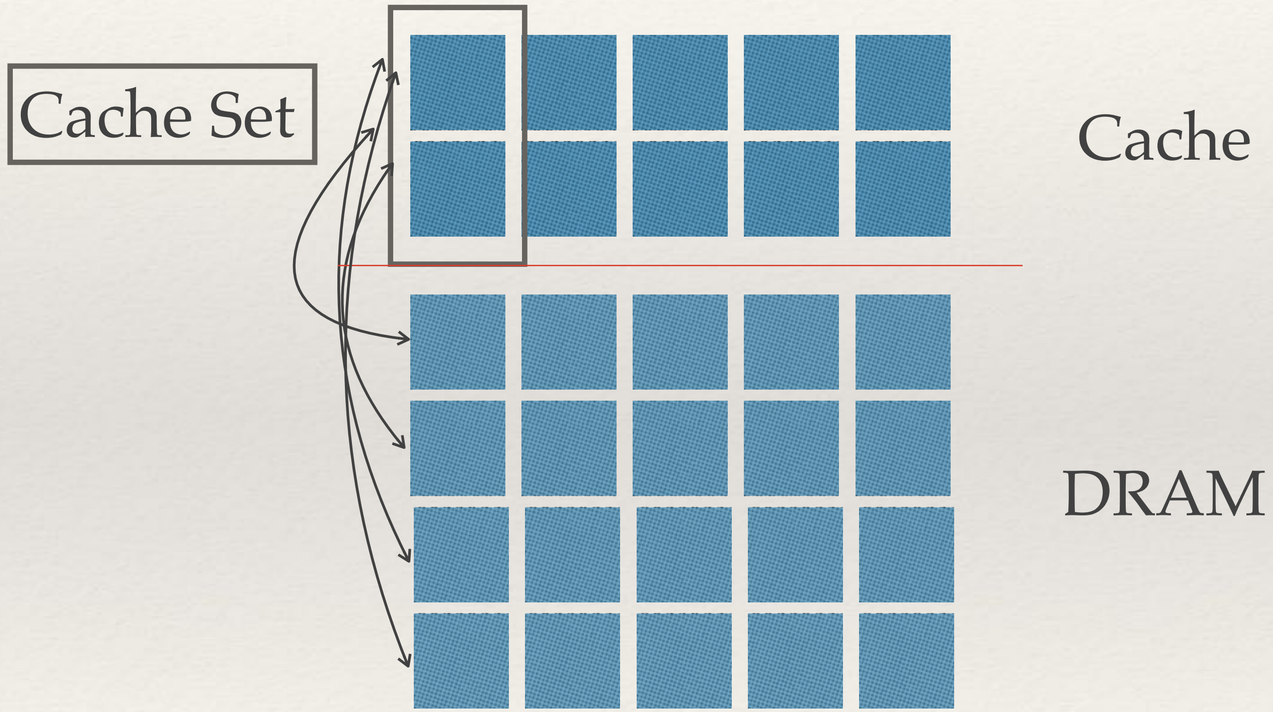


Pagetable Walks From DRAM



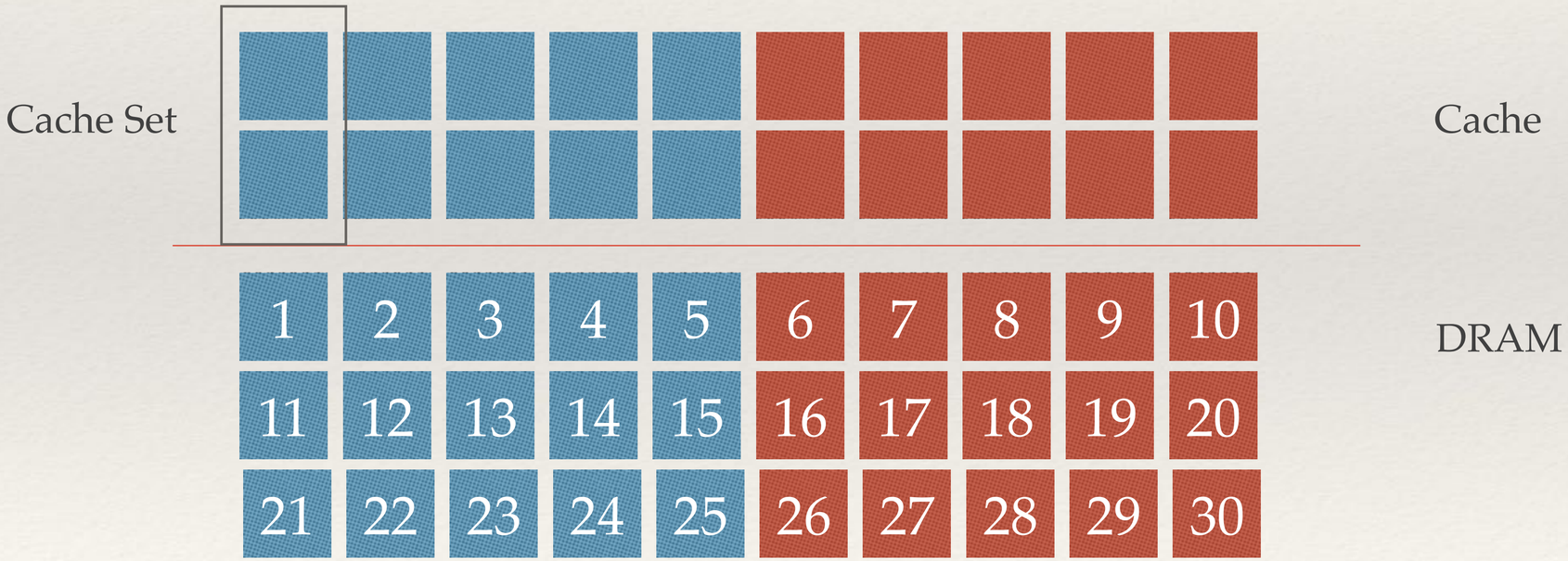
CPU Caches

- Memory cache lines can only go into one small cache set



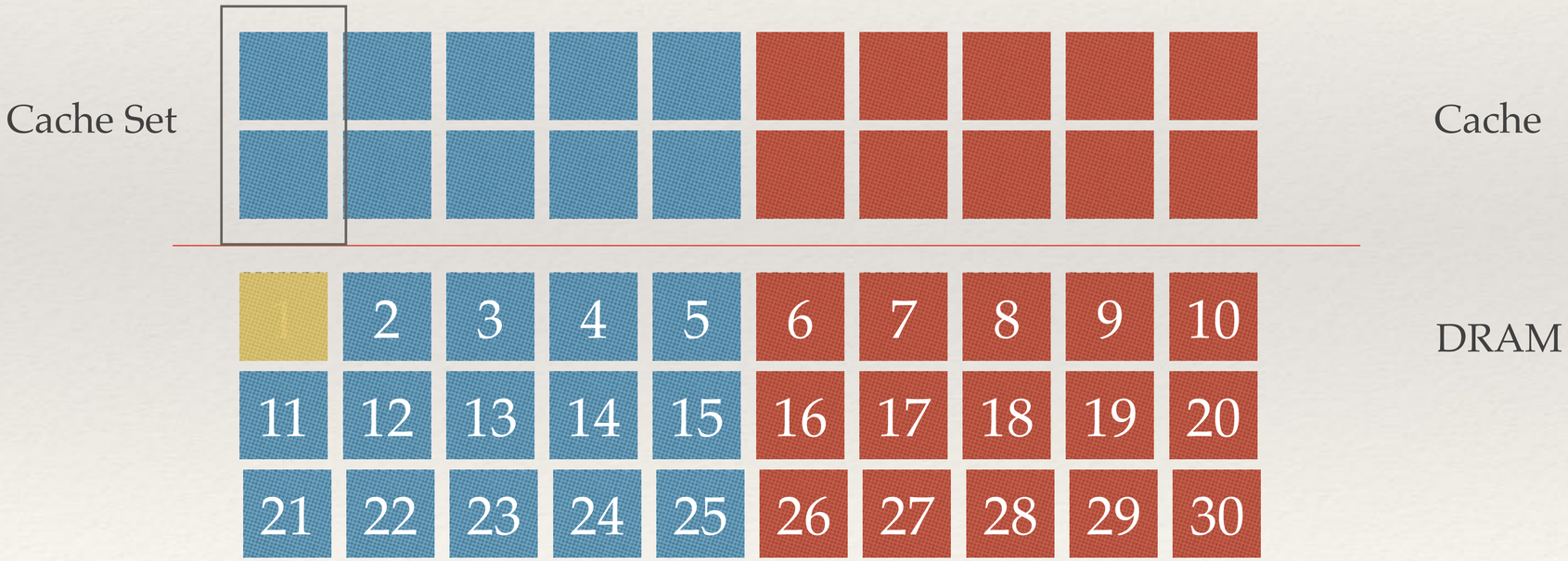
Tiny Cache Example

- 2-way cache, 5 sets per page, showing 2 colors



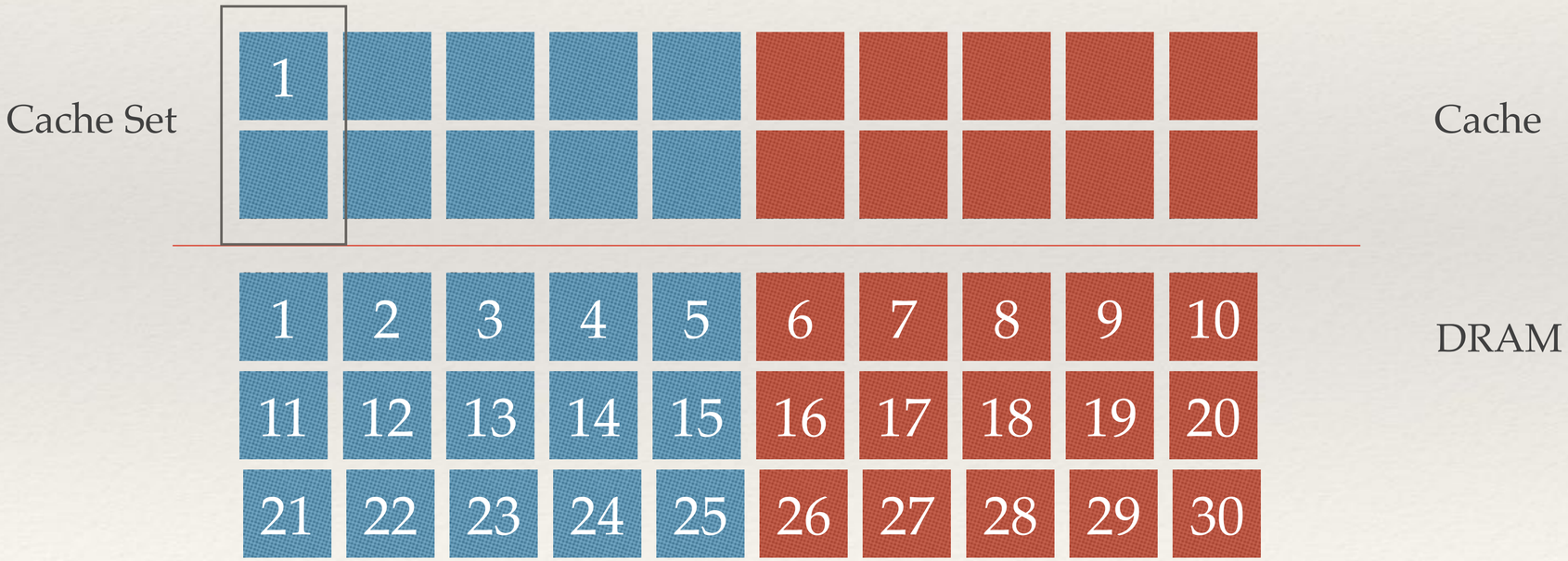
Tiny Cache Example

- 2-way cache, 5 sets per page, showing 2 colors



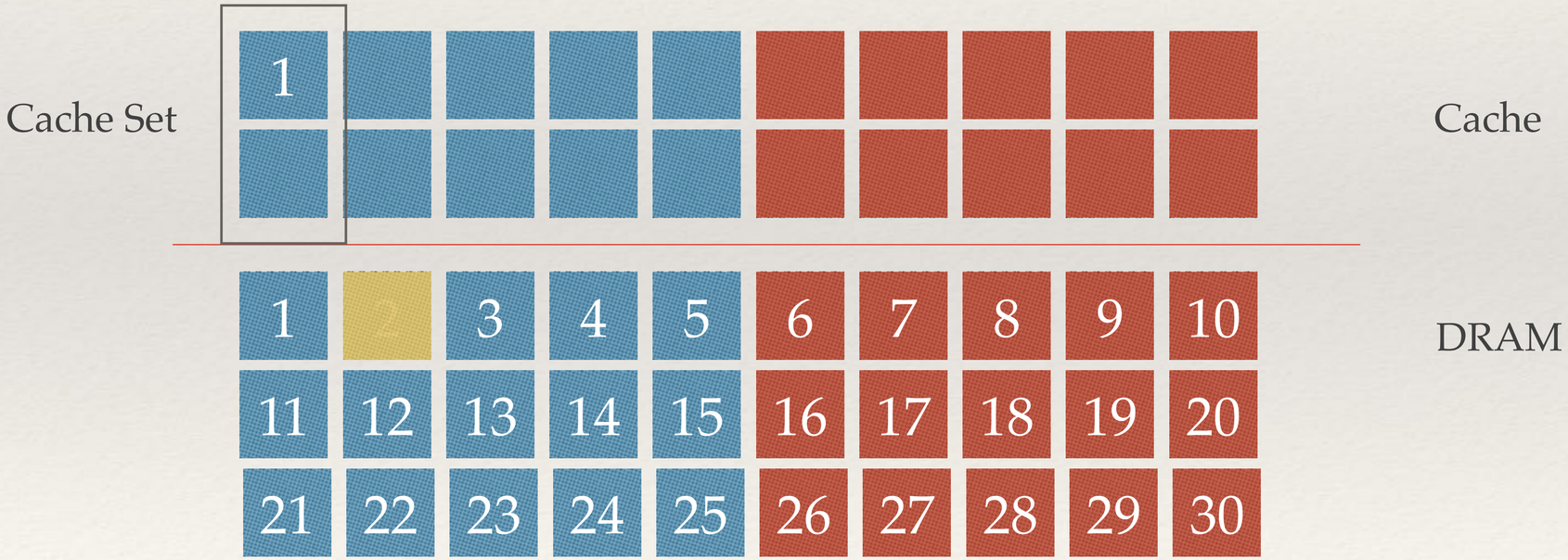
Tiny Cache Example

- 2-way cache, 5 sets per page, showing 2 colors



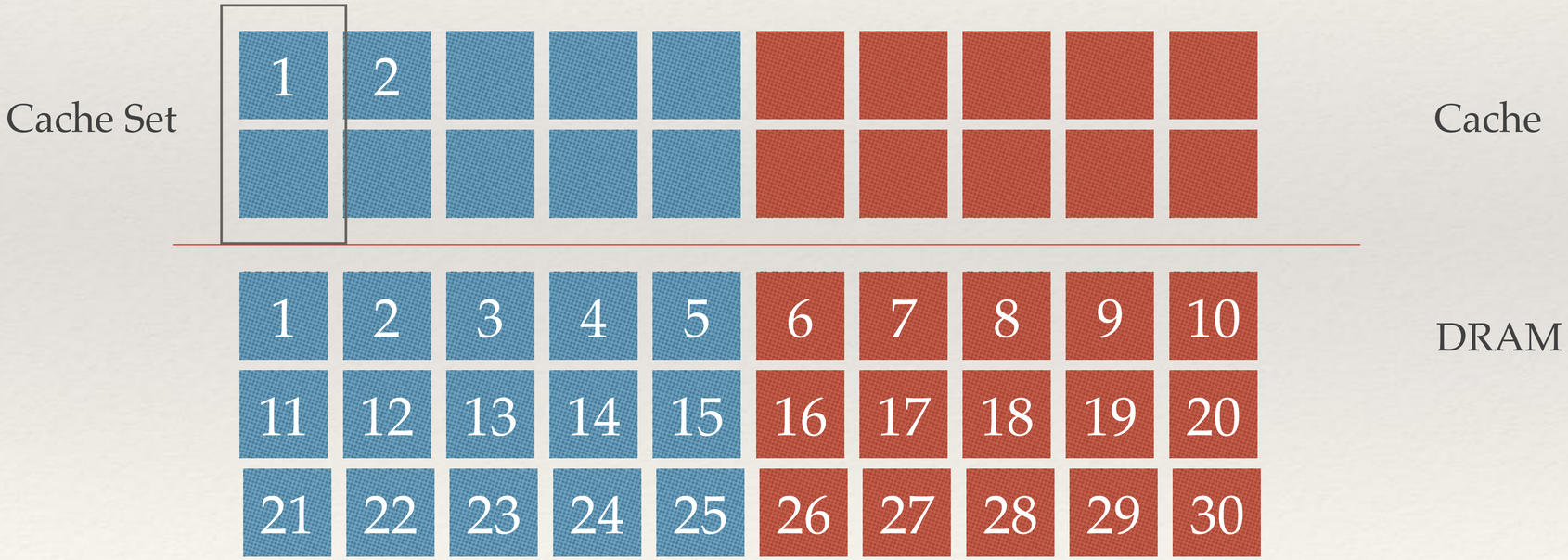
Tiny Cache Example

- 2-way cache, 5 sets per page, showing 2 colors



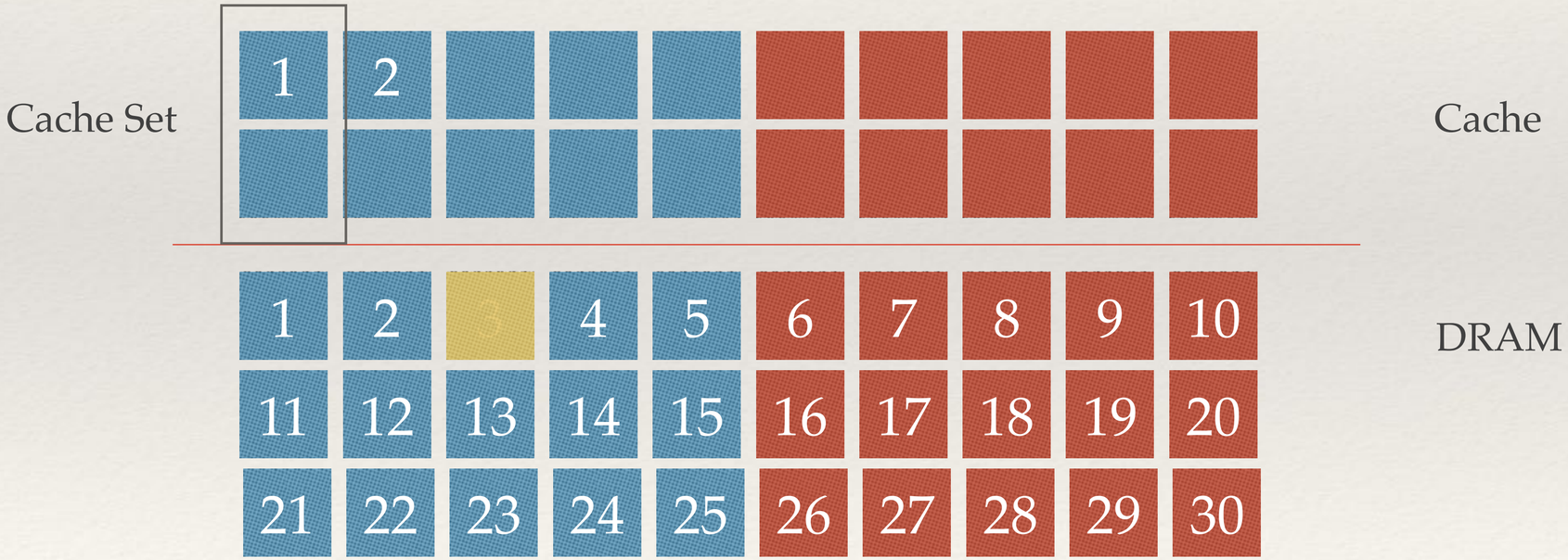
Tiny Cache Example

- 2-way cache, 5 sets per page, showing 2 colors



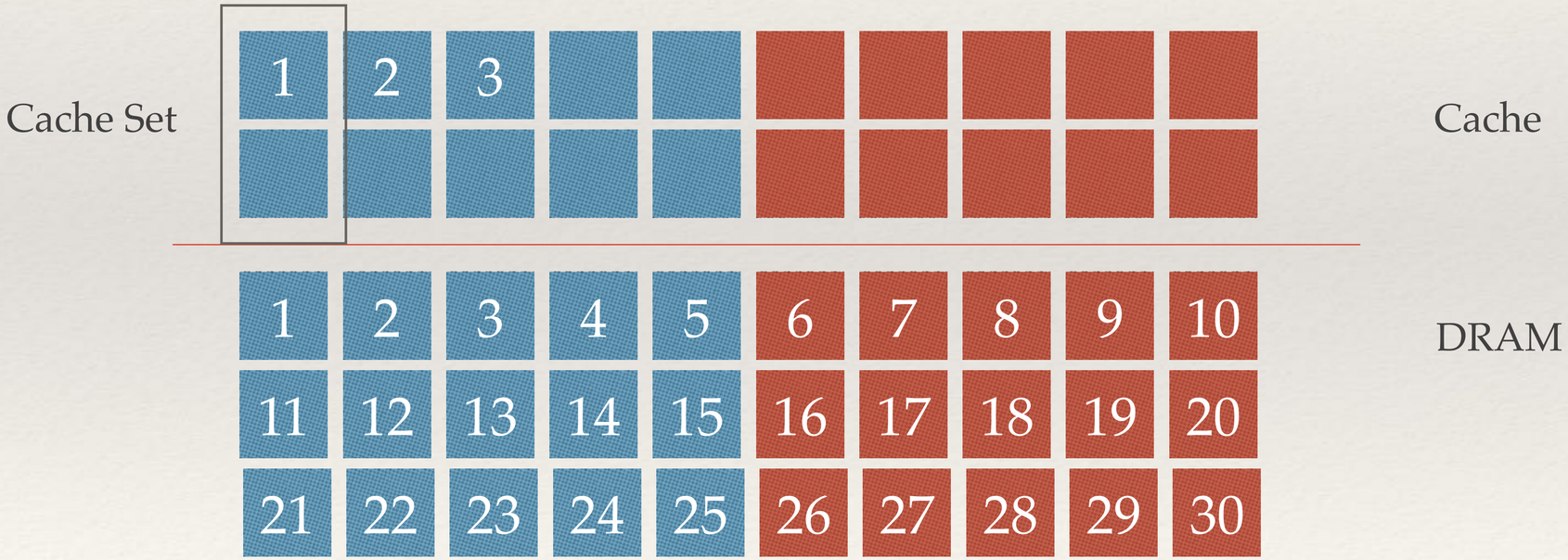
Tiny Cache Example

- 2-way cache, 5 sets per page, showing 2 colors



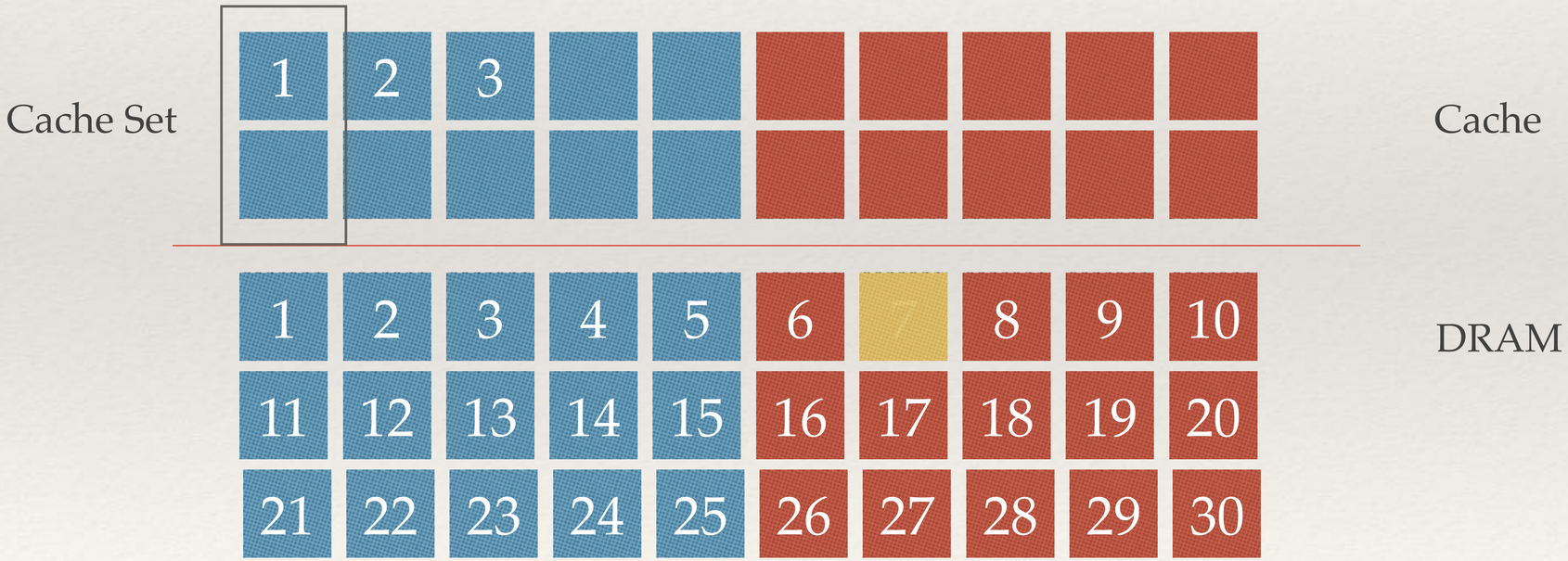
Tiny Cache Example

- 2-way cache, 5 sets per page, showing 2 colors



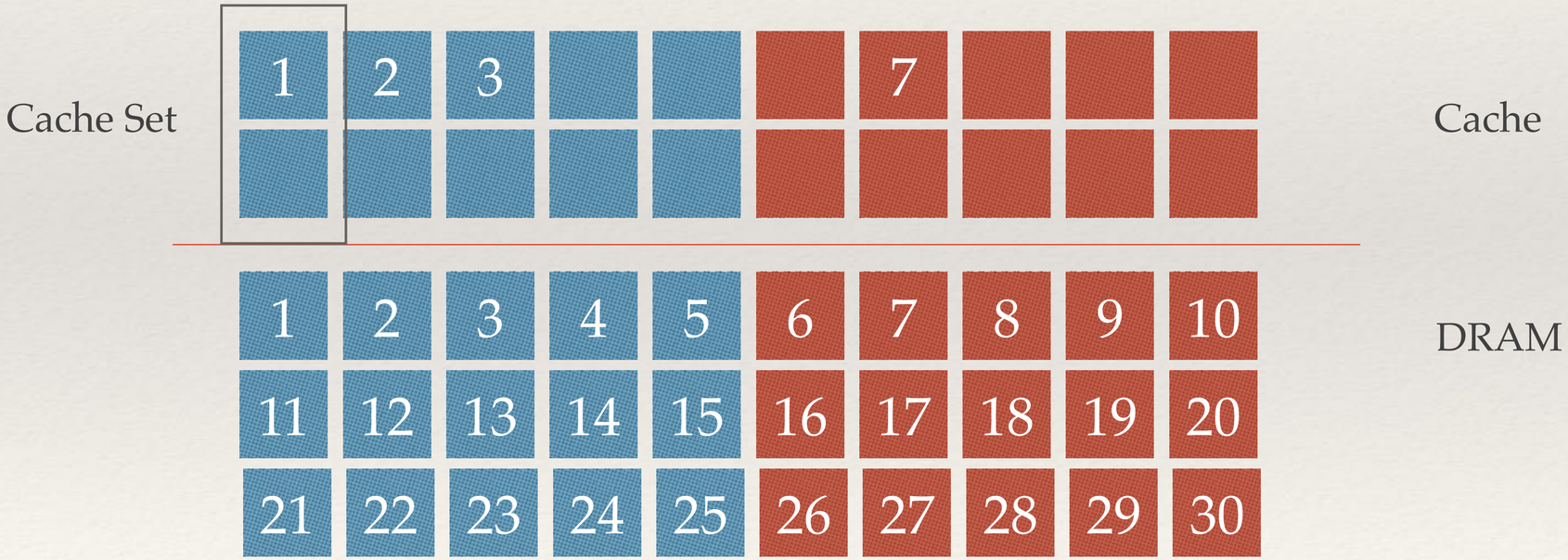
Tiny Cache Example

- 2-way cache, 5 sets per page, showing 2 colors



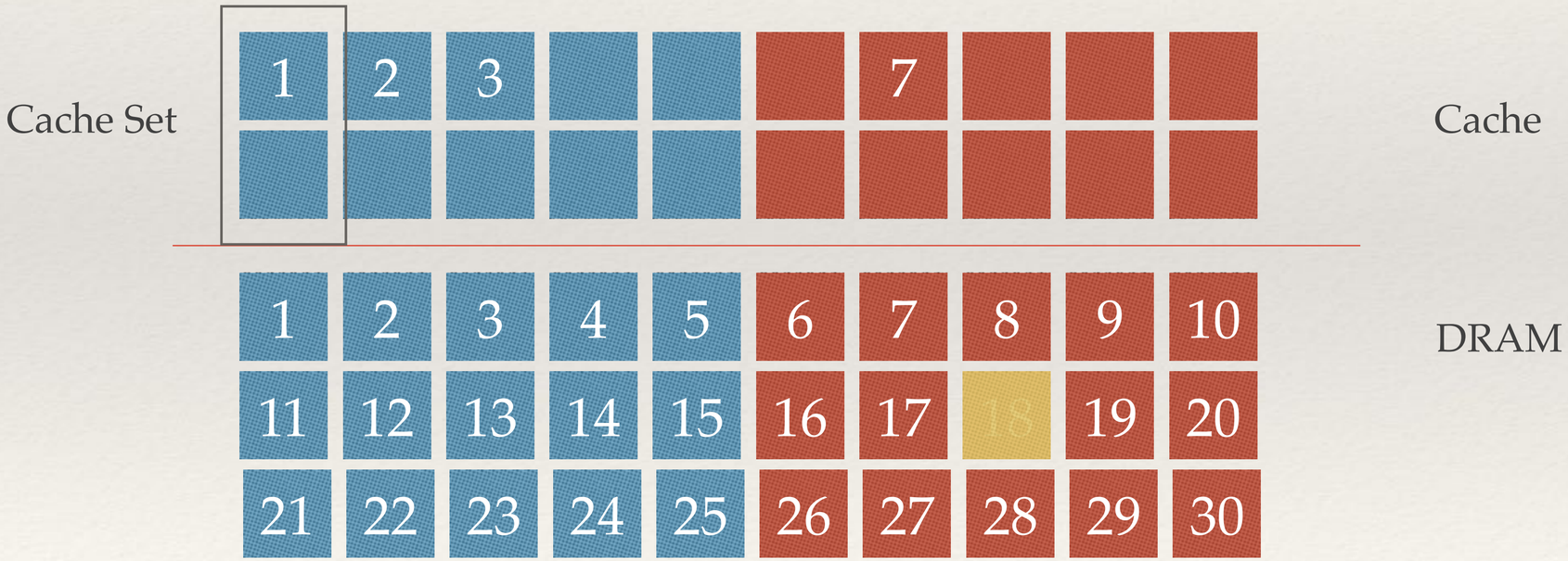
Tiny Cache Example

- 2-way cache, 5 sets per page, showing 2 colors



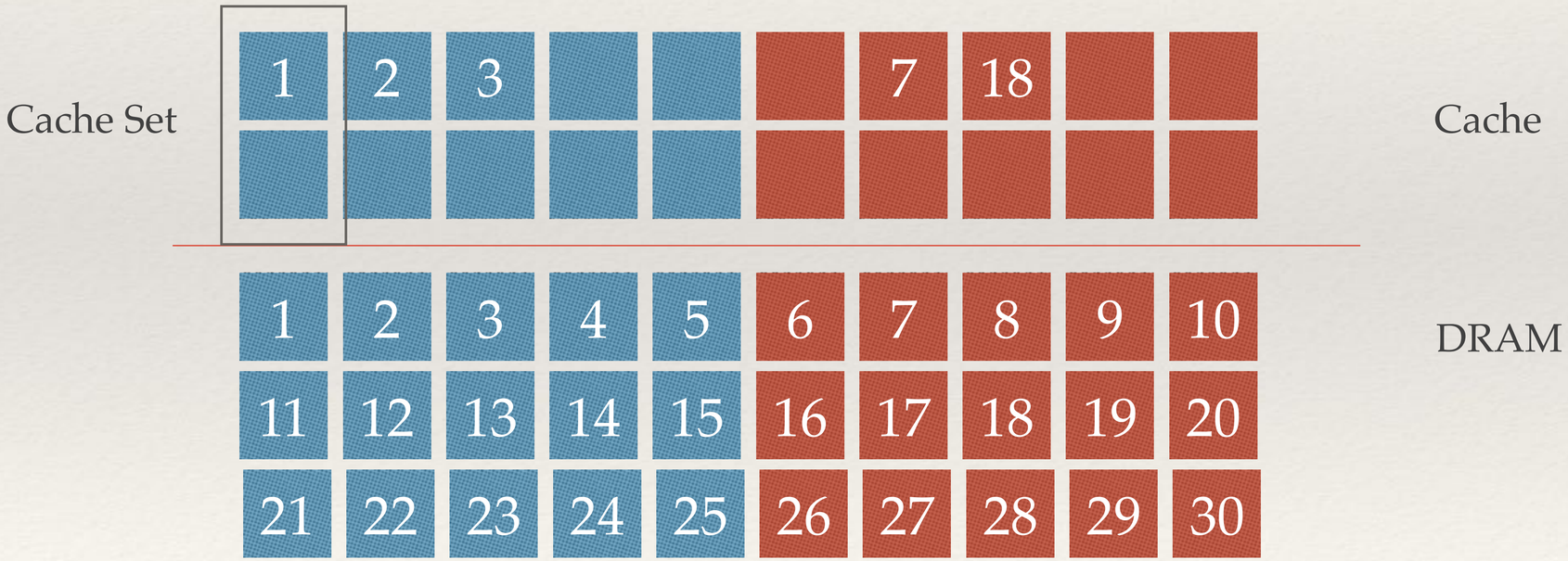
Tiny Cache Example

- 2-way cache, 5 sets per page, showing 2 colors



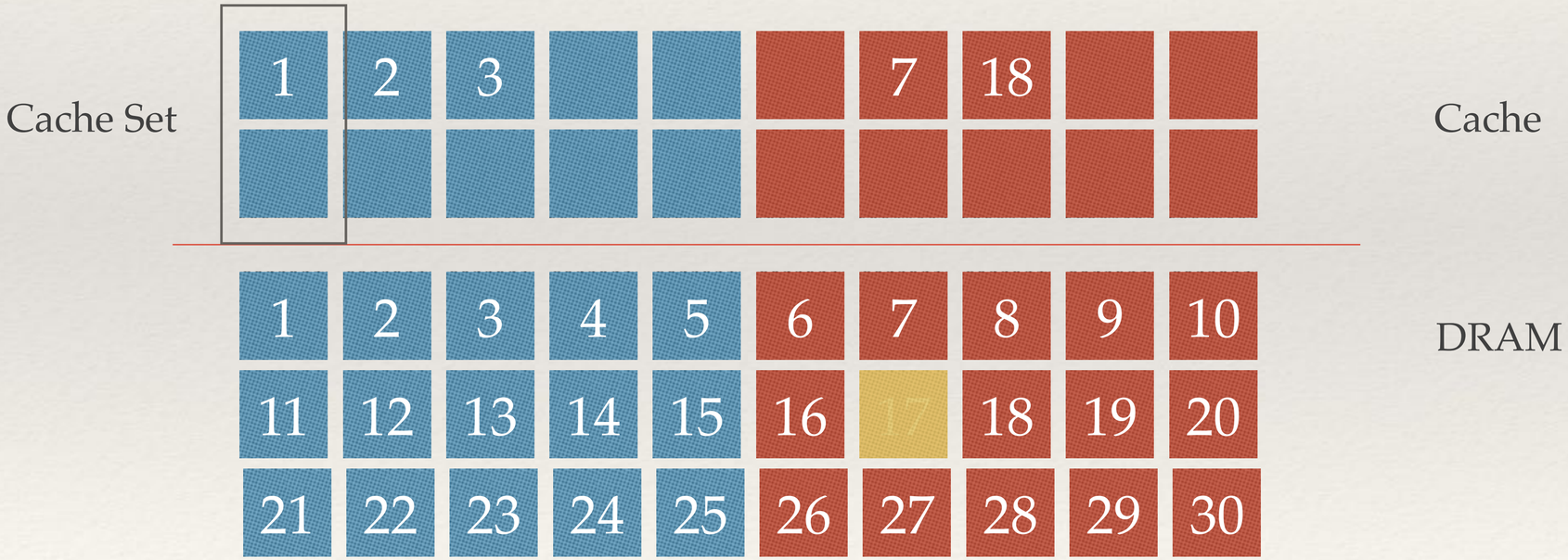
Tiny Cache Example

- 2-way cache, 5 sets per page, showing 2 colors



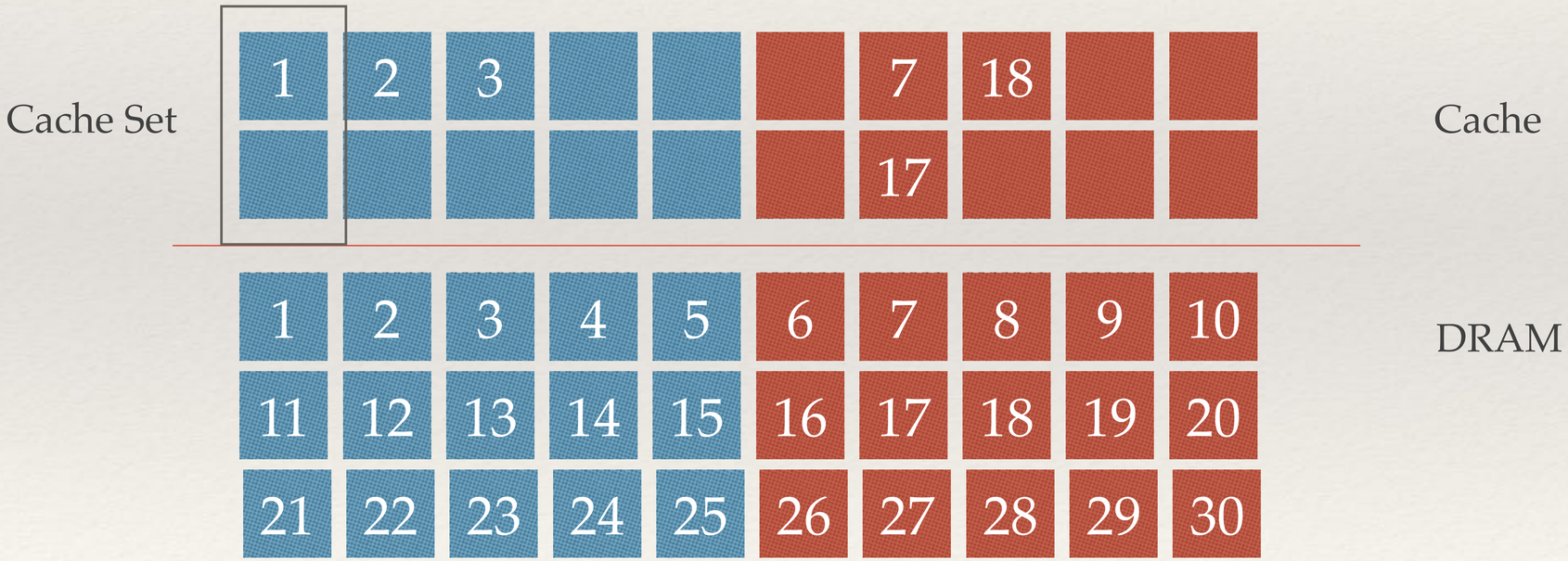
Tiny Cache Example

- 2-way cache, 5 sets per page, showing 2 colors



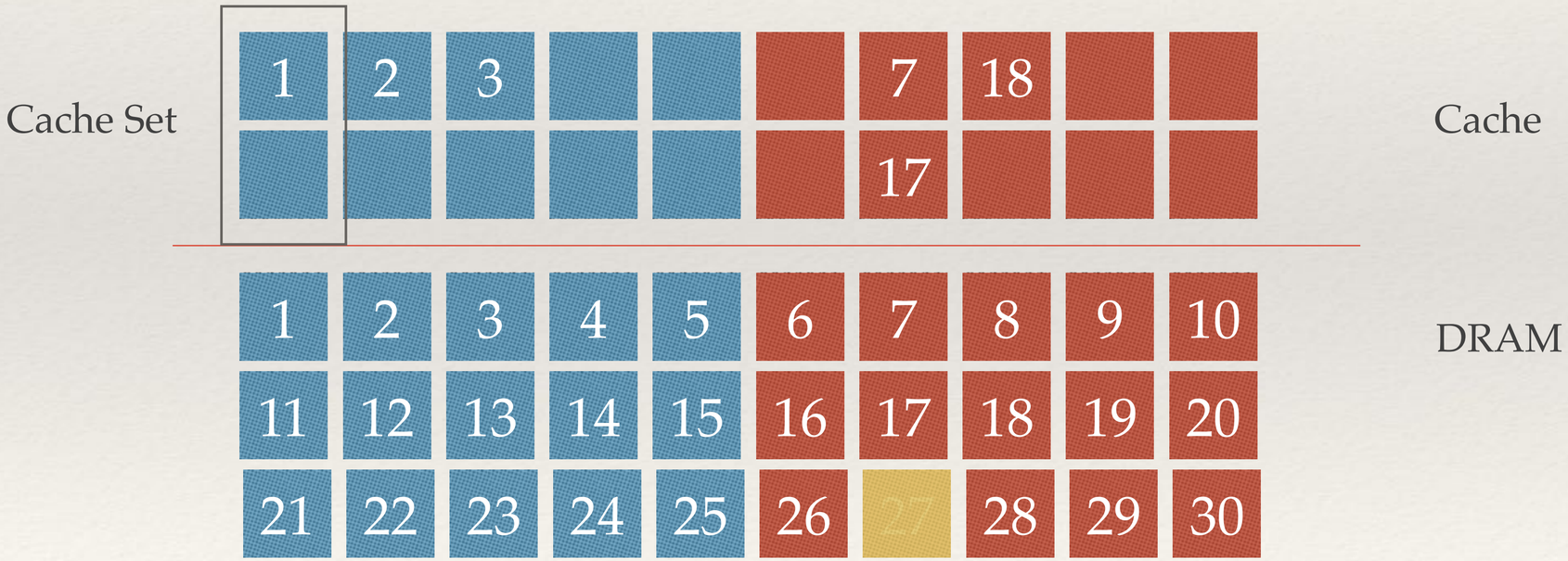
Tiny Cache Example

- 2-way cache, 5 sets per page, showing 2 colors



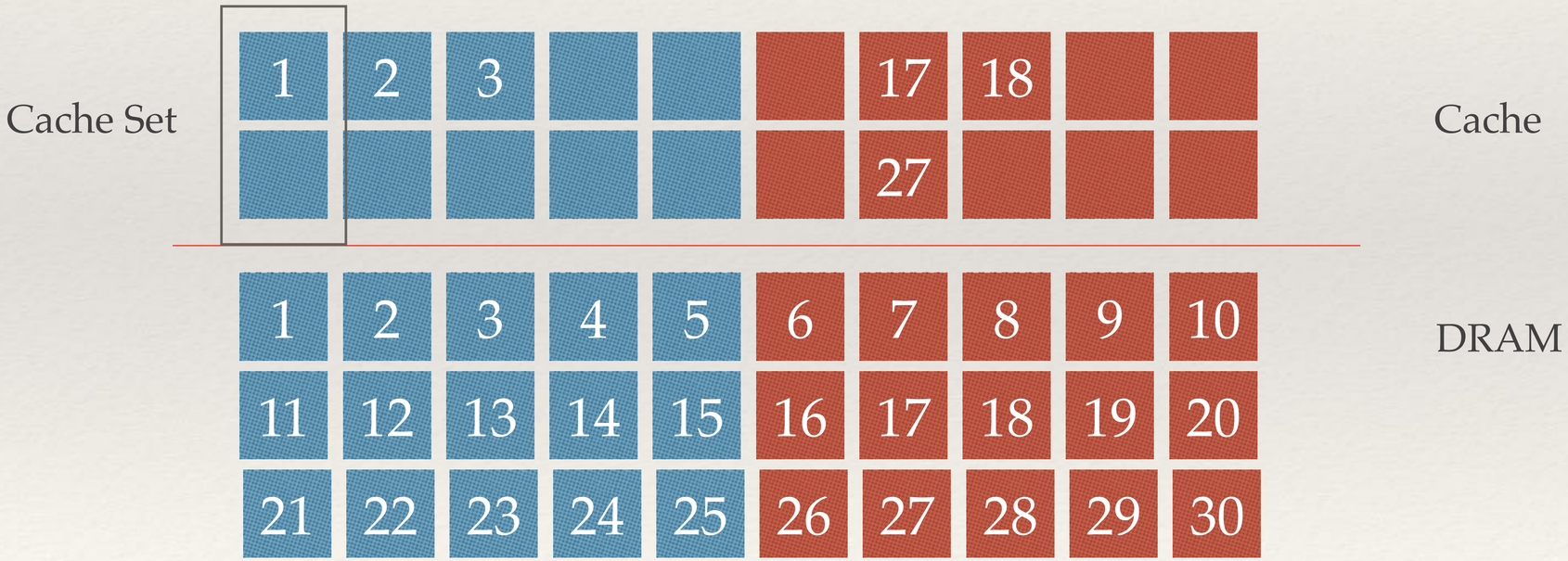
Tiny Cache Example

- 2-way cache, 5 sets per page, showing 2 colors



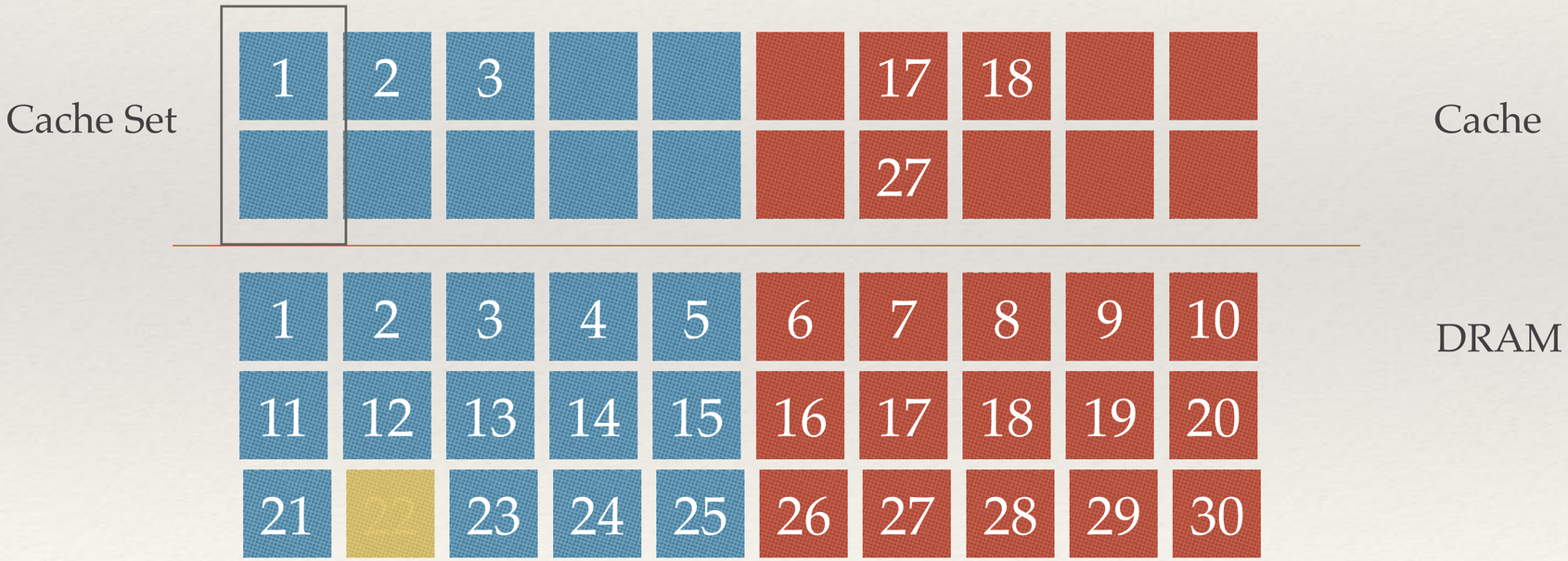
Tiny Cache Example

- 2-way cache, 5 sets per page, showing 2 colors



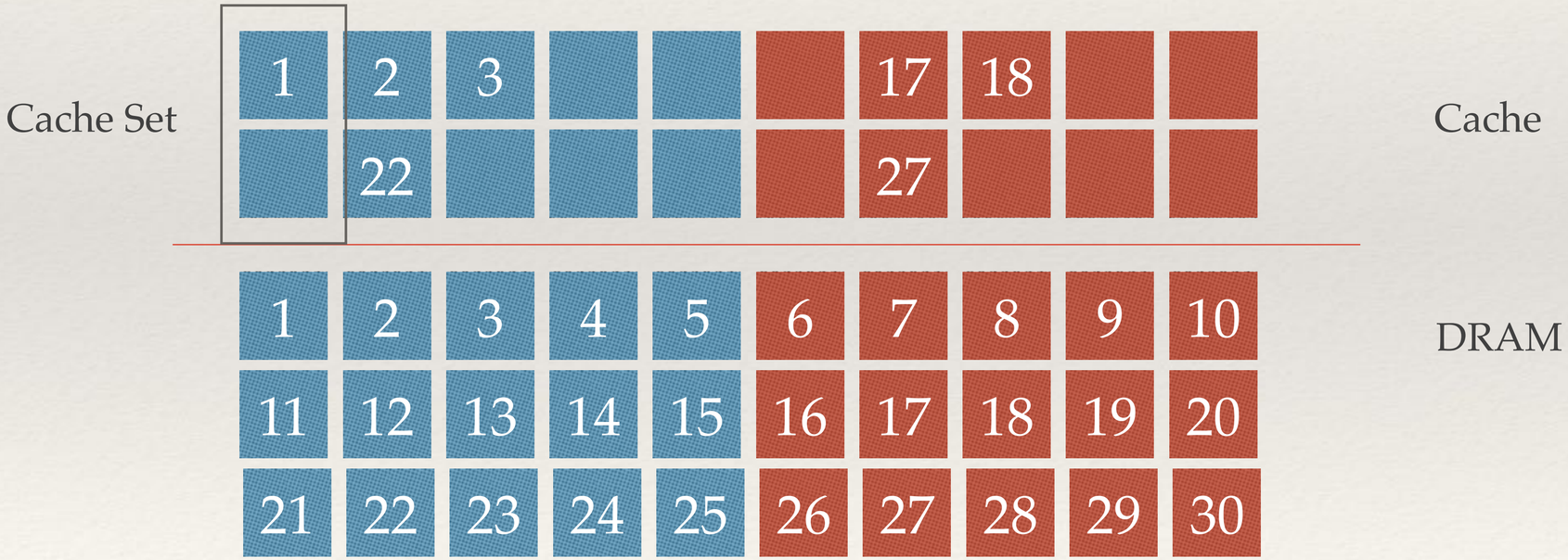
Tiny Cache Example

- 2-way cache, 5 sets per page, showing 2 colors



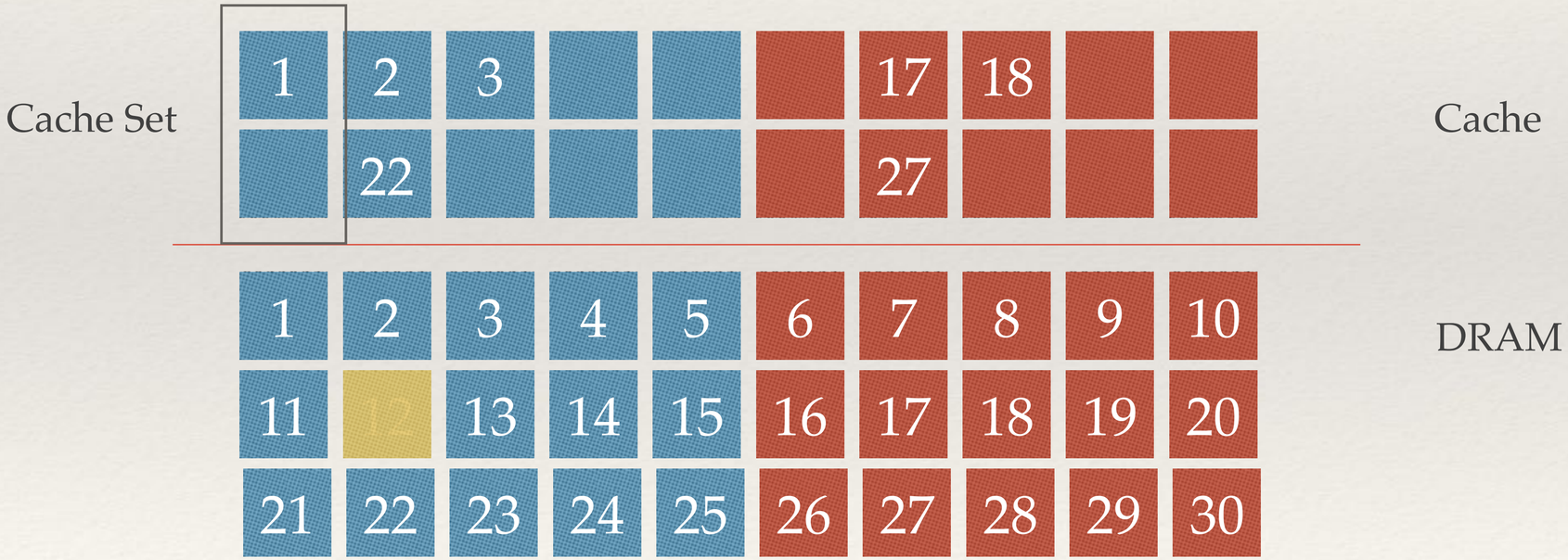
Tiny Cache Example

- 2-way cache, 5 sets per page, showing 2 colors



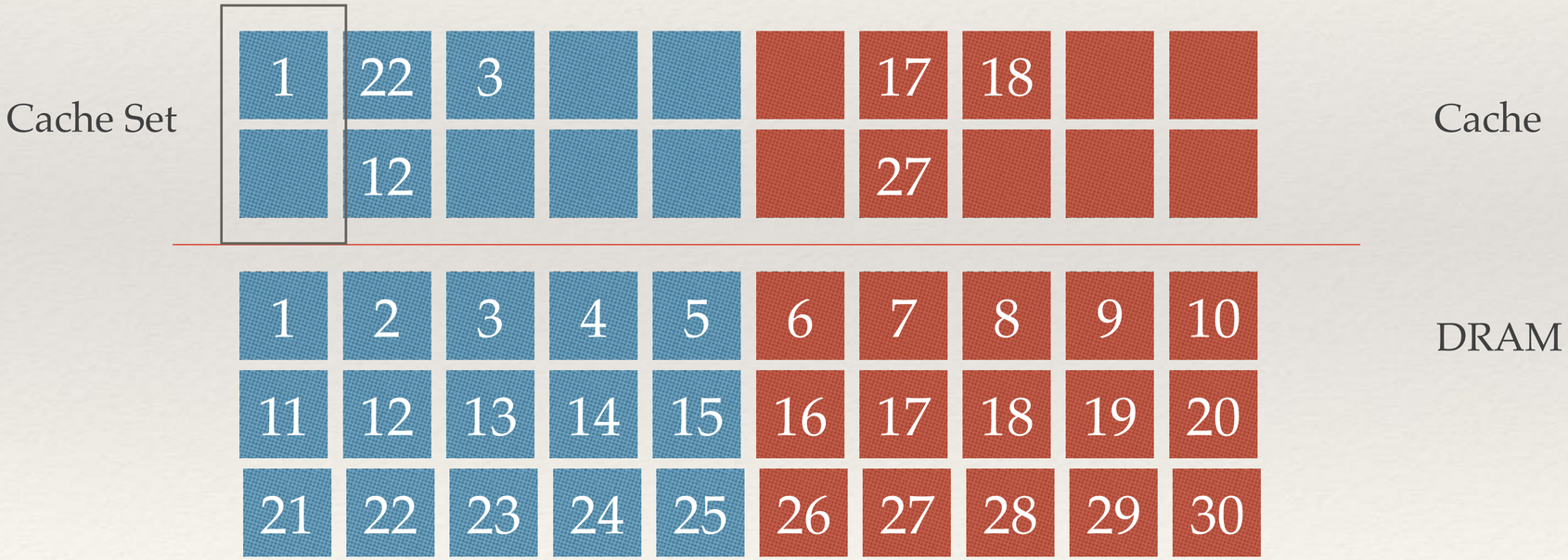
Tiny Cache Example

- 2-way cache, 5 sets per page, showing 2 colors



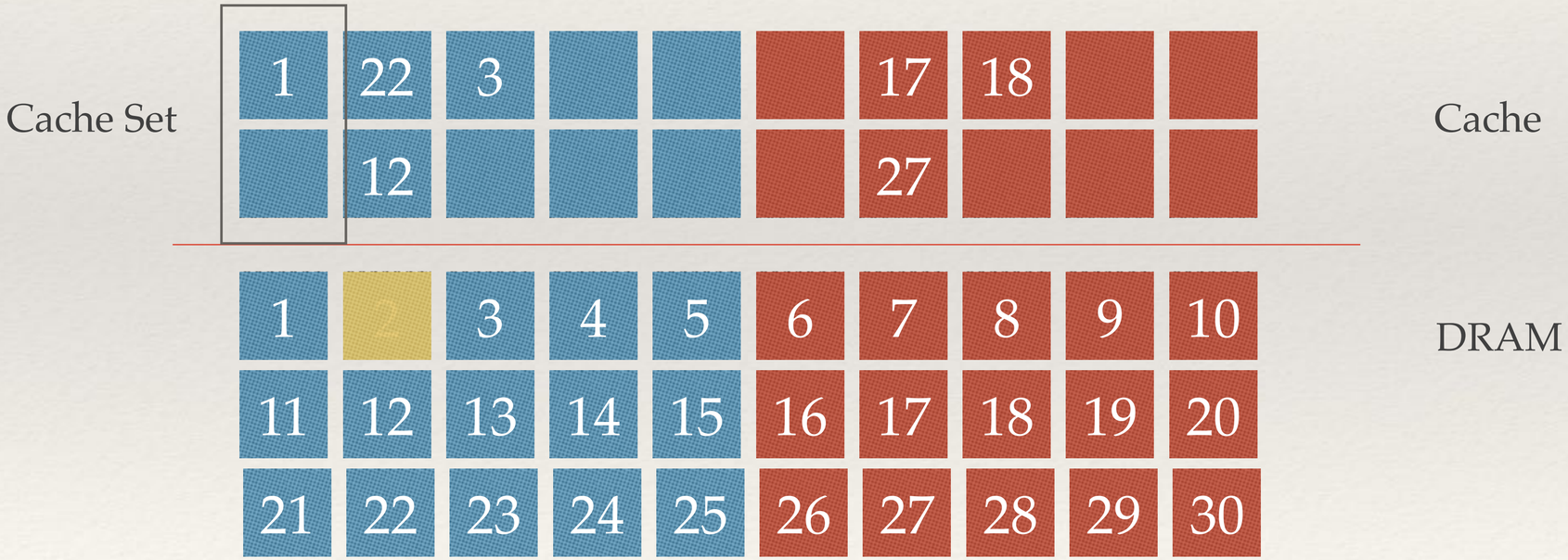
Tiny Cache Example

- 2-way cache, 5 sets per page, showing 2 colors



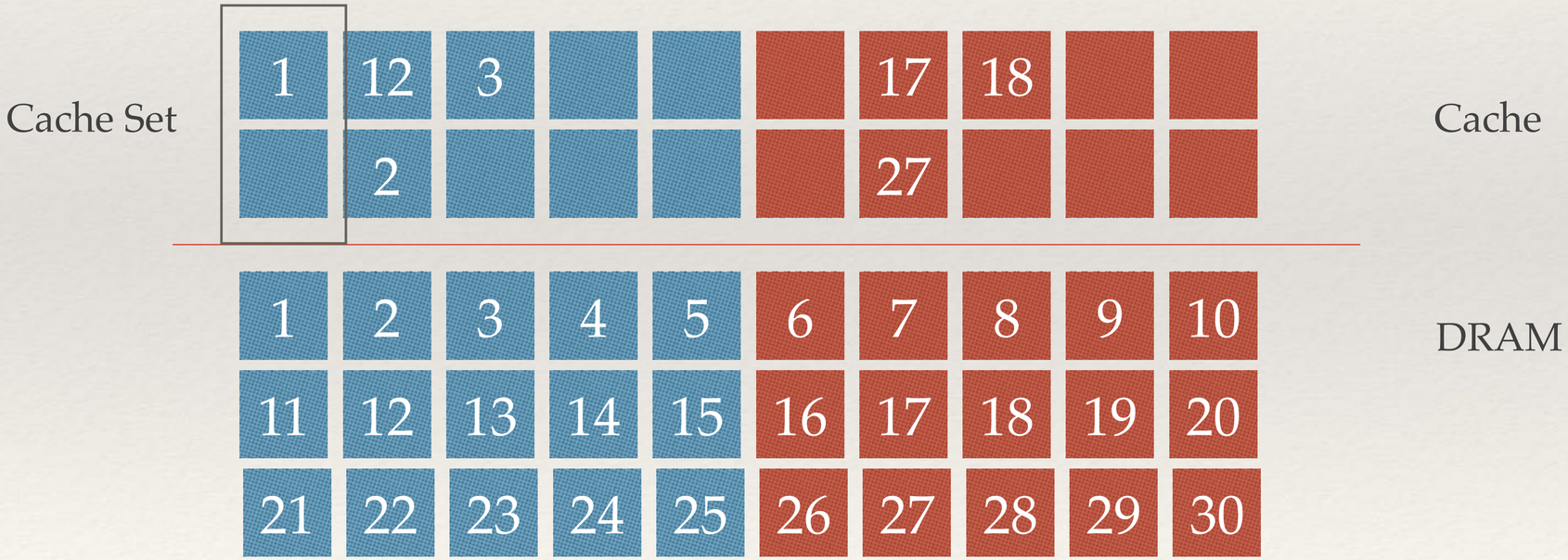
Tiny Cache Example

- 2-way cache, 5 sets per page, showing 2 colors



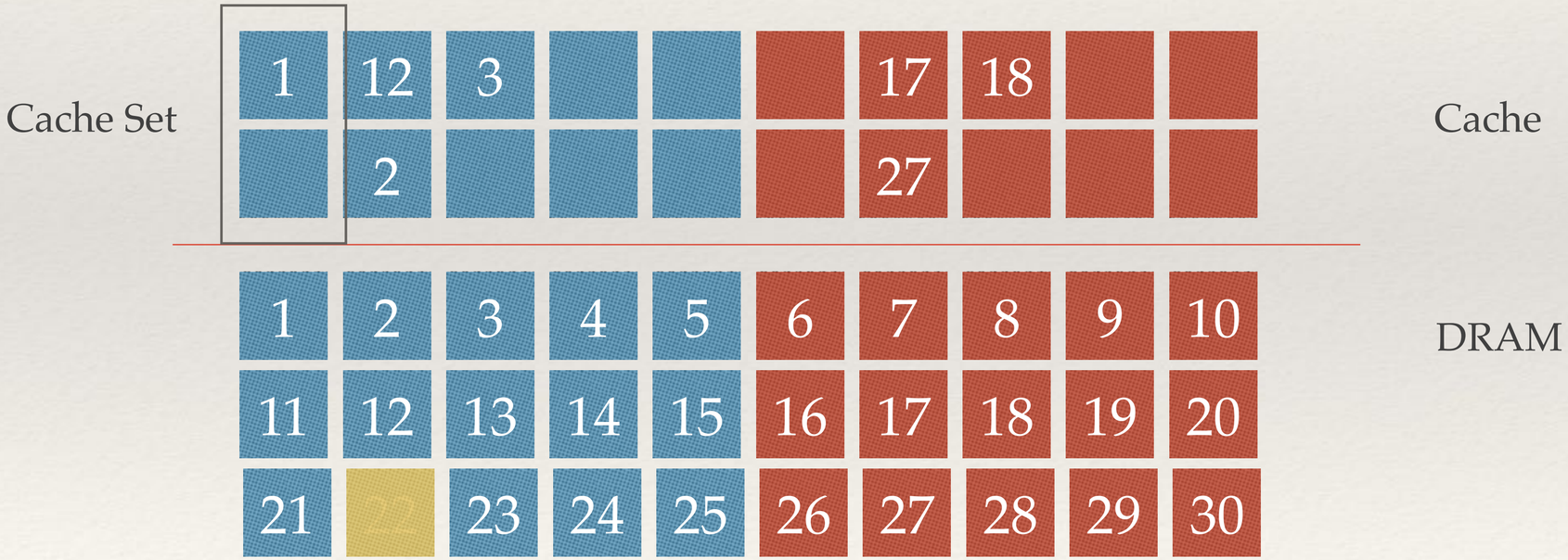
Tiny Cache Example

- 2-way cache, 5 sets per page, showing 2 colors



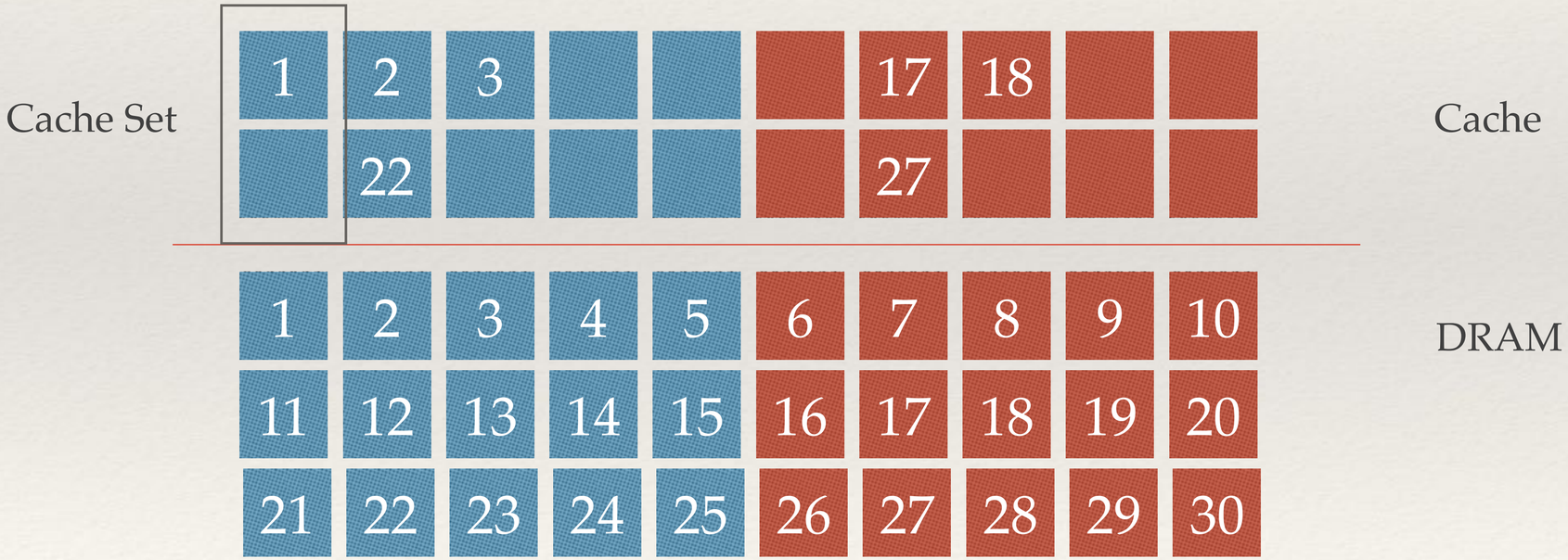
Tiny Cache Example

- 2-way cache, 5 sets per page, showing 2 colors



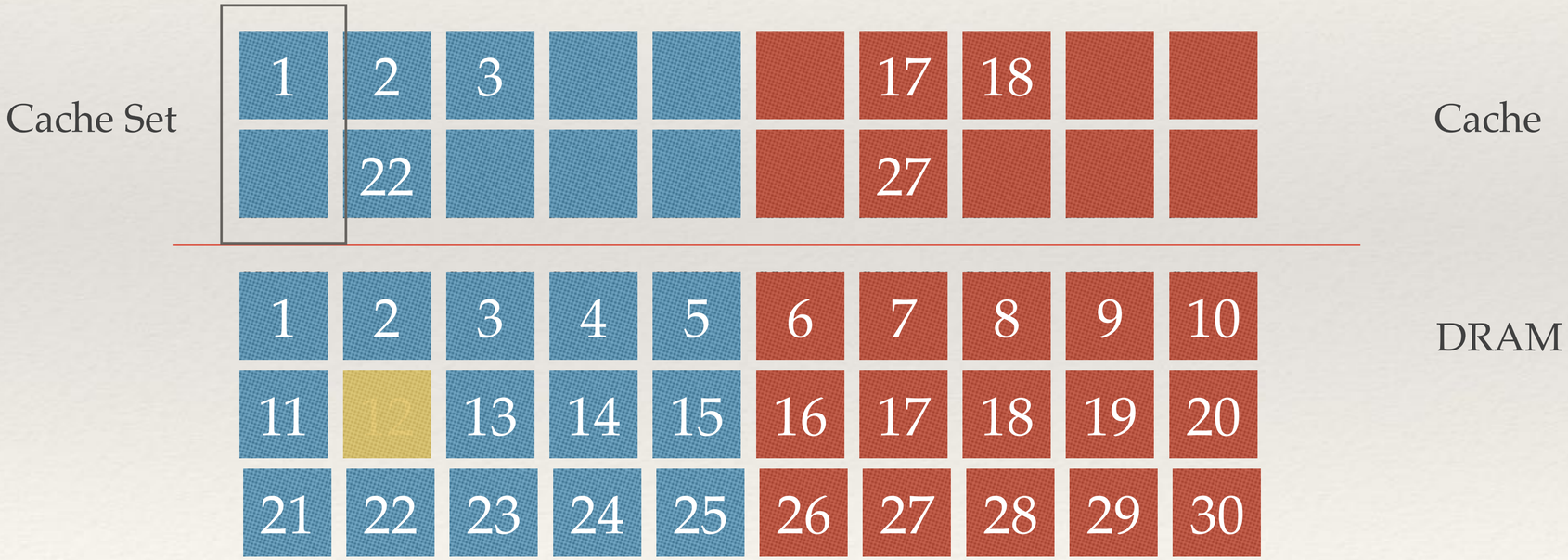
Tiny Cache Example

- 2-way cache, 5 sets per page, showing 2 colors



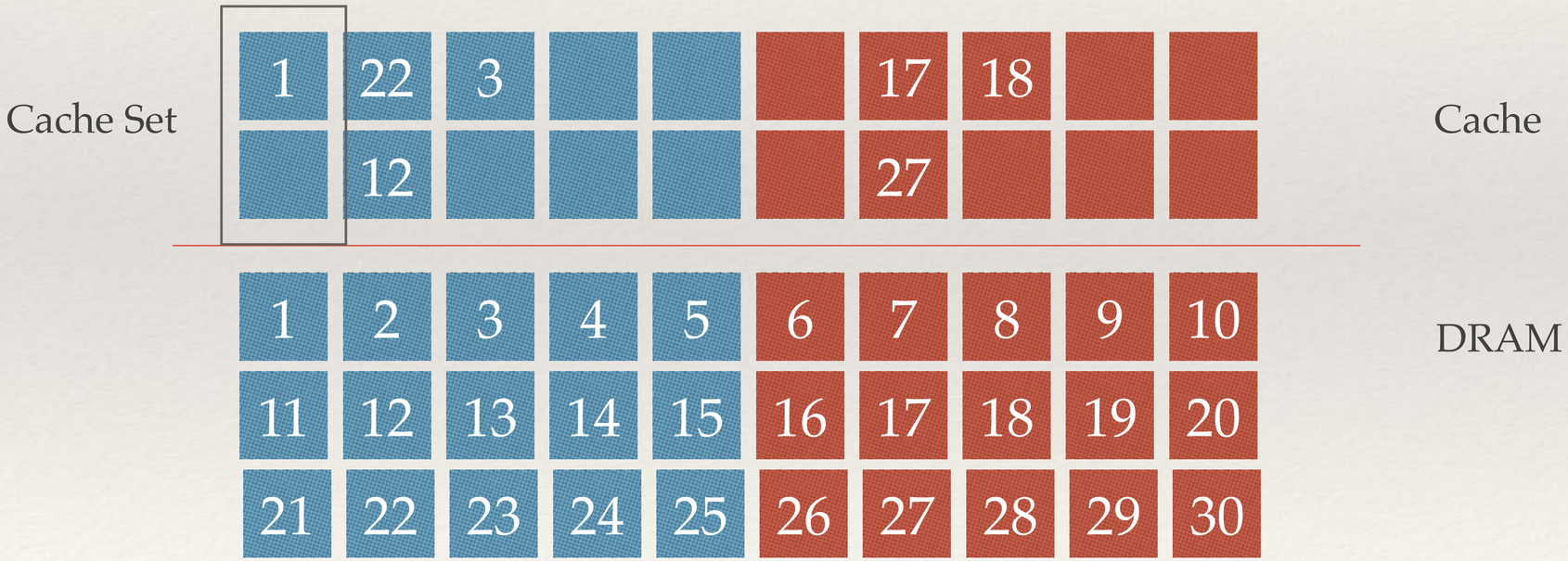
Tiny Cache Example

- 2-way cache, 5 sets per page, showing 2 colors



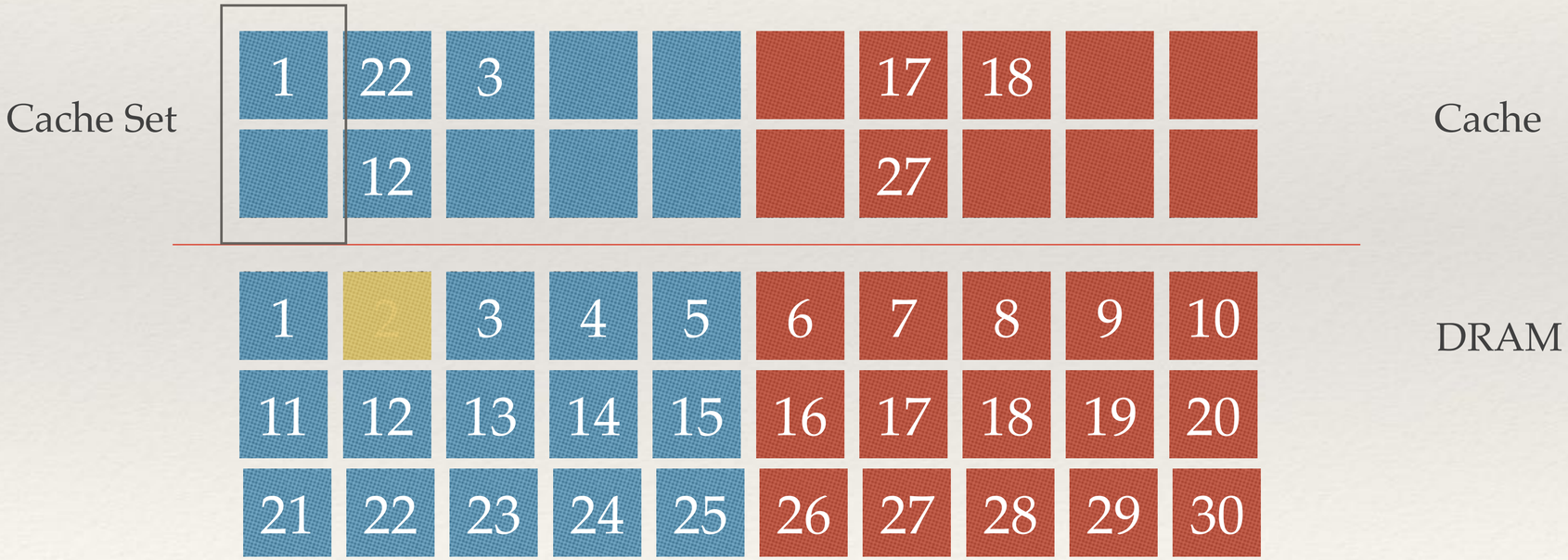
Tiny Cache Example

- 2-way cache, 5 sets per page, showing 2 colors



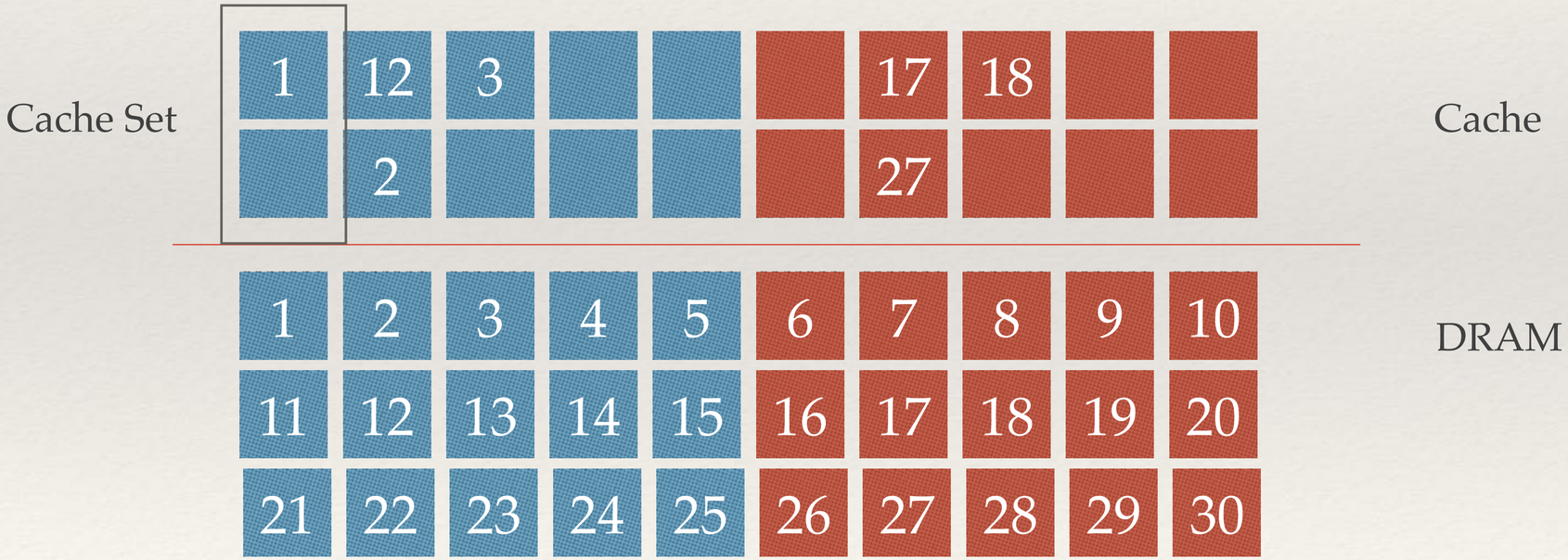
Tiny Cache Example

- 2-way cache, 5 sets per page, showing 2 colors



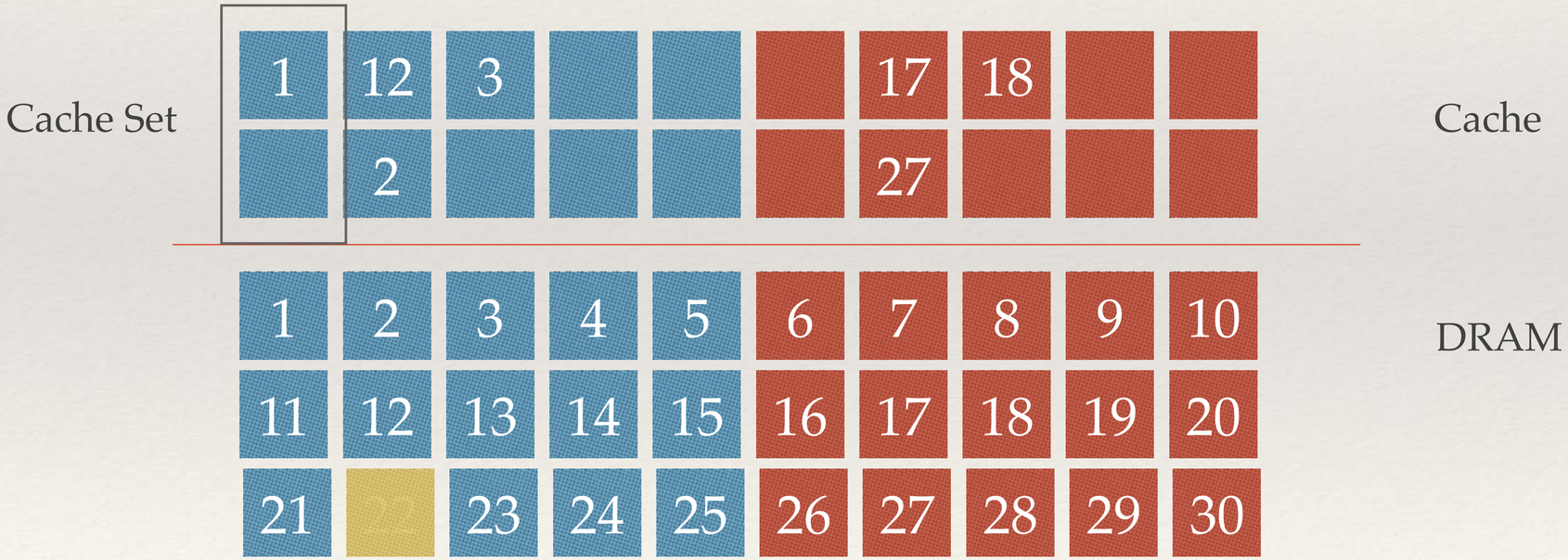
Tiny Cache Example

- 2-way cache, 5 sets per page, showing 2 colors



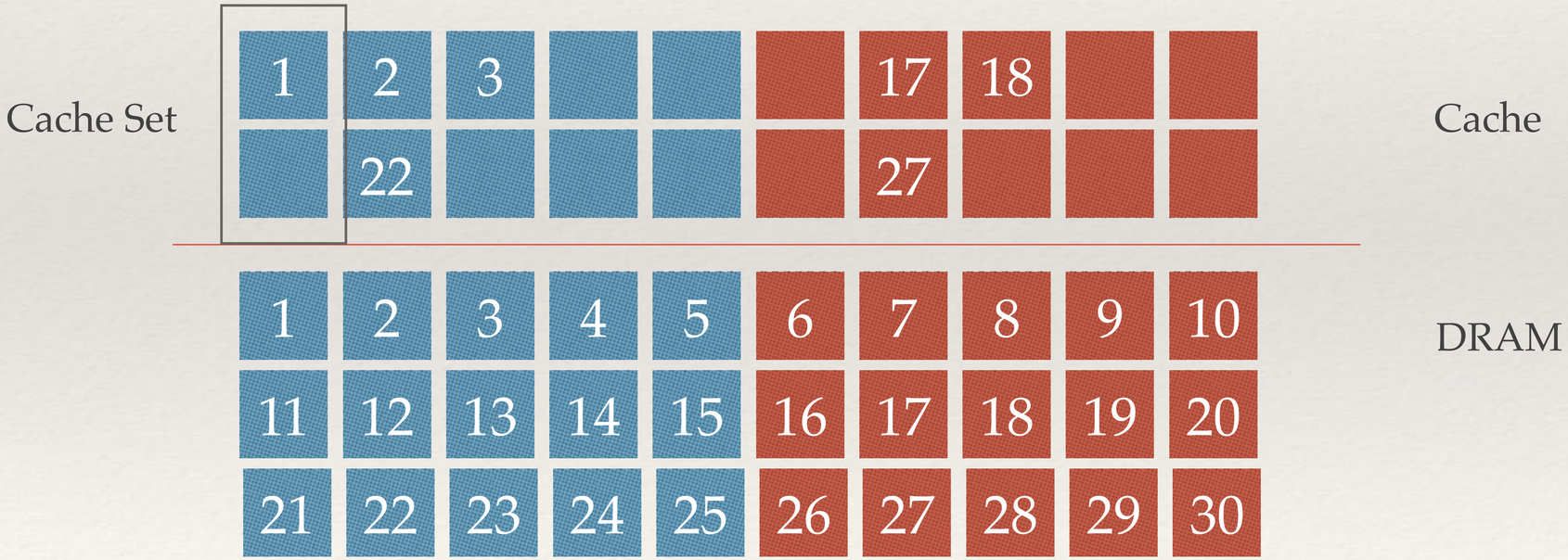
Tiny Cache Example

- 2-way cache, 5 sets per page, showing 2 colors



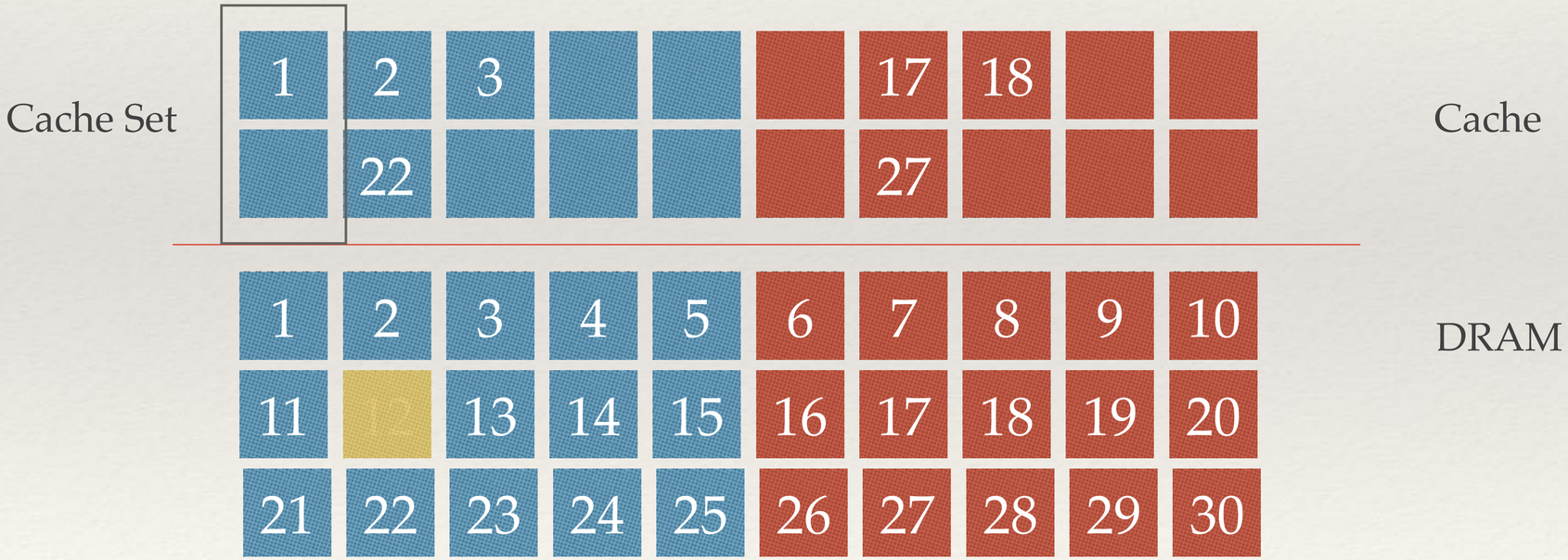
Tiny Cache Example

- 2-way cache, 5 sets per page, showing 2 colors



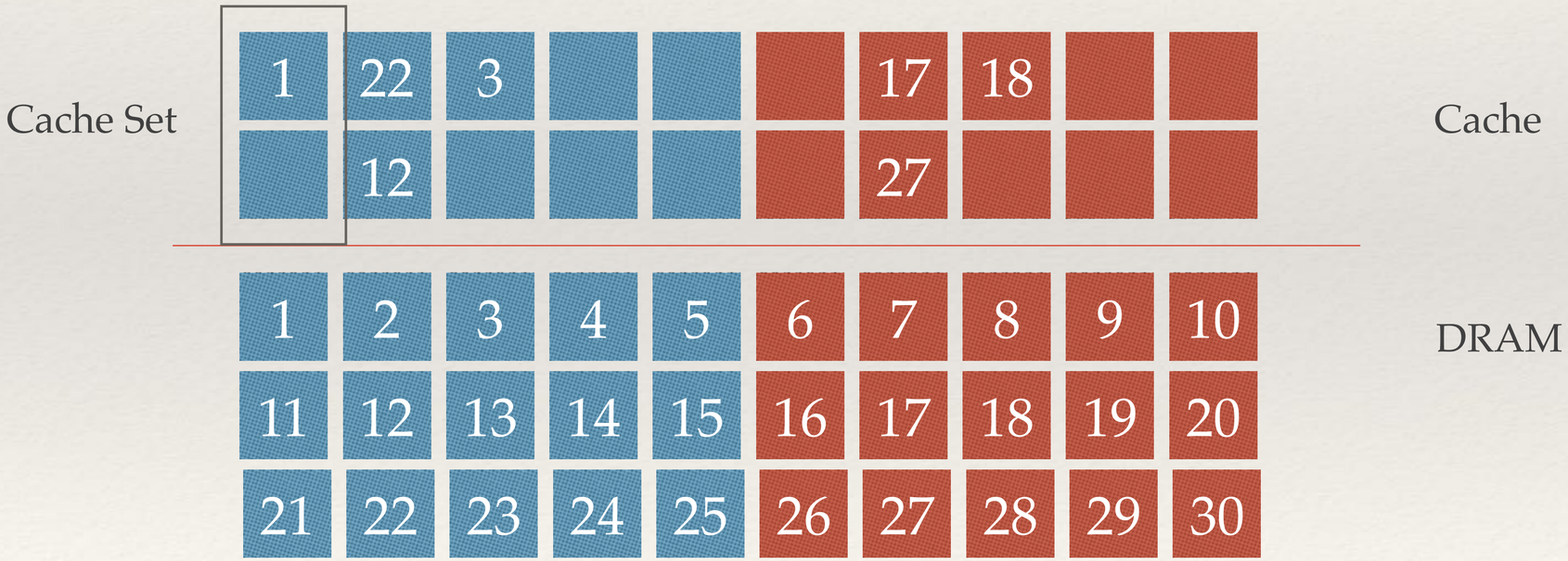
Tiny Cache Example

- 2-way cache, 5 sets per page, showing 2 colors



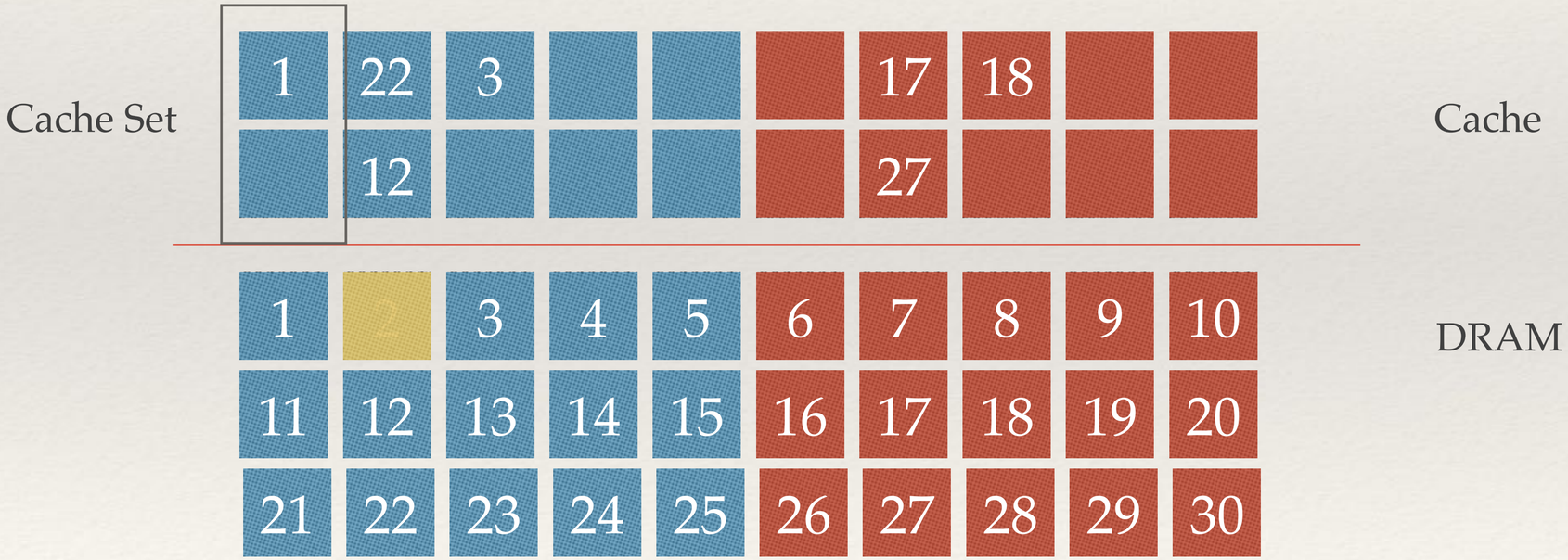
Tiny Cache Example

- 2-way cache, 5 sets per page, showing 2 colors



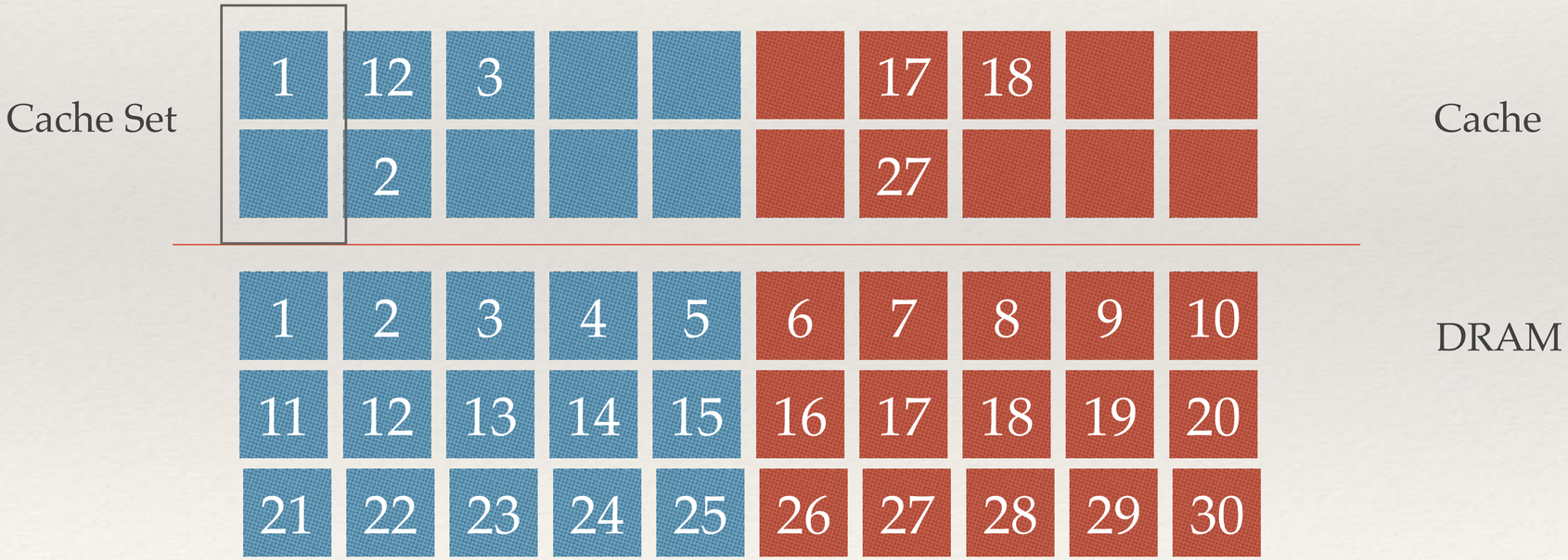
Tiny Cache Example

- 2-way cache, 5 sets per page, showing 2 colors

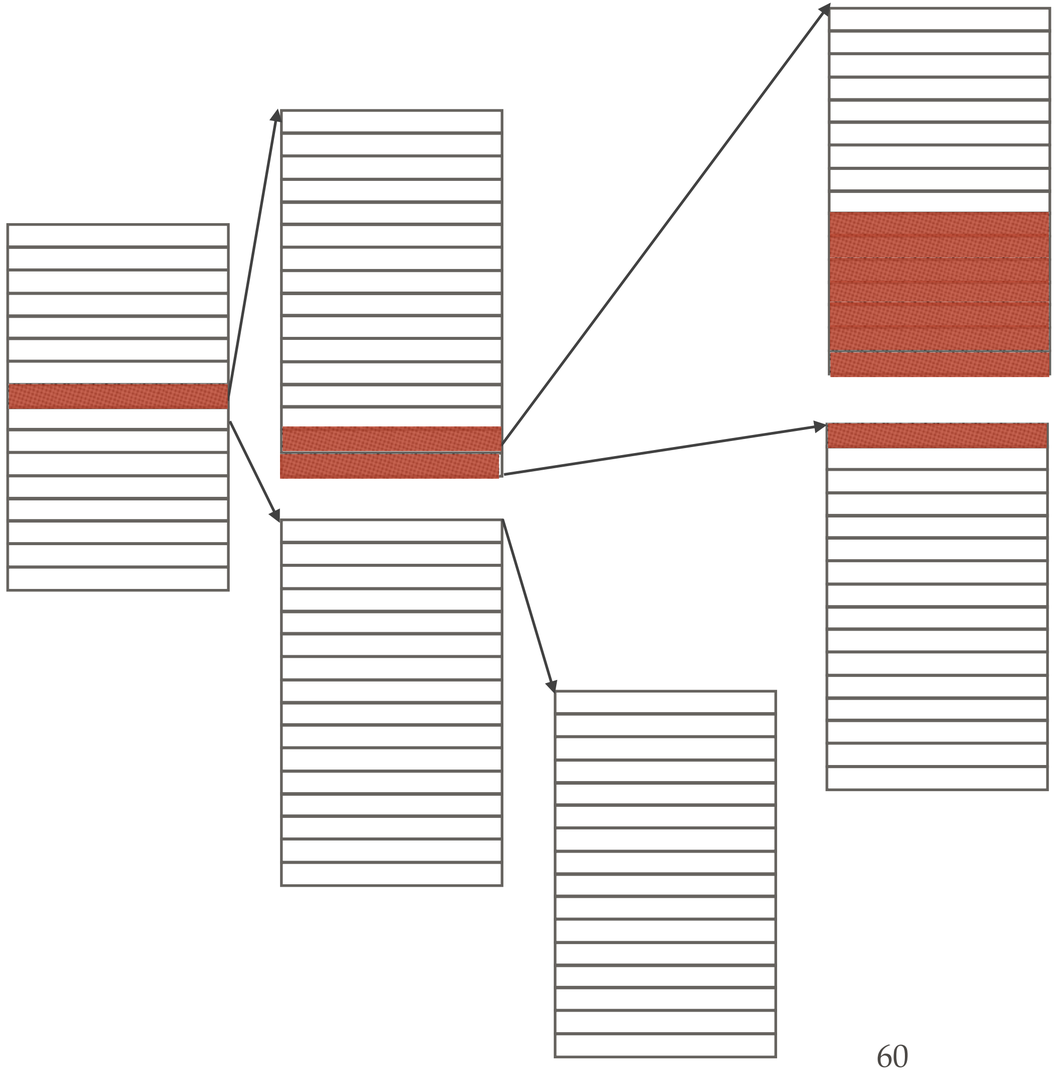


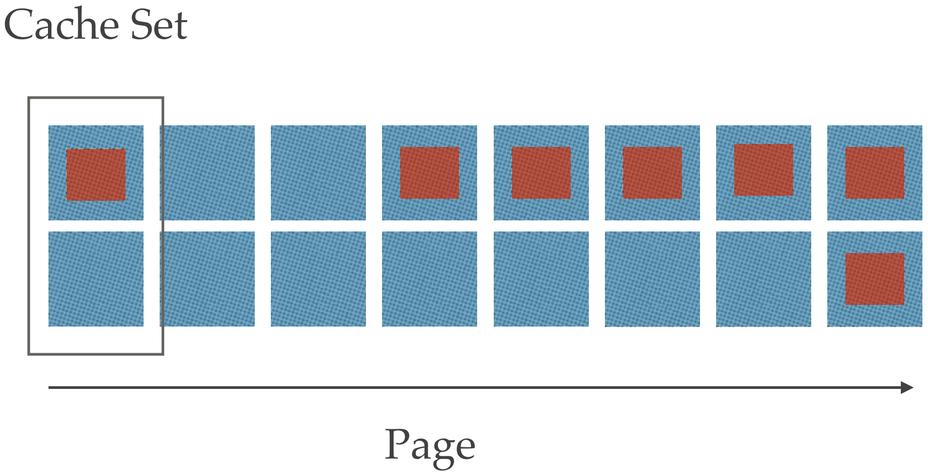
Tiny Cache Example

- 2-way cache, 5 sets per page, showing 2 colors
- Eviction sets follow page offsets



Big picture: cached page tables

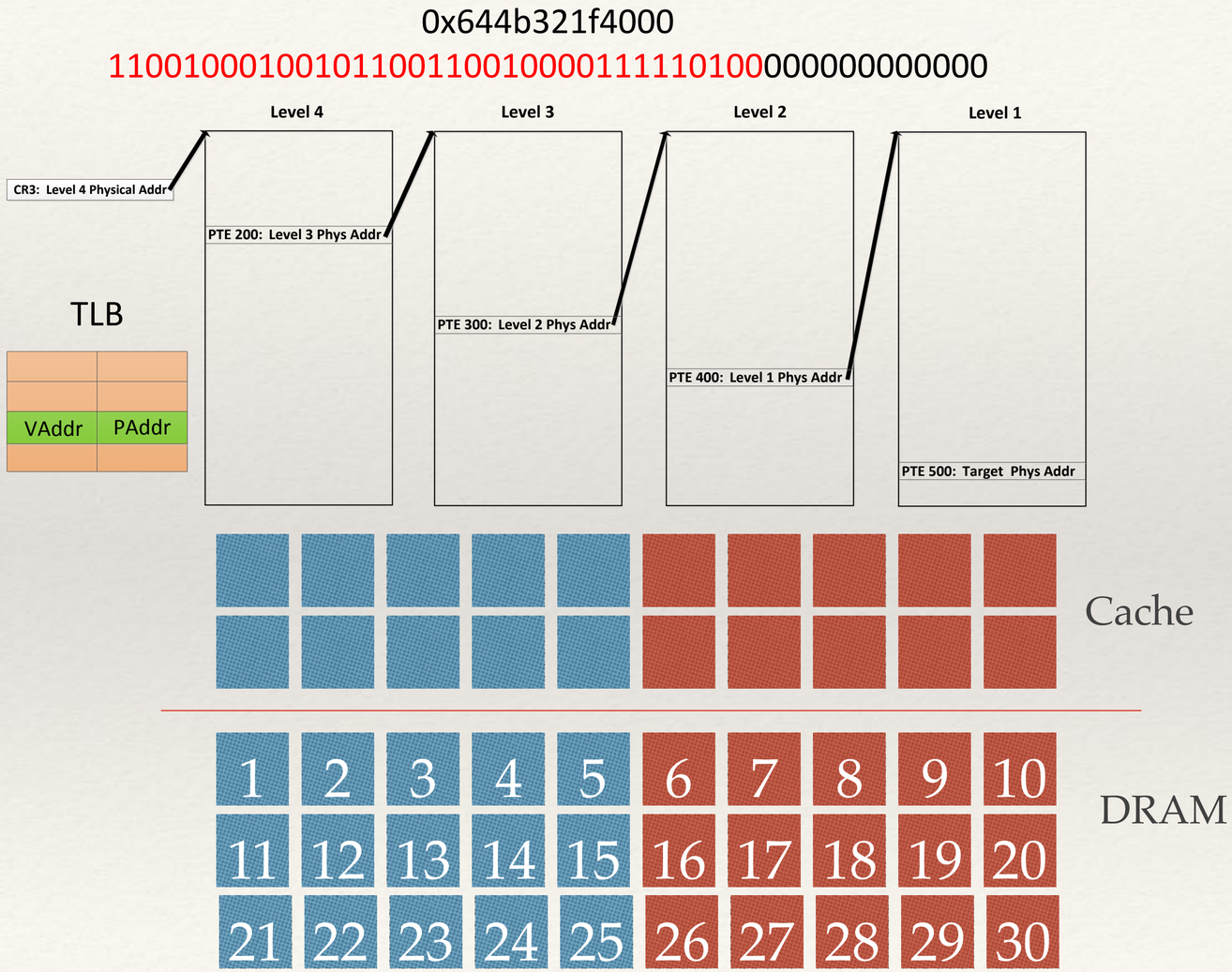




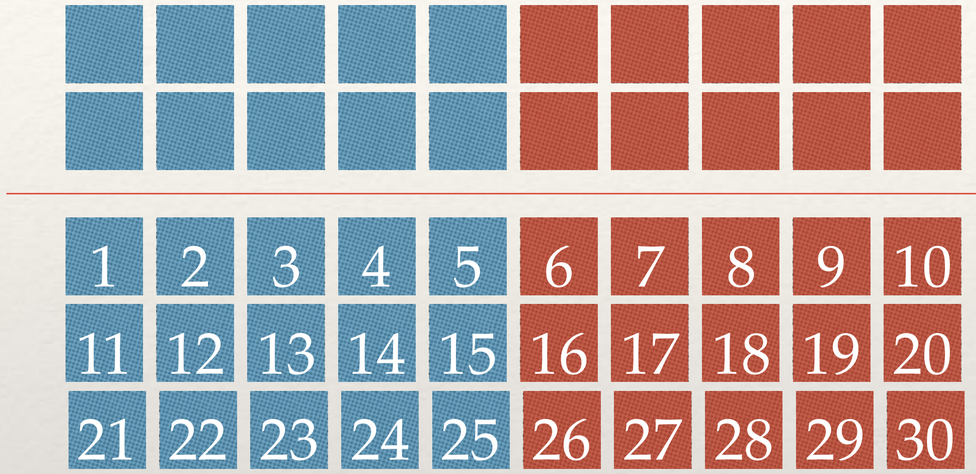
EVICT+TIME

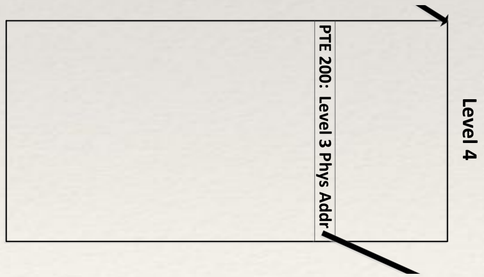
- Flush TLB, forcing pagetable walk
- 2x to measure cached lookup time
- Flush TLB
- Evict first cacheline
- Measure possibly-uncached lookup time
- Find cacheline dependencies

EVICT+TIME in Cache



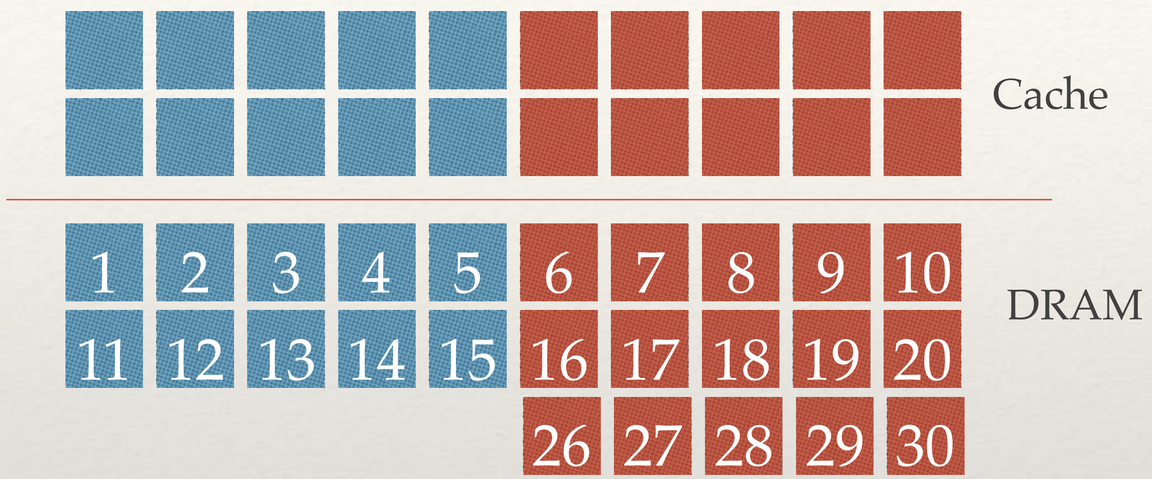
EVICT+TIME in Cache

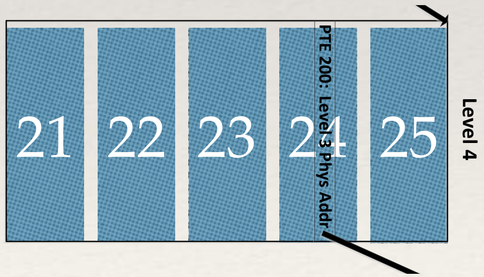




Pagetable in DRAM

EVICT+TIME in Cache

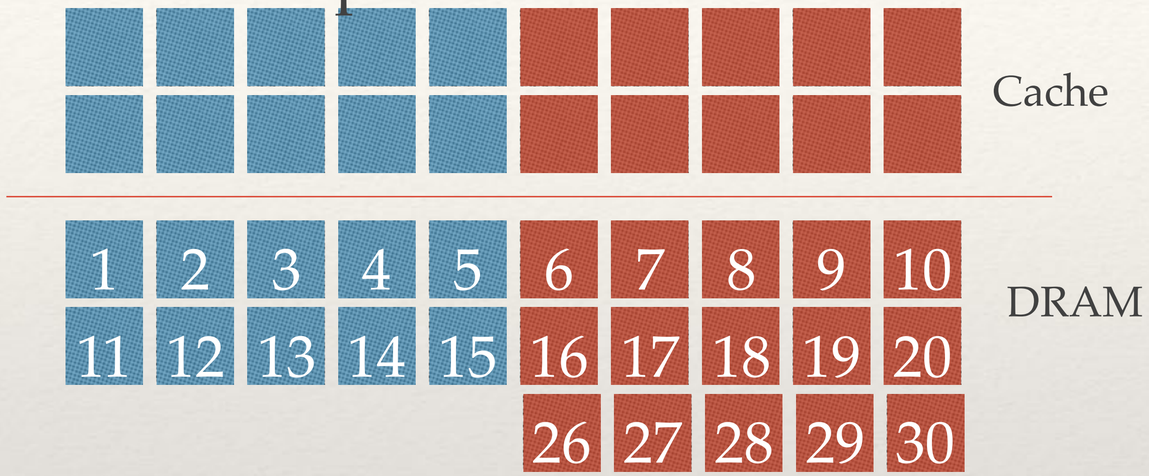


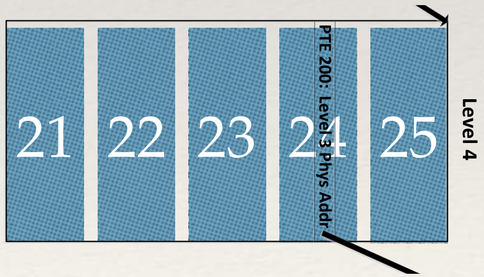


Pagetable in DRAM

EVICT+TIME in Cache

- Do address lookup

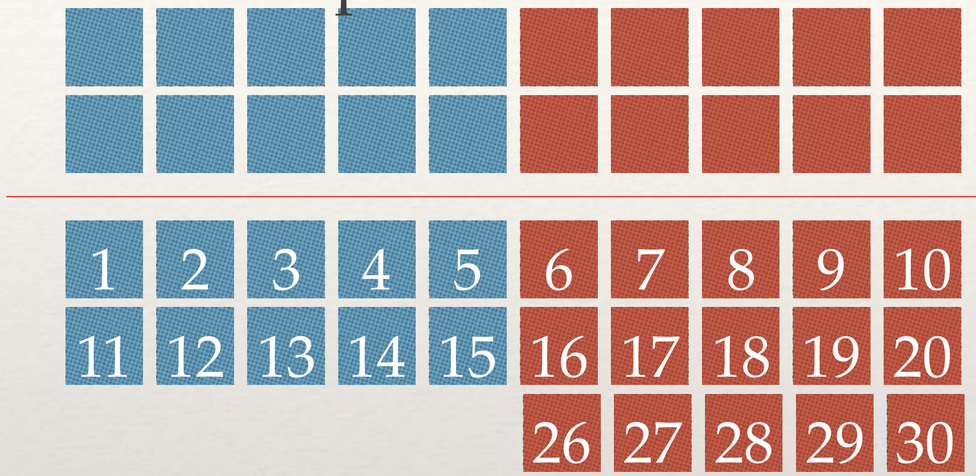


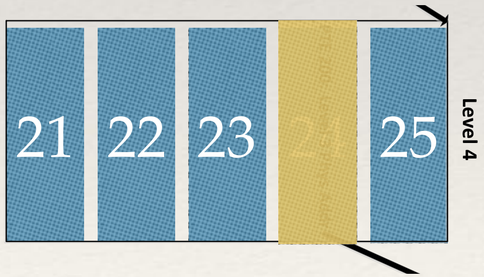


Pagetable in DRAM

EVICT+TIME in Cache

- Do address lookup

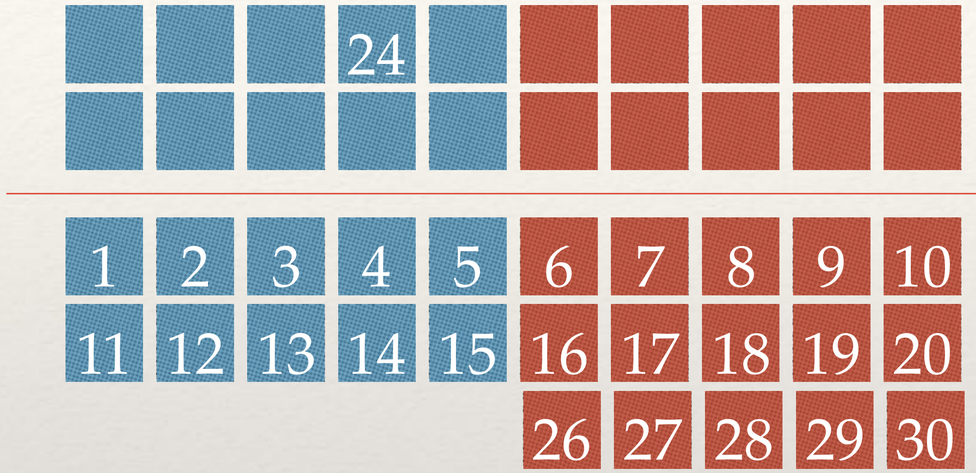


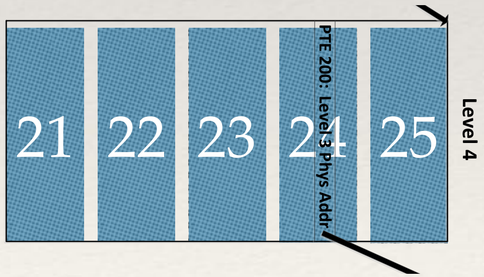


Pagetable in DRAM

EVICT+TIME in Cache

- It was uncached - slow

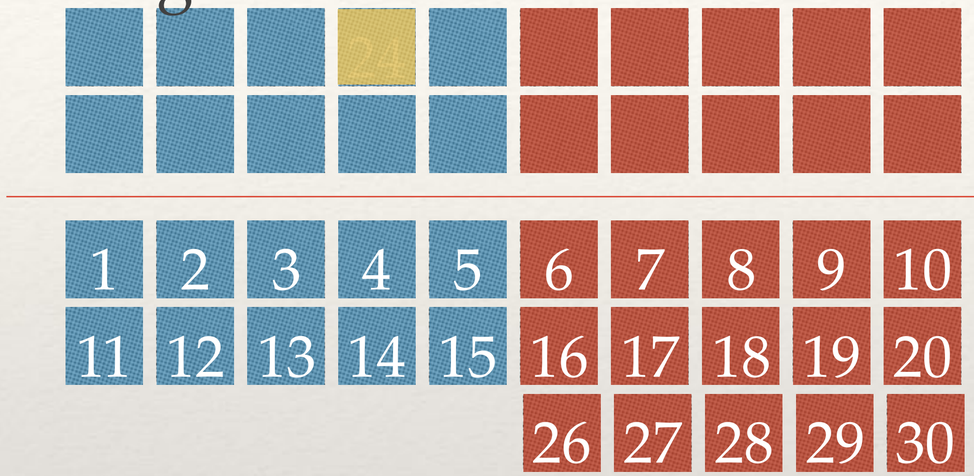


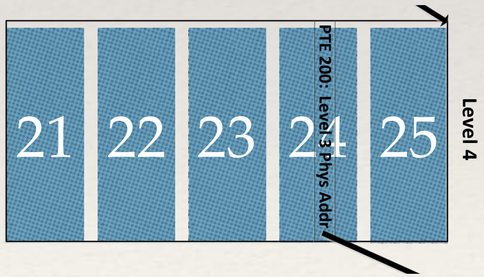


Pagetable in DRAM

EVICT+TIME in Cache

- Let's do it again

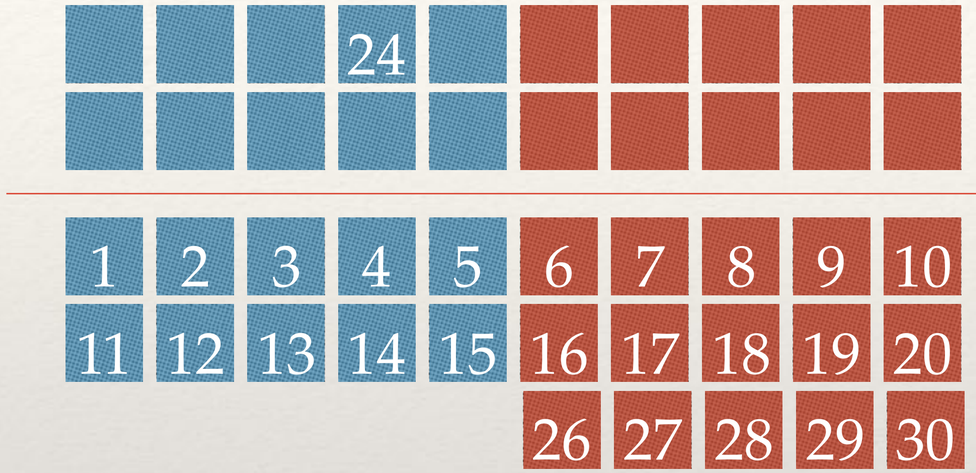


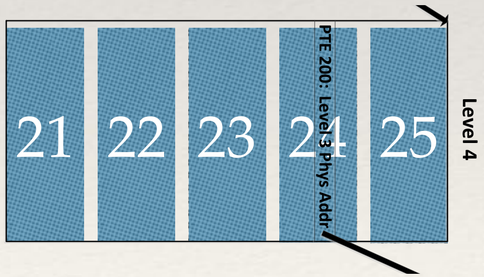


Pagetable in DRAM

EVICT+TIME in Cache

- It was cached - fast

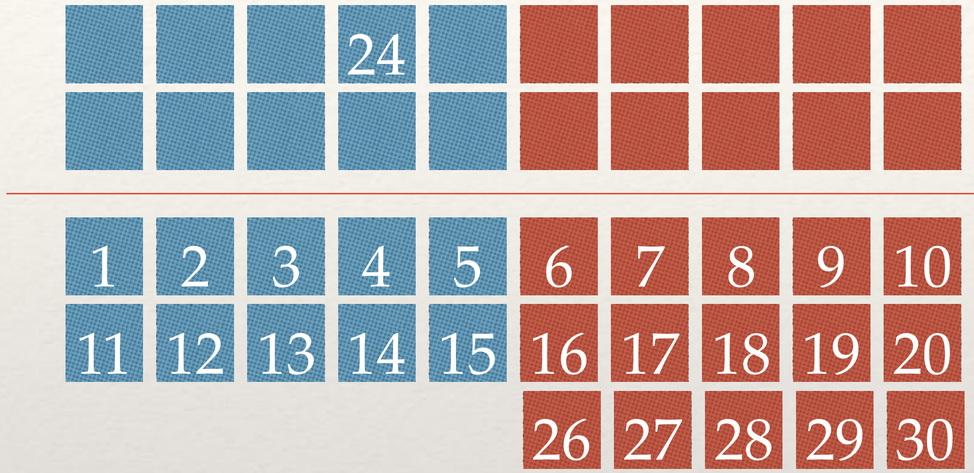


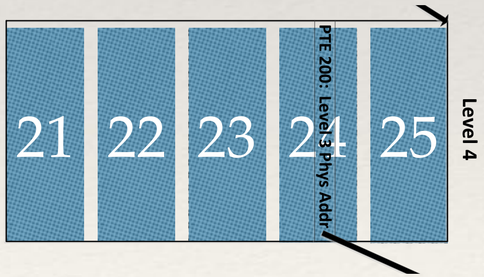


Pagetable in DRAM

EVICT+TIME in Cache

- Let's evict

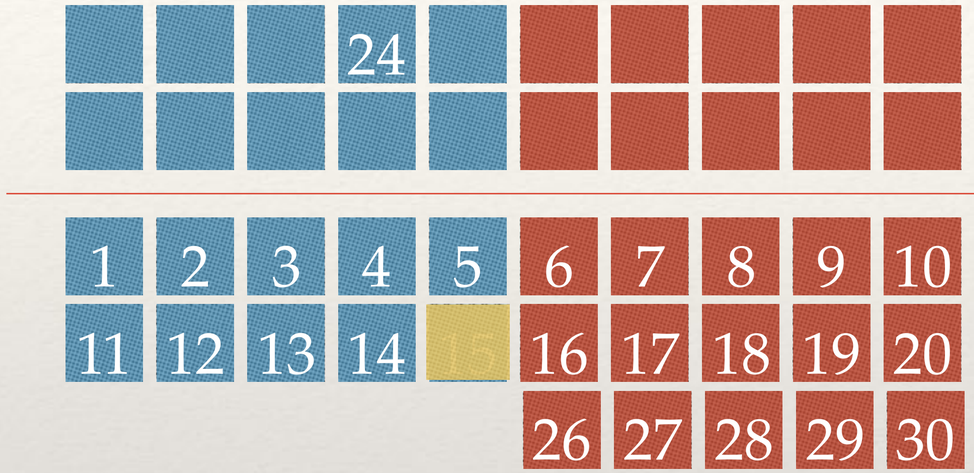


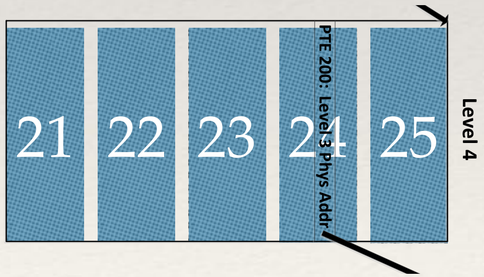


Pagetable in DRAM

EVICT+TIME in Cache

- Let's evict

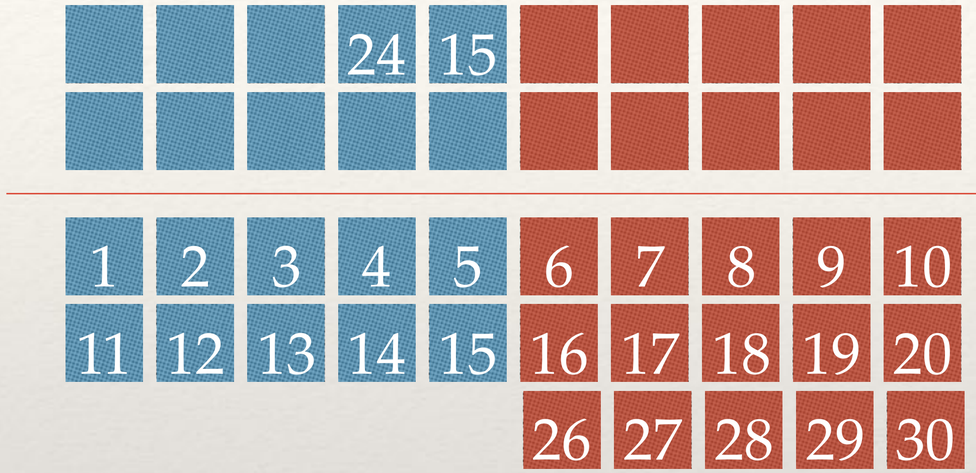


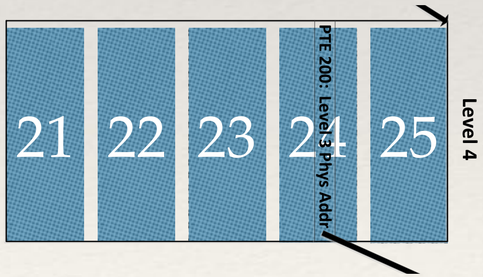


Pagetable in DRAM

EVICT+TIME in Cache

- Let's evict

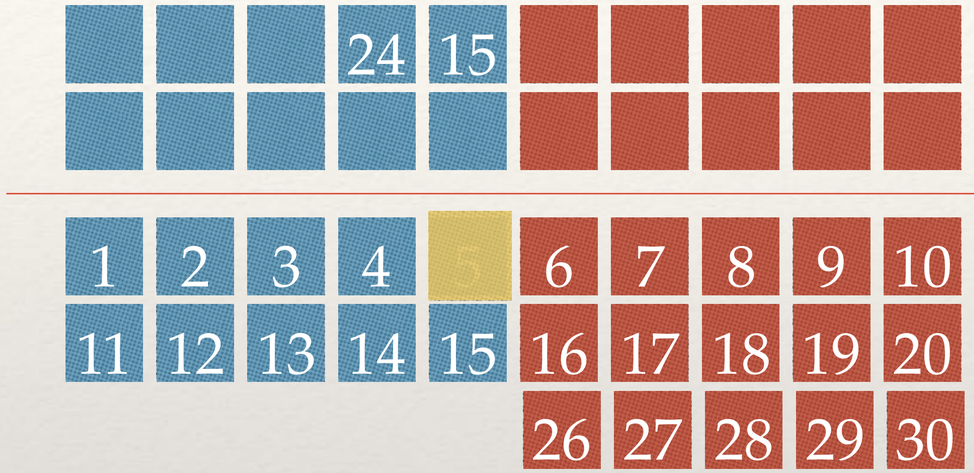


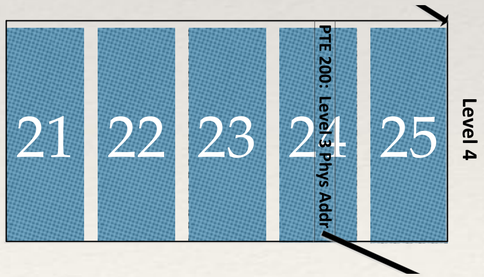


Pagetable in DRAM

EVICT+TIME in Cache

- Let's evict

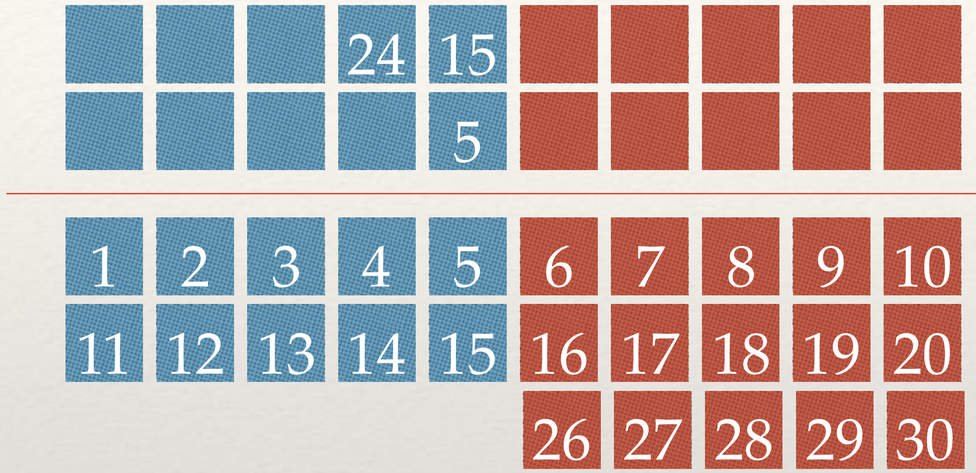


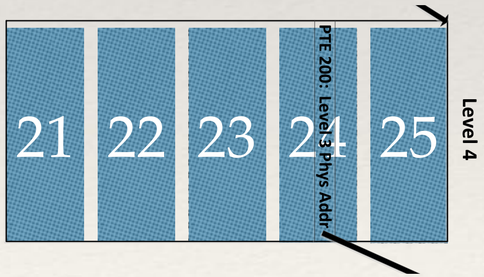


Pagetable in DRAM

EVICT+TIME in Cache

- Let's evict

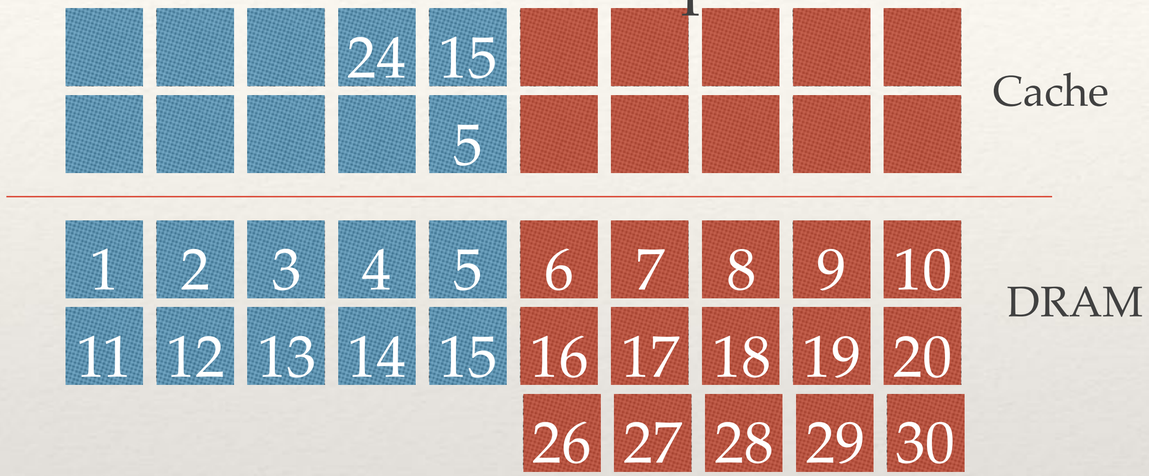


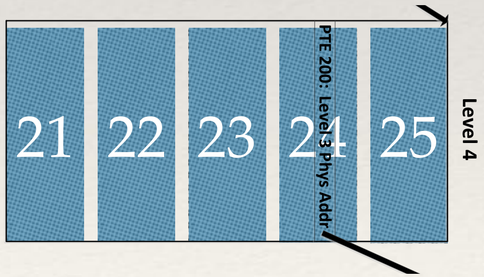


Pagetable in DRAM

EVICT+TIME in Cache

- Eviction done - let's do lookup

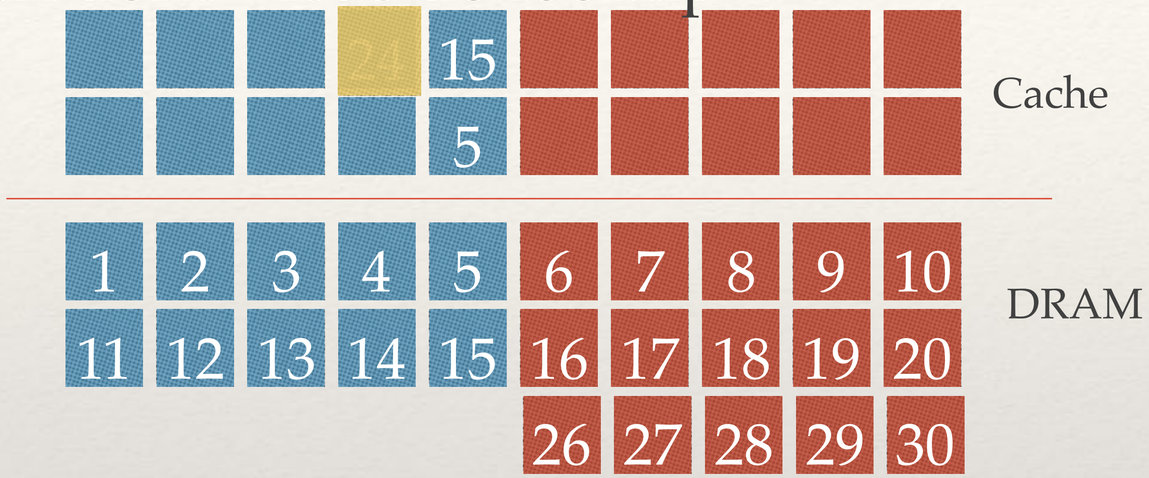


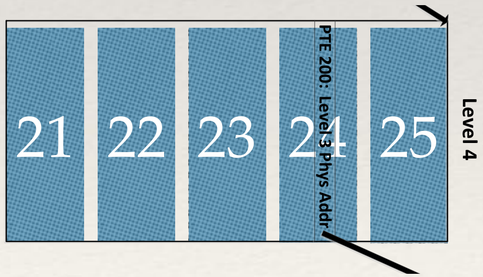


Pagetable in DRAM

EVICT+TIME in Cache

- Eviction done - let's do lookup

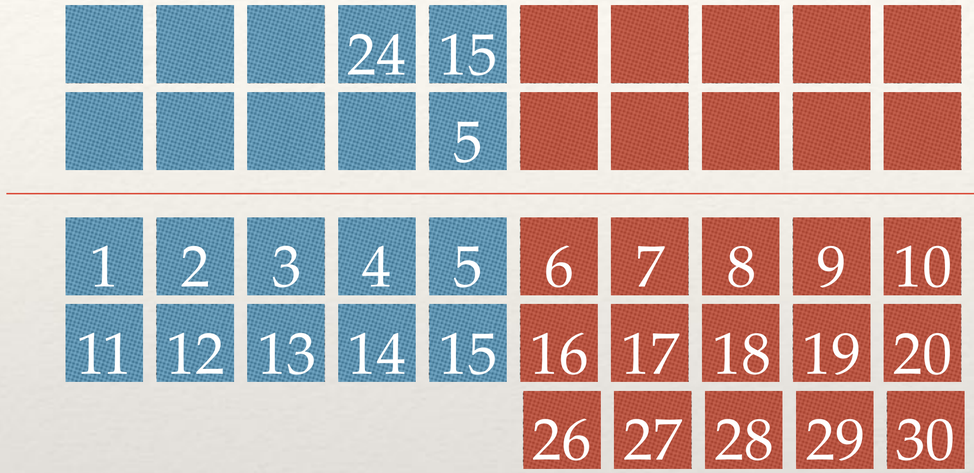


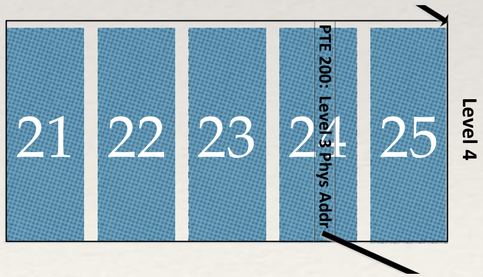


Pagetable in DRAM

EVICT+TIME in Cache

- Still cached

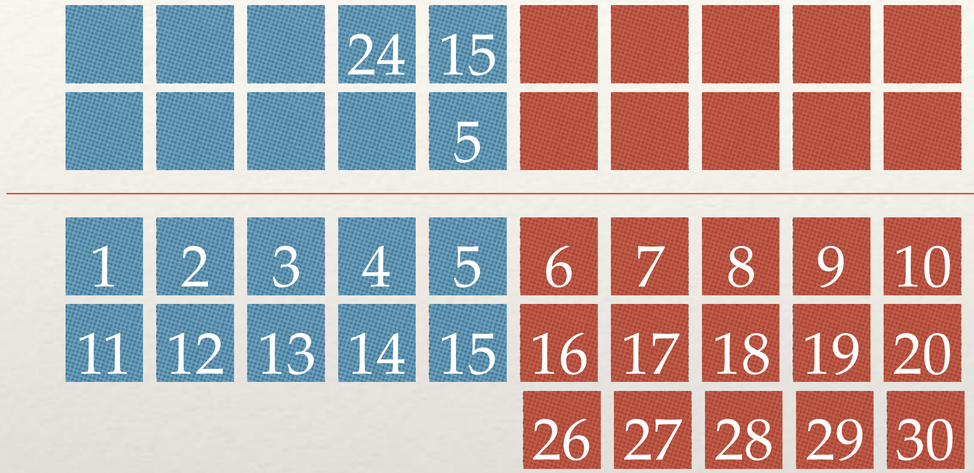


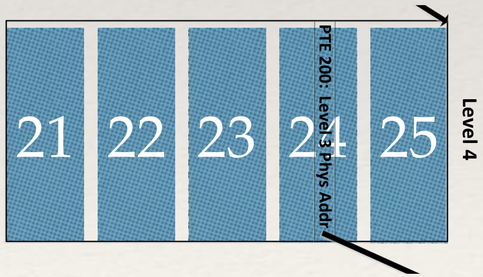


Pagetable in DRAM

EVICT+TIME in Cache

- Let's evict

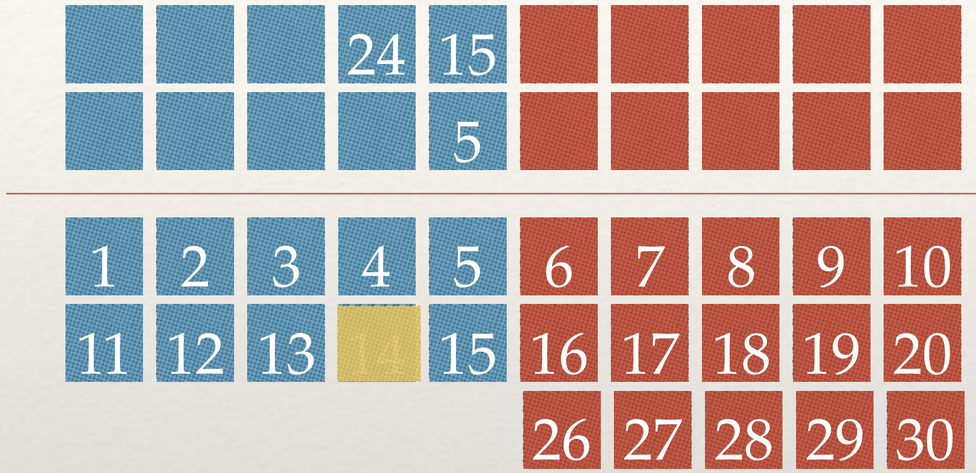


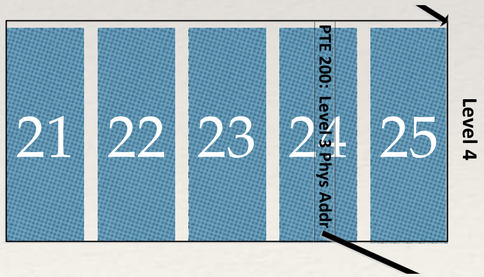


Pagetable in DRAM

EVICT+TIME in Cache

- Let's evict

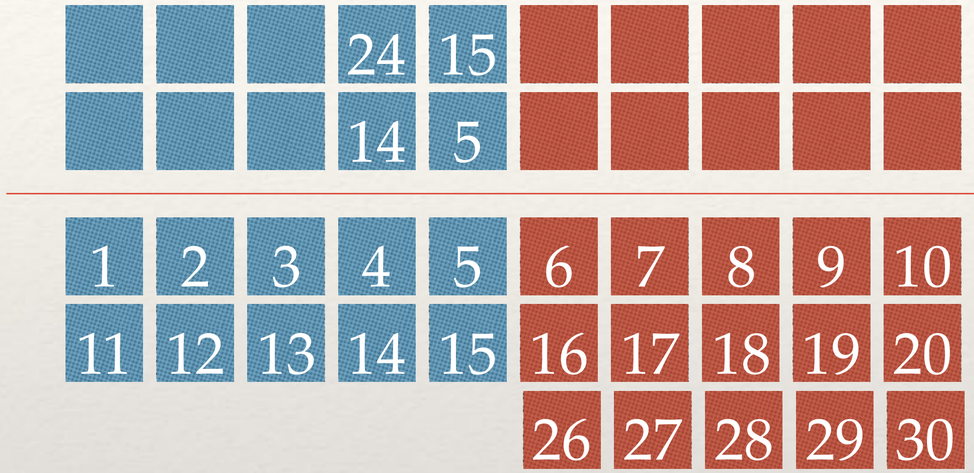


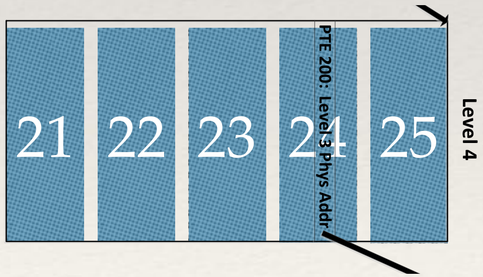


Pagetable in DRAM

EVICT+TIME in Cache

- Let's evict

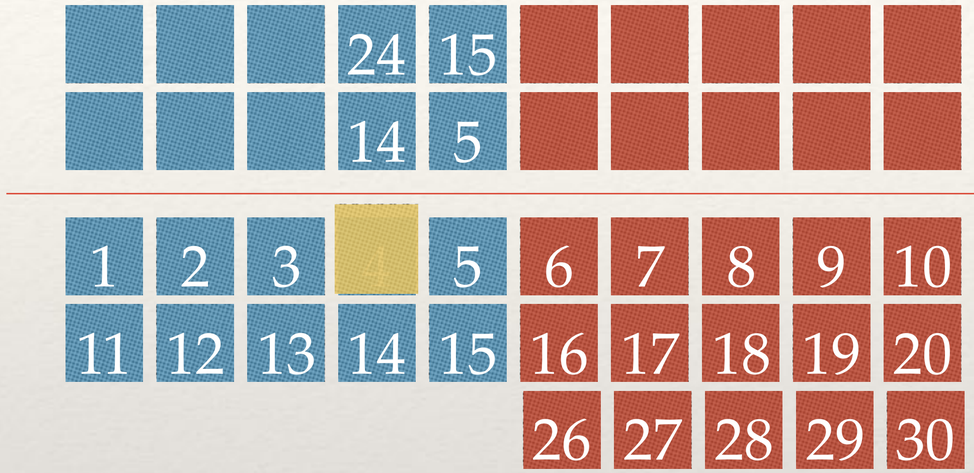


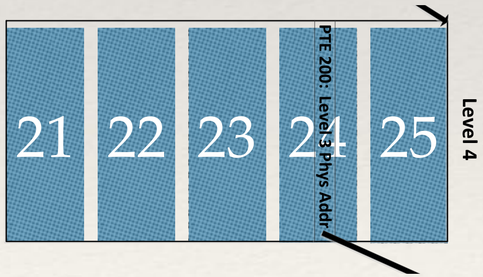


Pagetable in DRAM

EVICT+TIME in Cache

- Let's evict

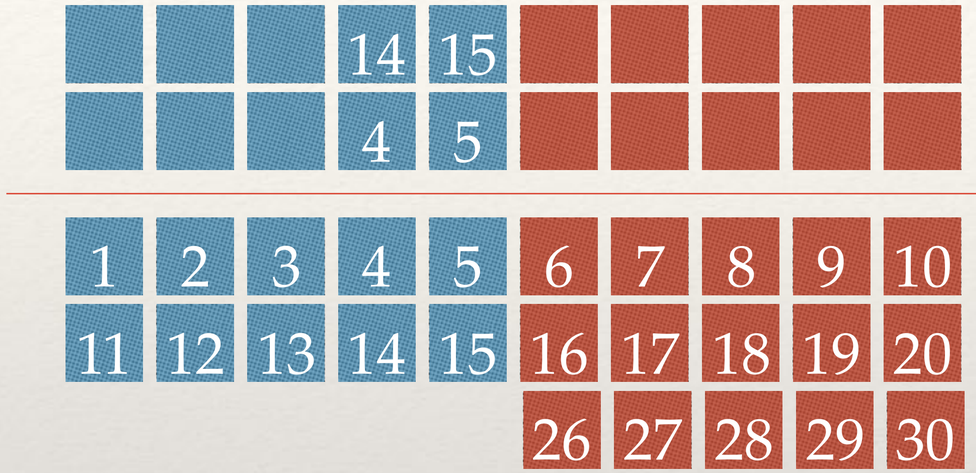


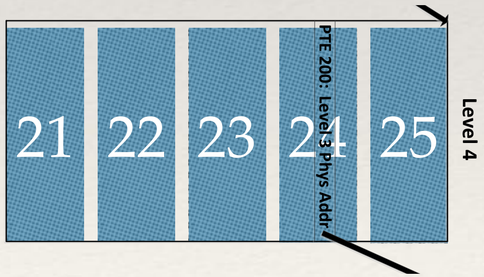


Pagetable in DRAM

EVICT+TIME in Cache

- Let's evict

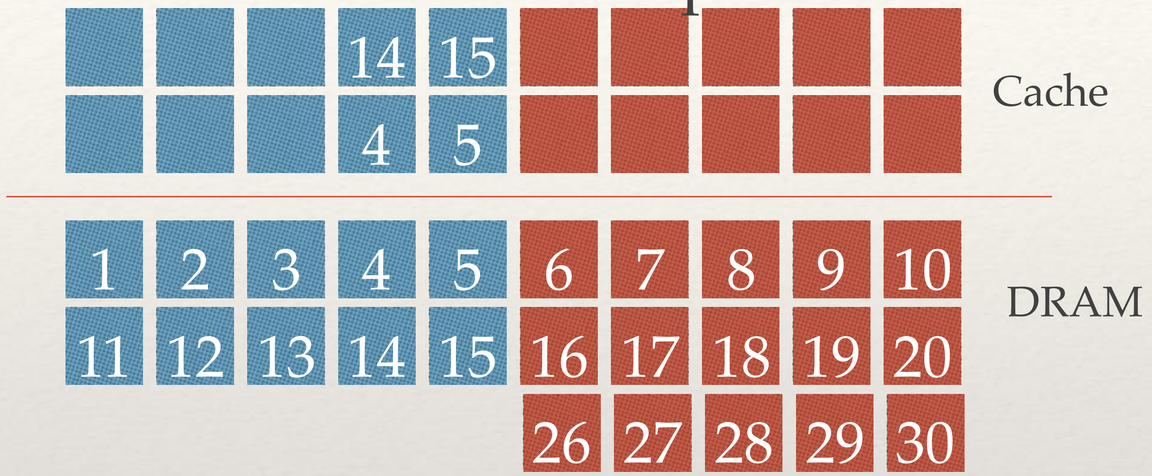


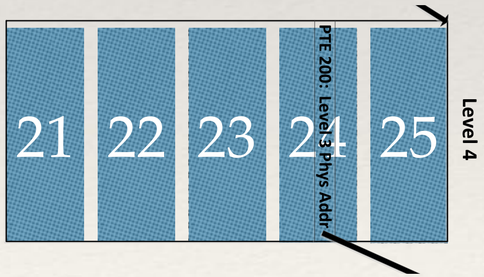


Pagetable in DRAM

EVICT+TIME in Cache

- Eviction done - let's do lookup

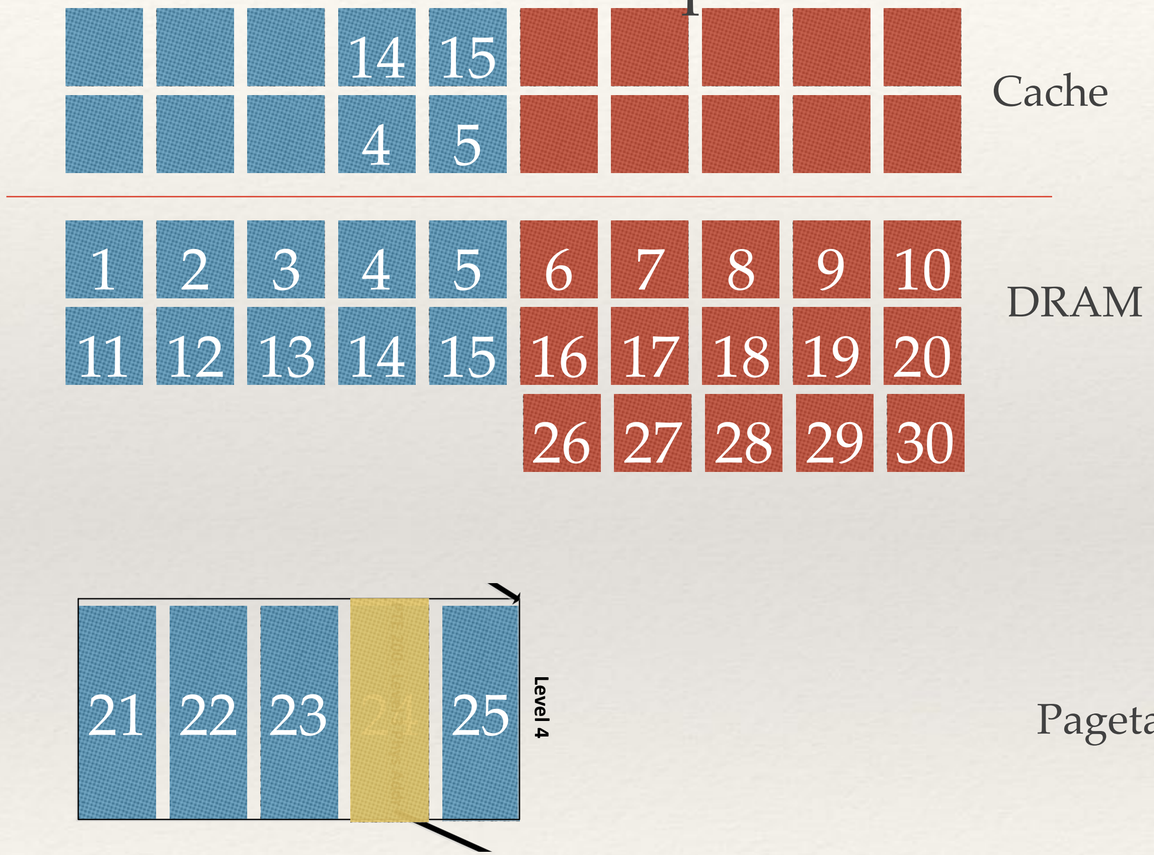




Pagetable in DRAM

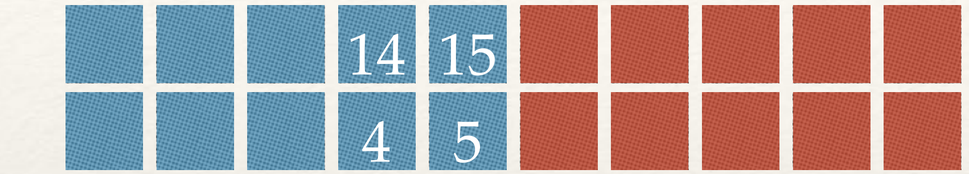
EVICT+TIME in Cache

- Eviction done - let's do lookup

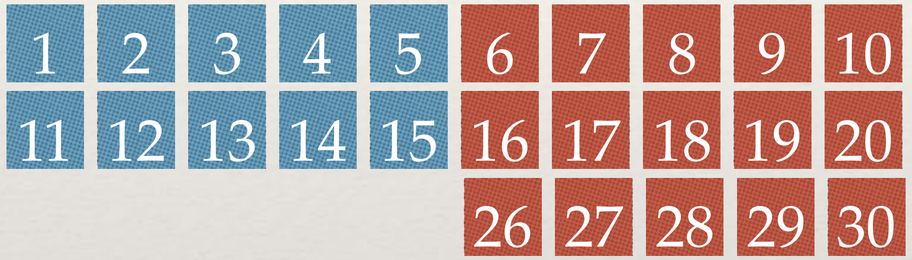


EVICT+TIME in Cache

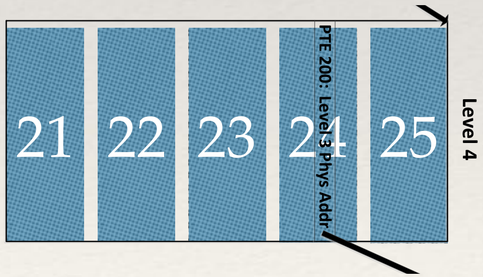
- Uncached now



Cache



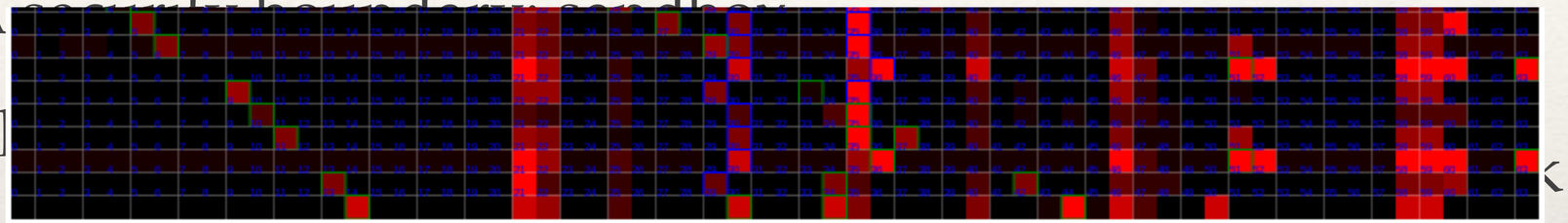
DRAM



Pagetable in DRAM

JavaScript Complications

- A
- Th
- Ir
- L
- Contiguous virtual address space



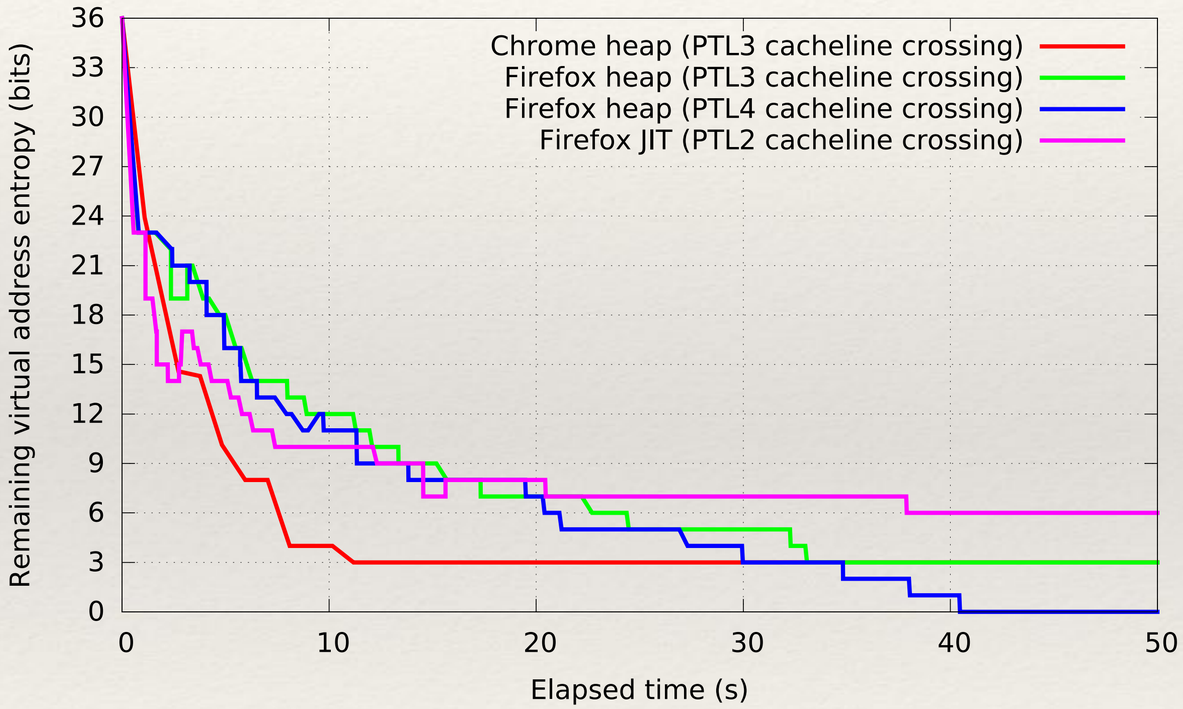
```
24.457 got level 4 - start slot 148, address 0x94000
24.993 got level 3 - start slot 295, address 0x24e94000
24.993 estimated remaining entropy 6 slot solutions: -1,-1,295,148
68.737 got level 4 - start slot 0, address 0x0
69.502 got level 3 - start slot 359, address 0x2ce00000
70.259 got level 2 - start slot 411, address 0x66ece00000
88.041 got level 1 - start slot 238, address 0x7766ece00000
88.041 estimated remaining entropy 0 slot solutions: 238,411,359,0
data: 0x7766ece00000, code slots: -1,-1,295,148, code: 0x7966e4e94000
```

k of JIT

he iTLB

Prototypes in Firefox and Chrome

Results: speed



Results: tested microarchitectures

CPU Model	Microarchitecture	Year
Intel Xeon E3-1240 v5	Skylake	2015
Intel Core i7-6700K	Skylake	2015
Intel Celeron N2840	Silvermont	2014
Intel Xeon E5-2658 v2	Ivy Bridge EP	2013
Intel Atom C2750	Silvermont	2013
Intel Core i7-4500U	Haswell	2013
Intel Core i7-3632QM	Ivy Bridge	2012
Intel Core i7-2620QM	Sandy Bridge	2011
Intel Core i5 M480	Westmere	2010
Intel Core i7 920	Nehalem	2008
AMD FX-8350 8-Core	Piledriver	2012
AMD FX-8320 8-Core	Piledriver	2012
AMD FX-8120 8-Core	Bulldozer	2011
AMD Athlon II 640 X4	K10	2010
AMD E-350	Bobcat	2010
AMD Phenom 9550 4-Core	K10	2008
Allwinner A64	ARM Cortex A53	2016
Samsung Exynos 5800	ARM Cortex A15	2014
Samsung Exynos 5800	ARM Cortex A7	2014
Nvidia Tegra K1 CD580M-A1	ARM Cortex A15	2014
Nvidia Tegra K1 CD570M-A1	ARM Cortex A15; LPAE	2014

FLIP FENG SHUI

Flip Feng Shui

- Malicious VM in the cloud compromising neighbors
- Rowhammer + Memory Deduplication
- You can flip a bit in another VM

What would YOU flip?

Rowhammer

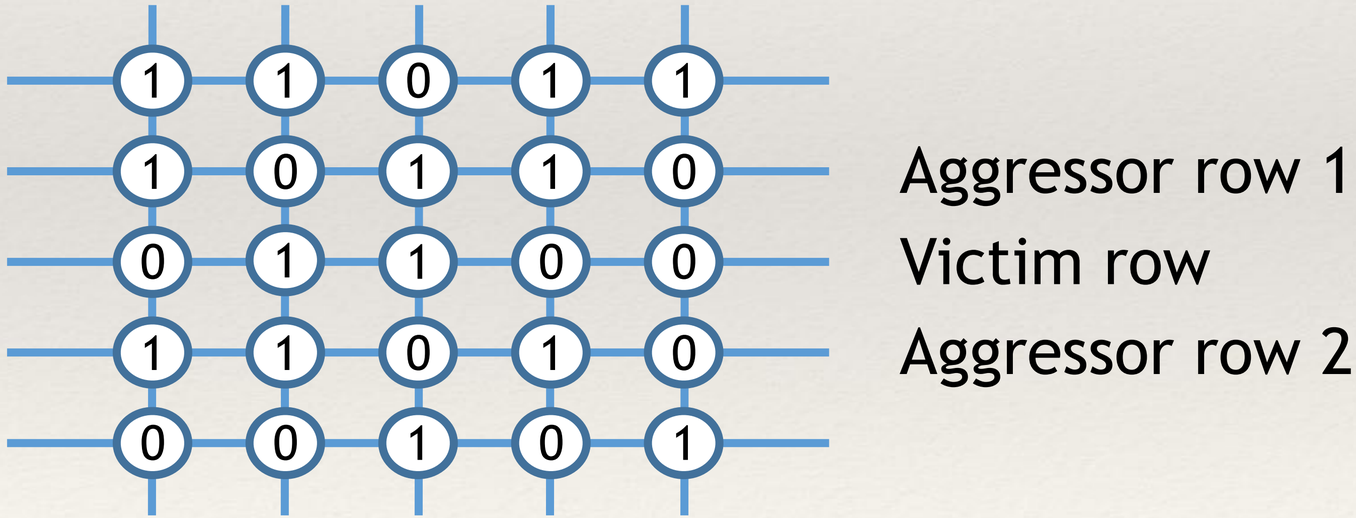
A DRAM hardware glitch

- Your DRAM is not as reliable as you think…
- If you read quickly charge leakages will result in bits to flip
- Does not happen due to multiple levels of caches
- 80+% of DDR3 DIMMs affected, reports also on DDR4

Rowhammer

A DRAM hardware glitch

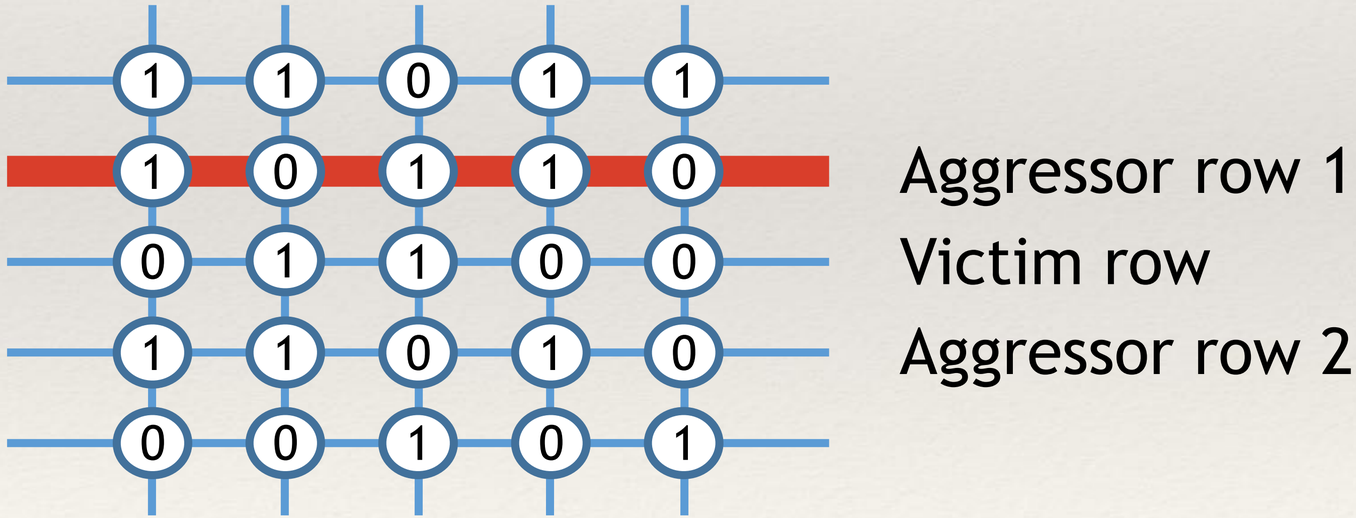
- Your DRAM is not as reliable as you think…
- If you read quickly charge leakages will result in bits to flip
- Does not happen due to multiple levels of caches
- 80+% of DDR3 DIMMs affected, reports also on DDR4



Rowhammer

A DRAM hardware glitch

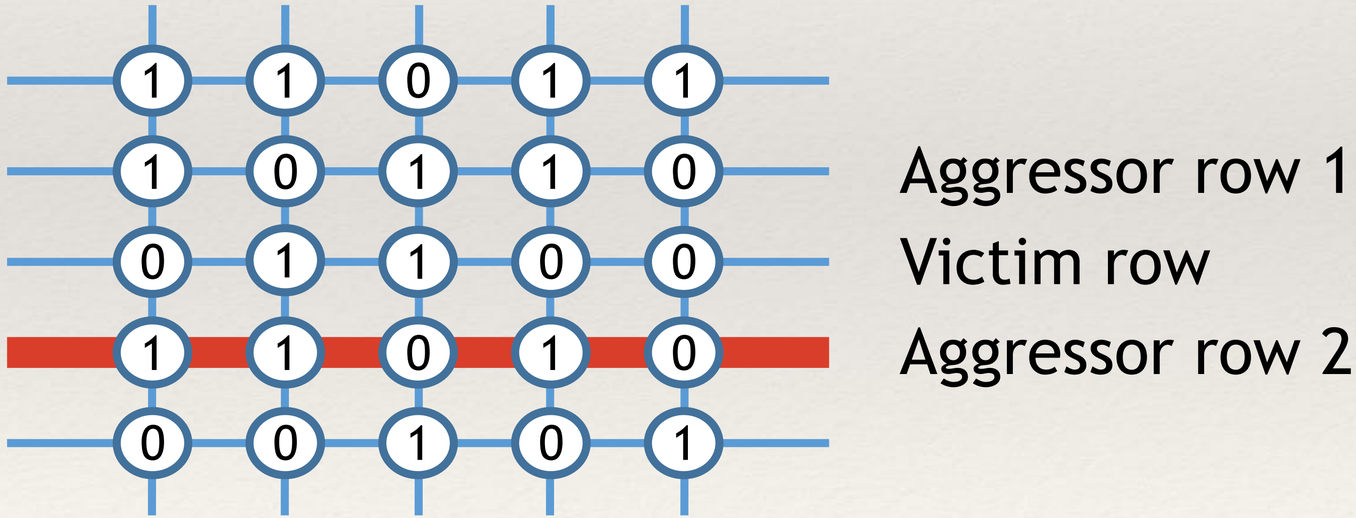
- Your DRAM is not as reliable as you think…
- If you read quickly charge leakages will result in bits to flip
- Does not happen due to multiple levels of caches
- 80+% of DDR3 DIMMs affected, reports also on DDR4



Rowhammer

A DRAM hardware glitch

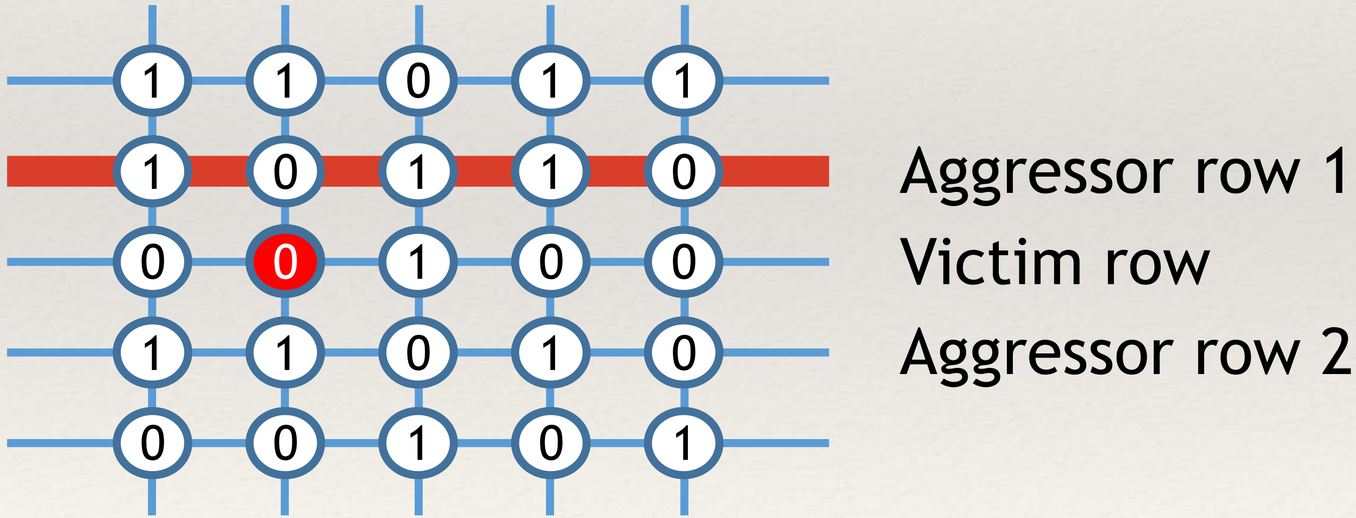
- Your DRAM is not as reliable as you think…
- If you read quickly charge leakages will result in bits to flip
- Does not happen due to multiple levels of caches
- 80+% of DDR3 DIMMs affected, reports also on DDR4



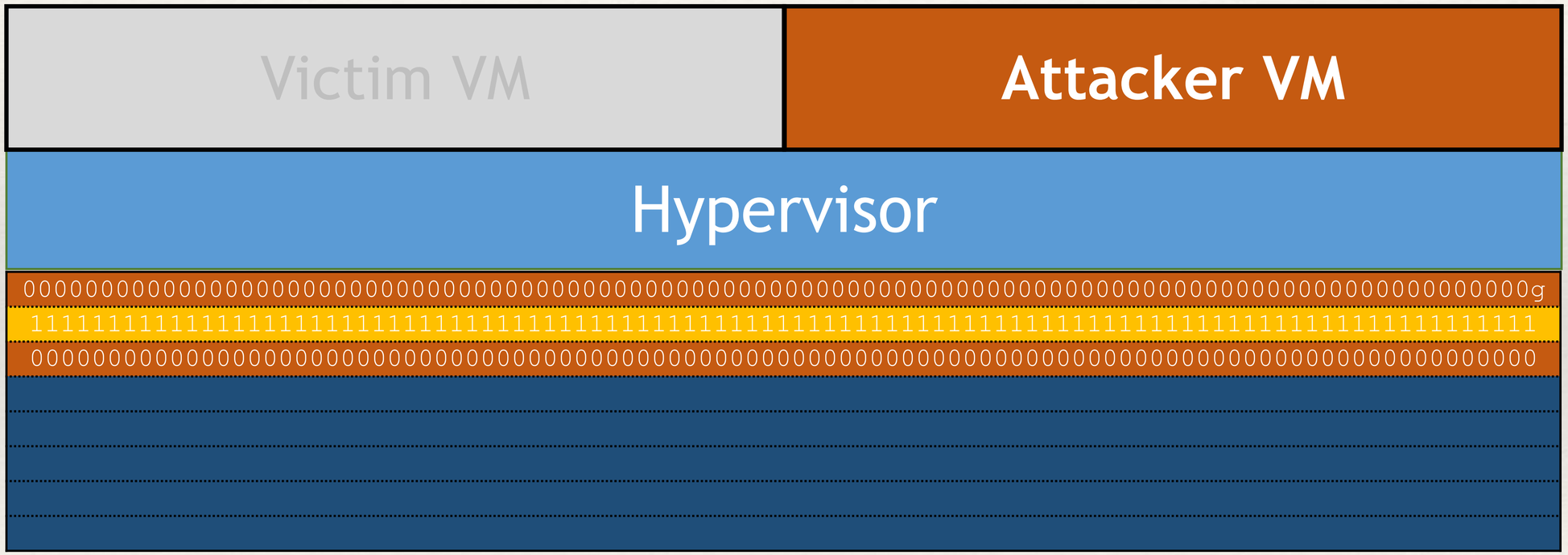
Rowhammer

A DRAM hardware glitch

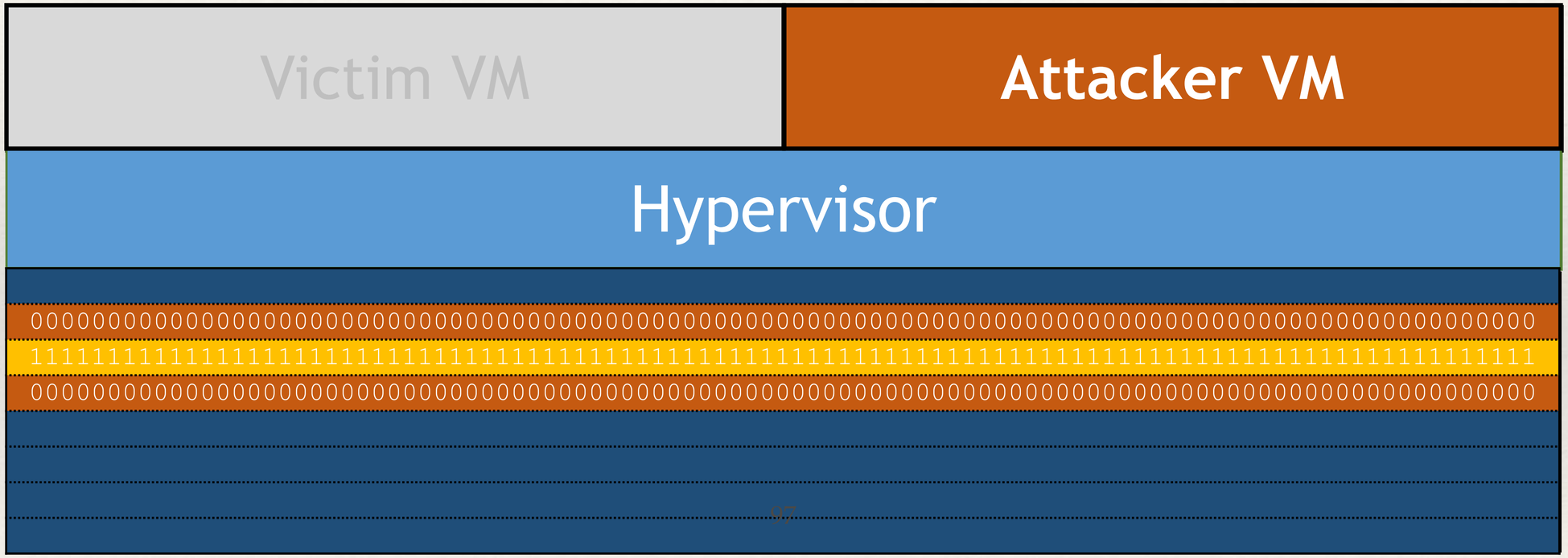
- Your DRAM is not as reliable as you think…
- If you read quickly charge leakages will result in bits to flip
- Does not happen due to multiple levels of caches
- 80+% of DDR3 DIMMs affected, reports also on DDR4



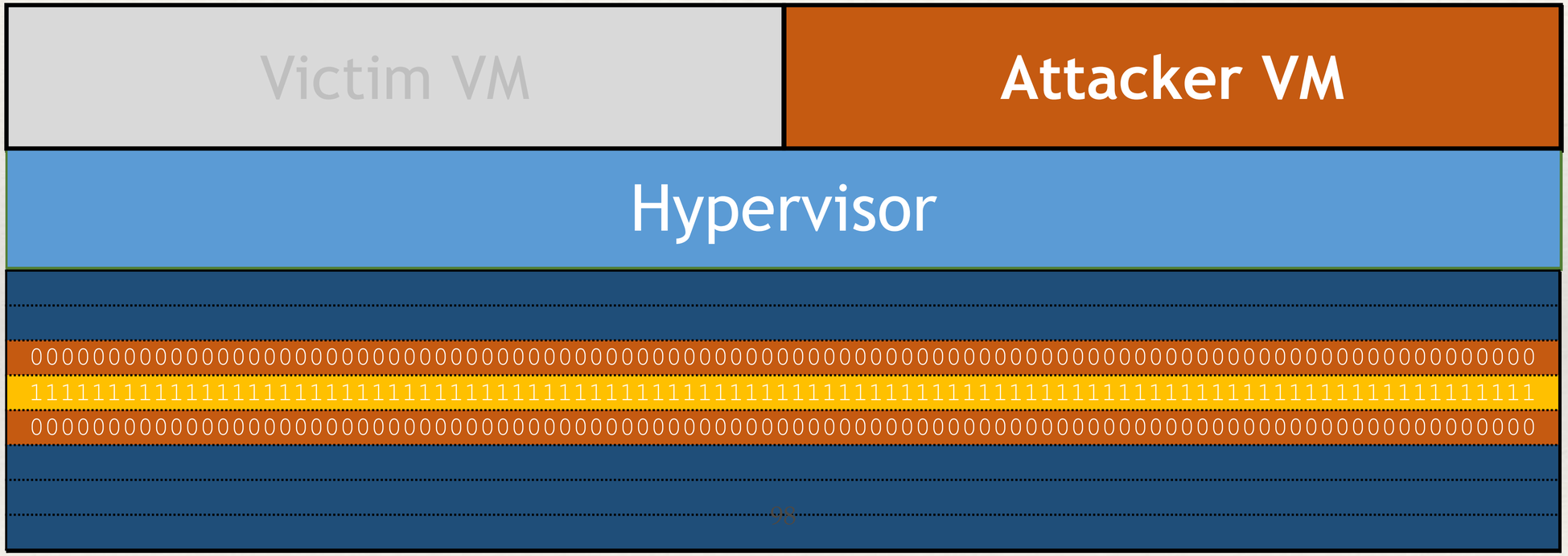
Rowhammering in the Cloud



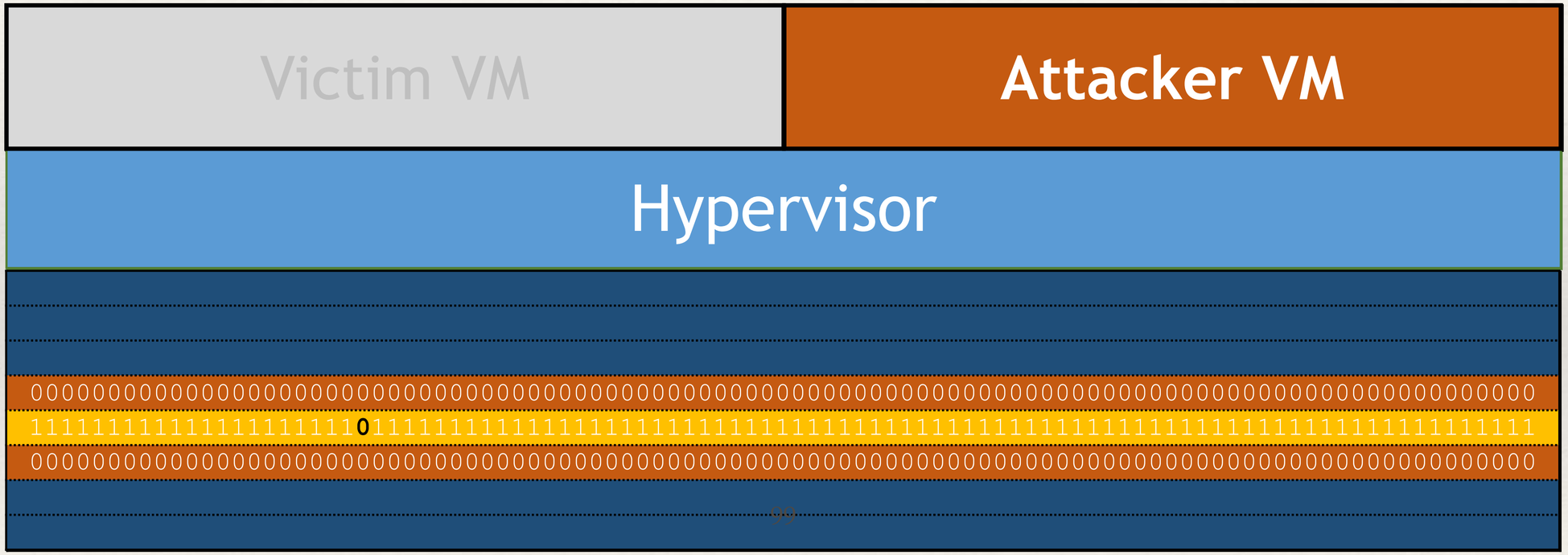
Rowhammering in the Cloud



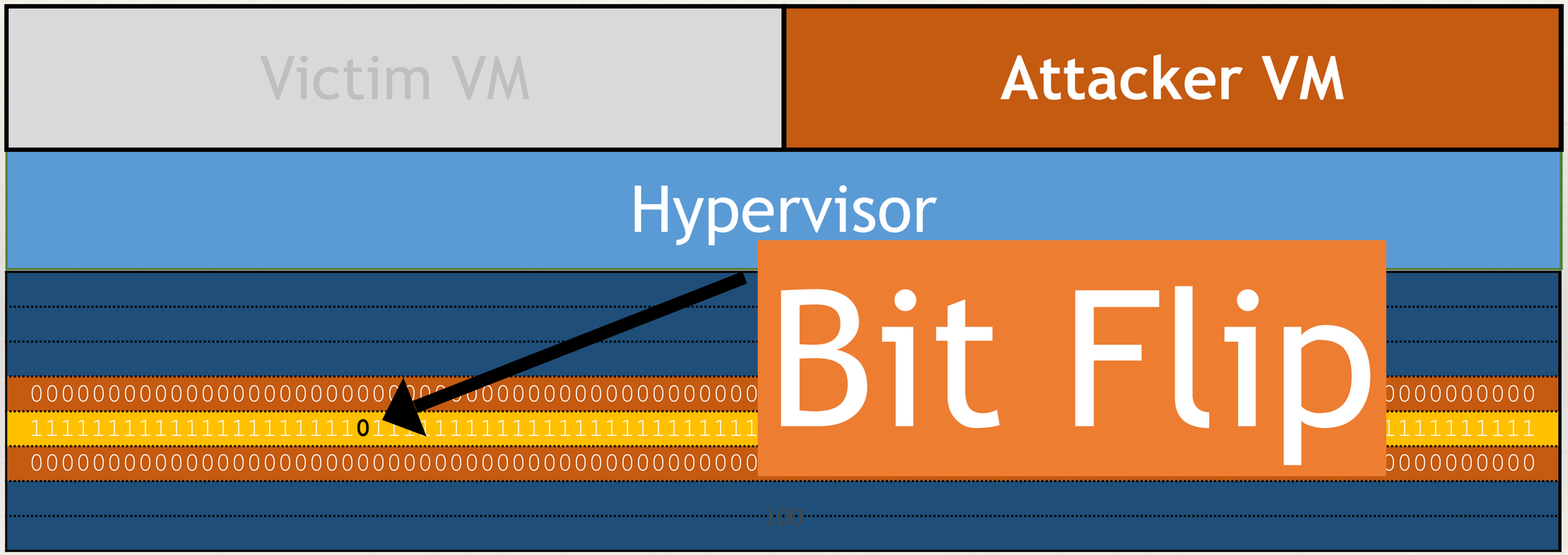
Rowhammering in the Cloud



Rowhammering in the Cloud

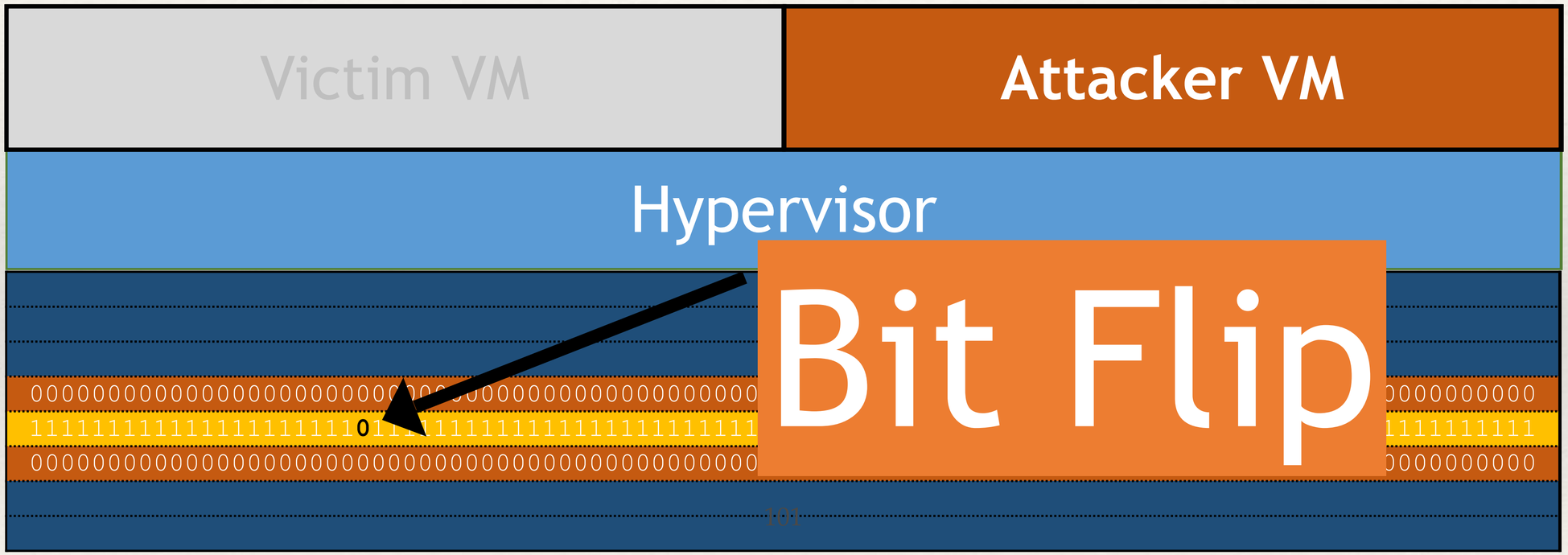


Rowhammering in the Cloud



Rowhammering in the Cloud

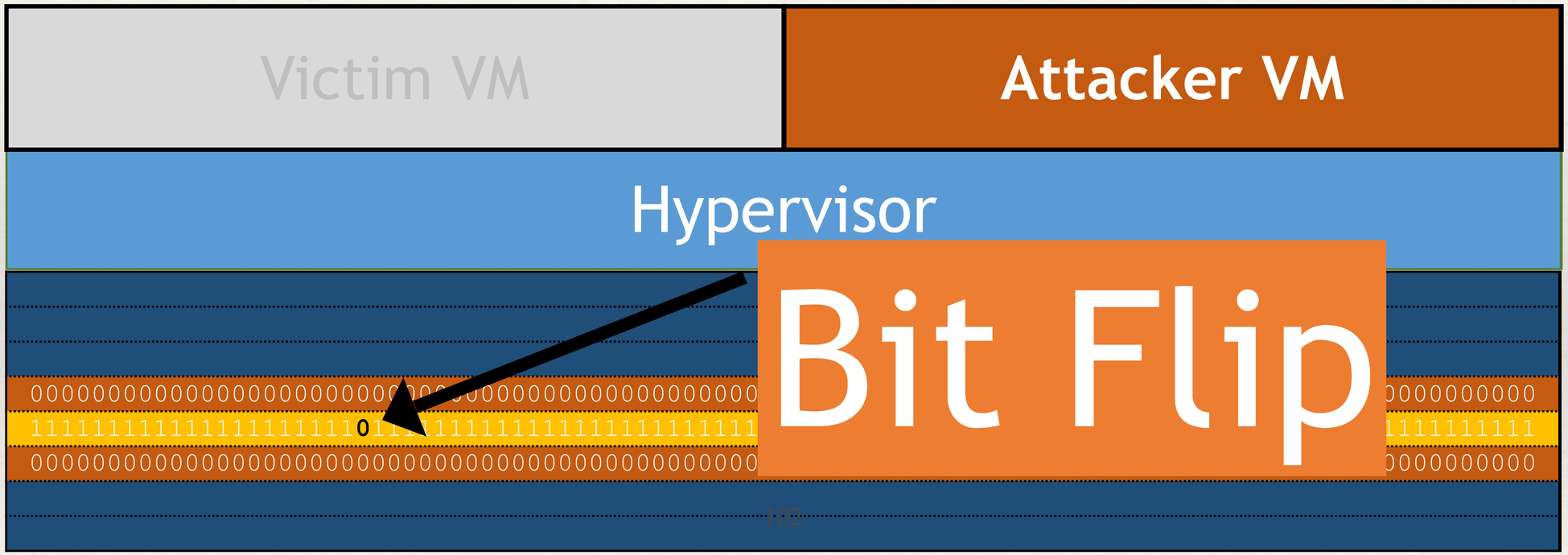
How do we force the Victim VM to store sensitive data here?



Rowhammering in the Cloud

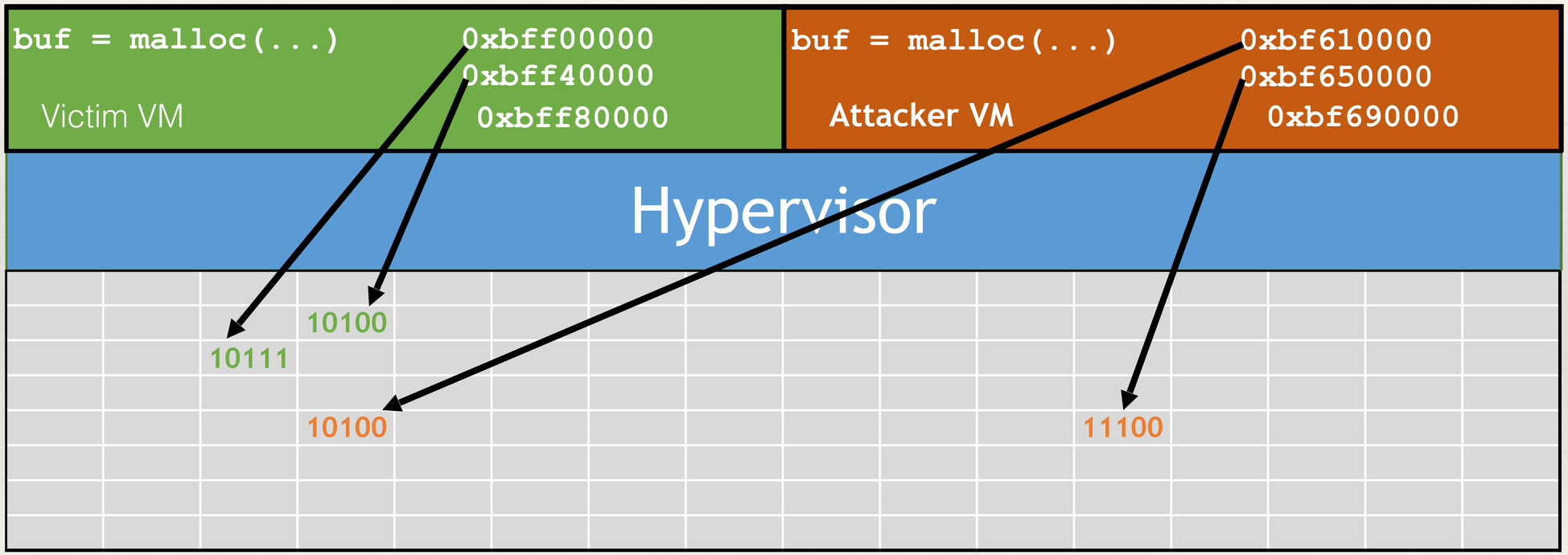
How do we force the Victim VM to store sensitive data here?

Memory Deduplication



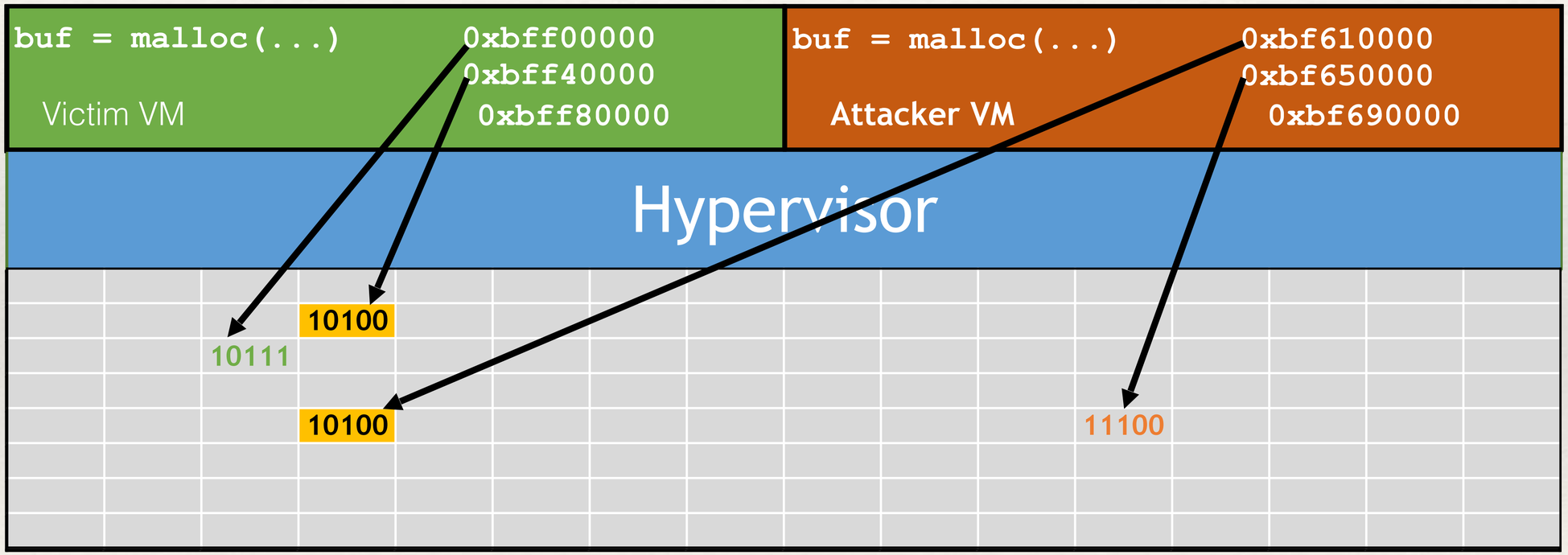
Memory Deduplication

Operating System / Hypervisor searches for *duplicate* pages



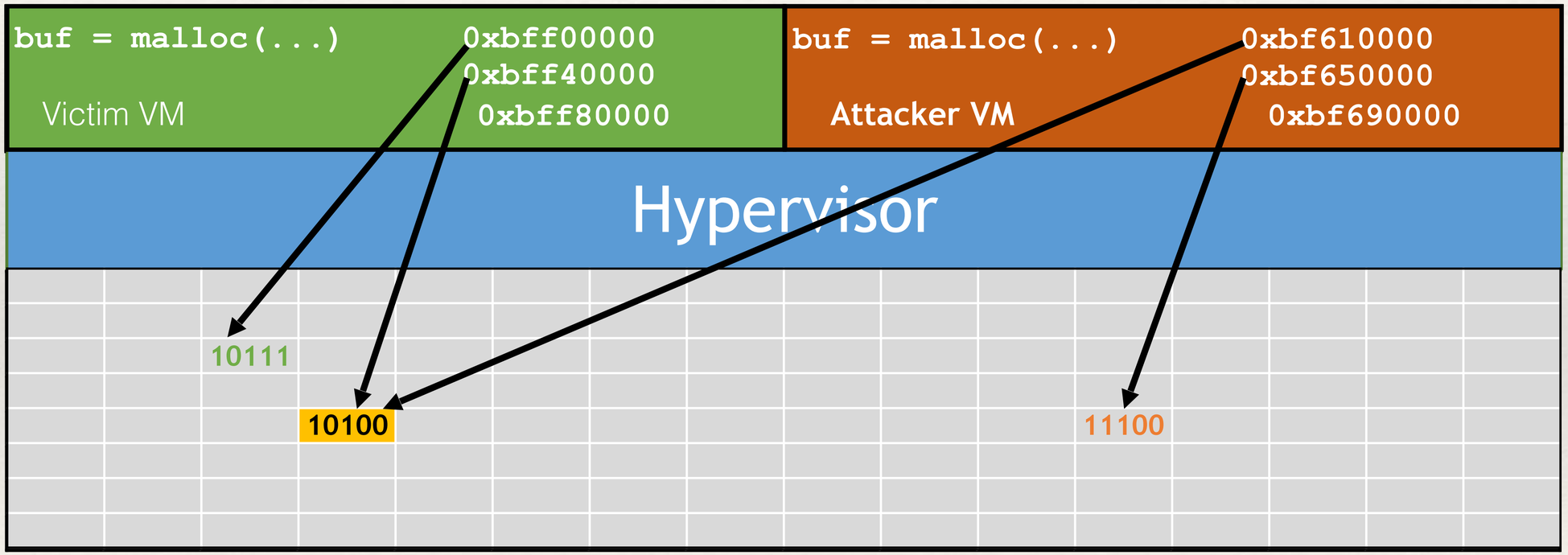
Memory Deduplication

Operating System / Hypervisor searches for *duplicate* pages



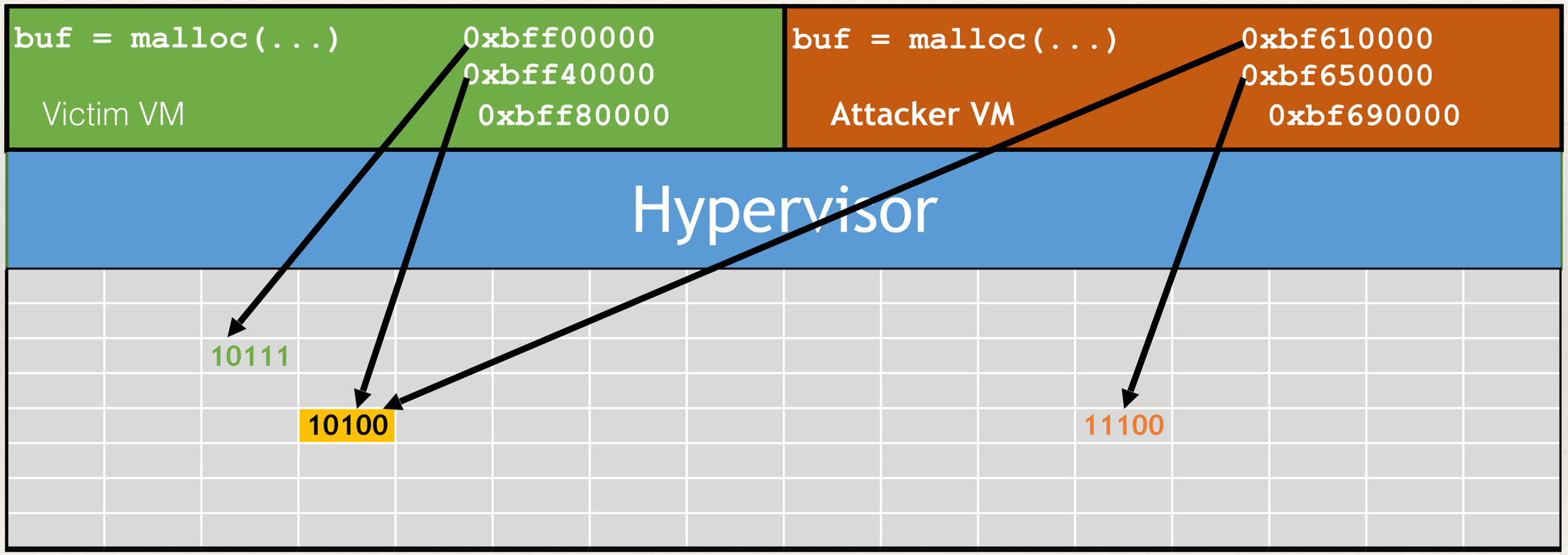
Memory Deduplication

Operating System / Hypervisor searches for *duplicate* pages



Memory Deduplication

Operating System / Hypervisor searches for *duplicate* pages



Move victim's page into a location with a bit flip!

Flip Feng Shui

What would you flip?

Juicy Targets

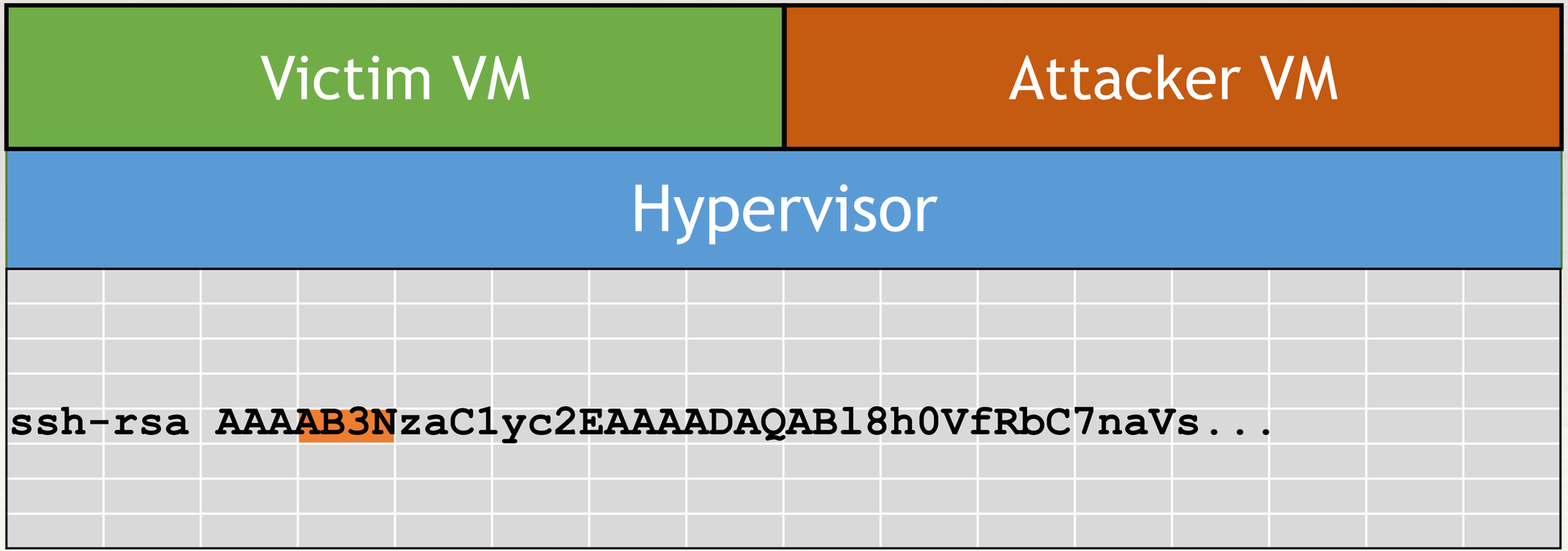
Cryptographic public keys

Domain names

Reproduce the bit flip

Cryptographic keys

Public RSA key from `.ssh/authorized_keys`

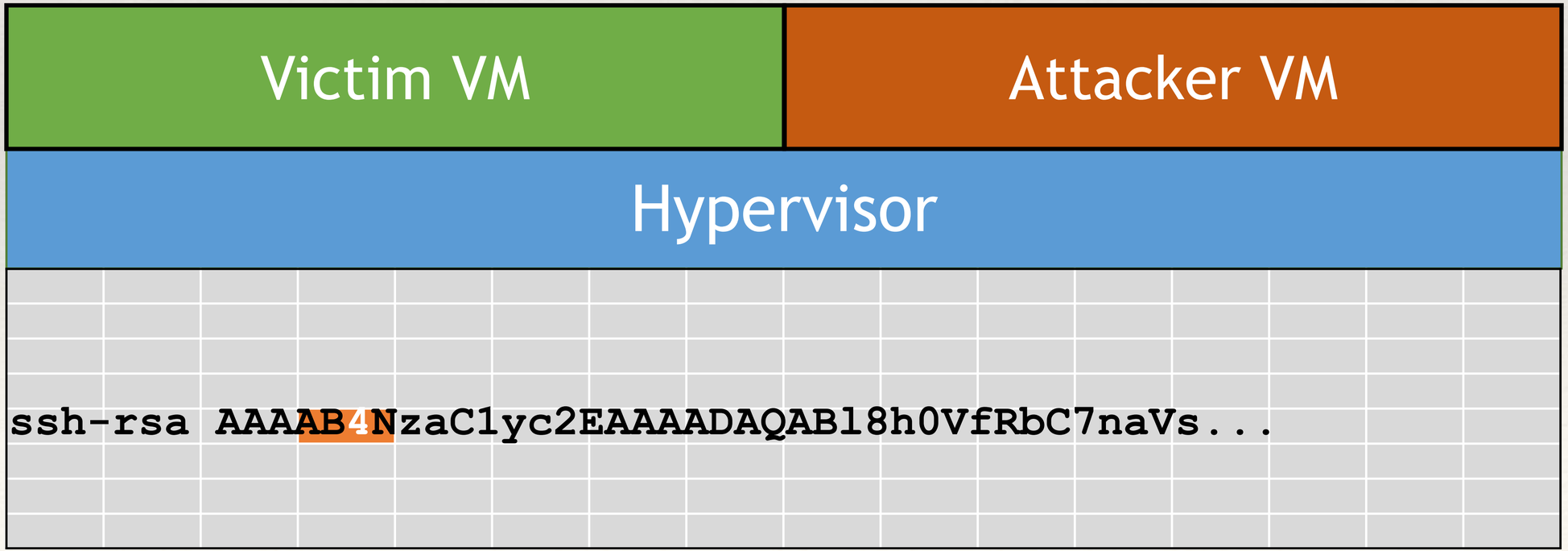


Reproduce the bit flip

Cryptographic keys

Public RSA key from `.ssh/authorized_keys`

- Flipping a bit changes the public/private key pair

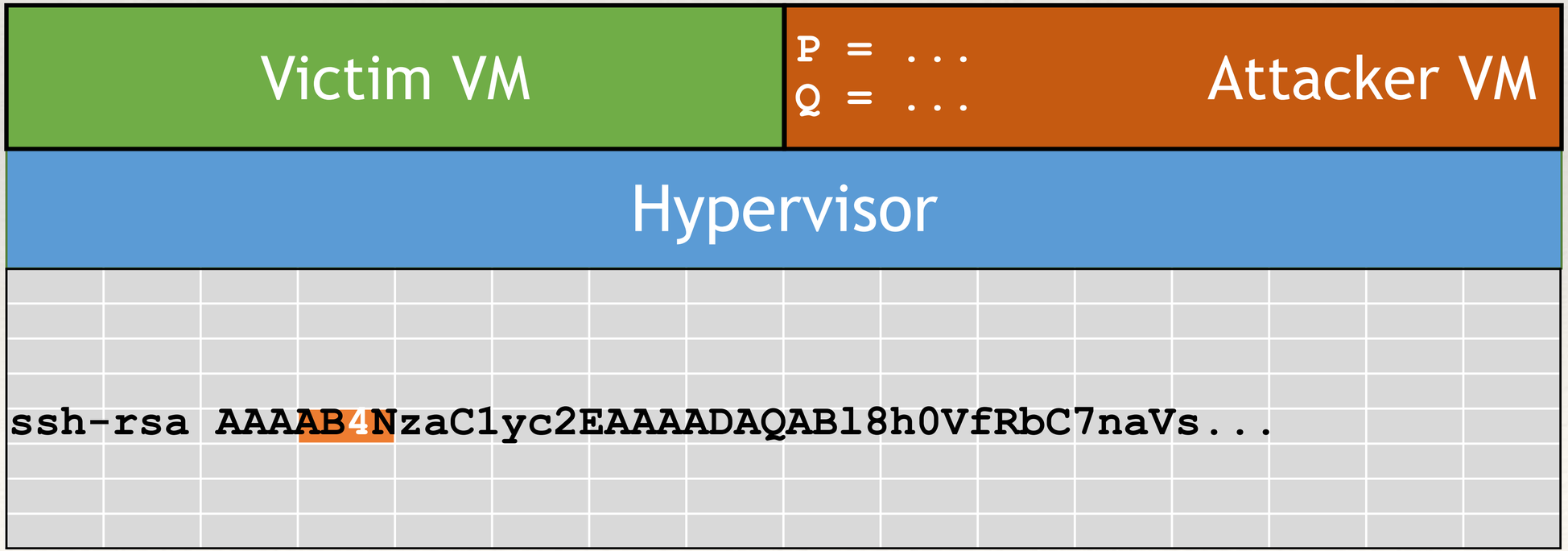


Reproduce the bit flip

Cryptographic keys

Public RSA key from `.ssh/authorized_keys`

- Flipping a bit changes the public/private key pair
- New public key is *easy* to factorize

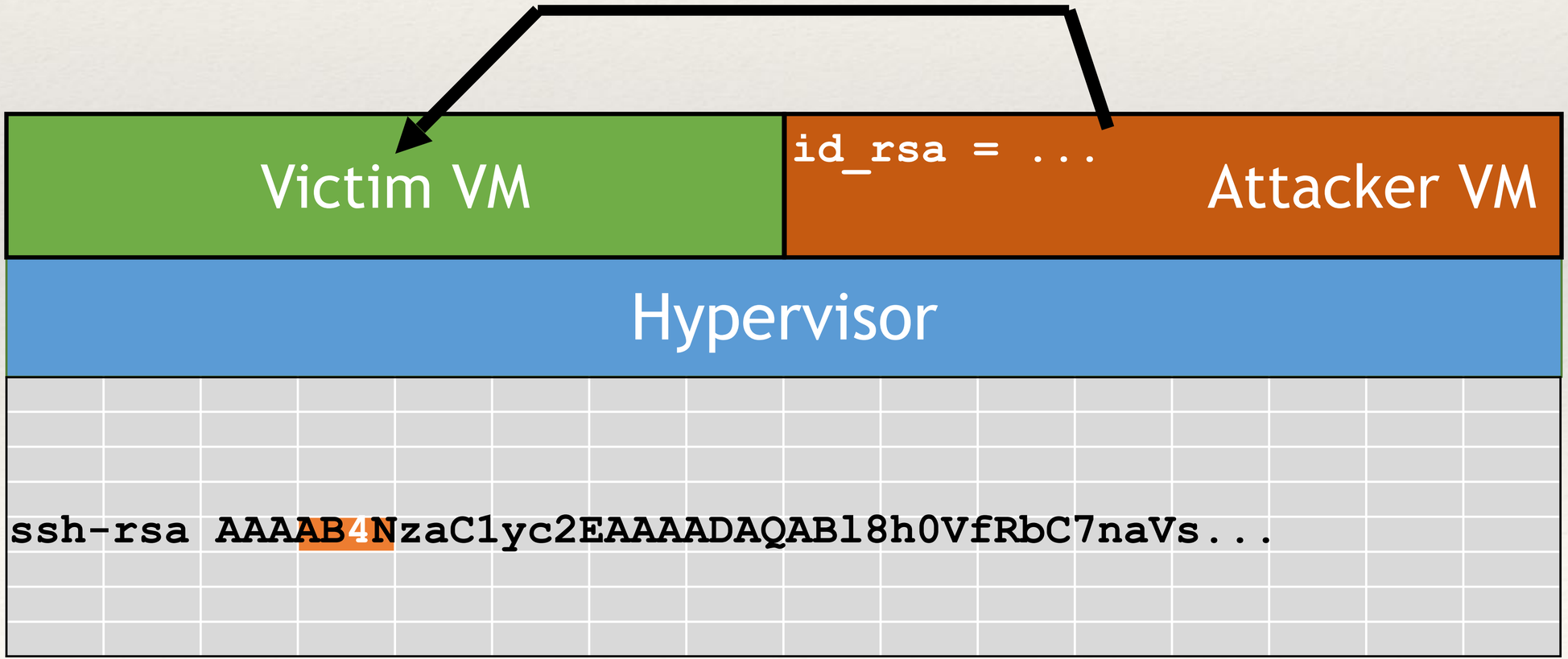


Reproduce the bit flip

Cryptographic keys

Public RSA key from `.ssh/authorized_keys`

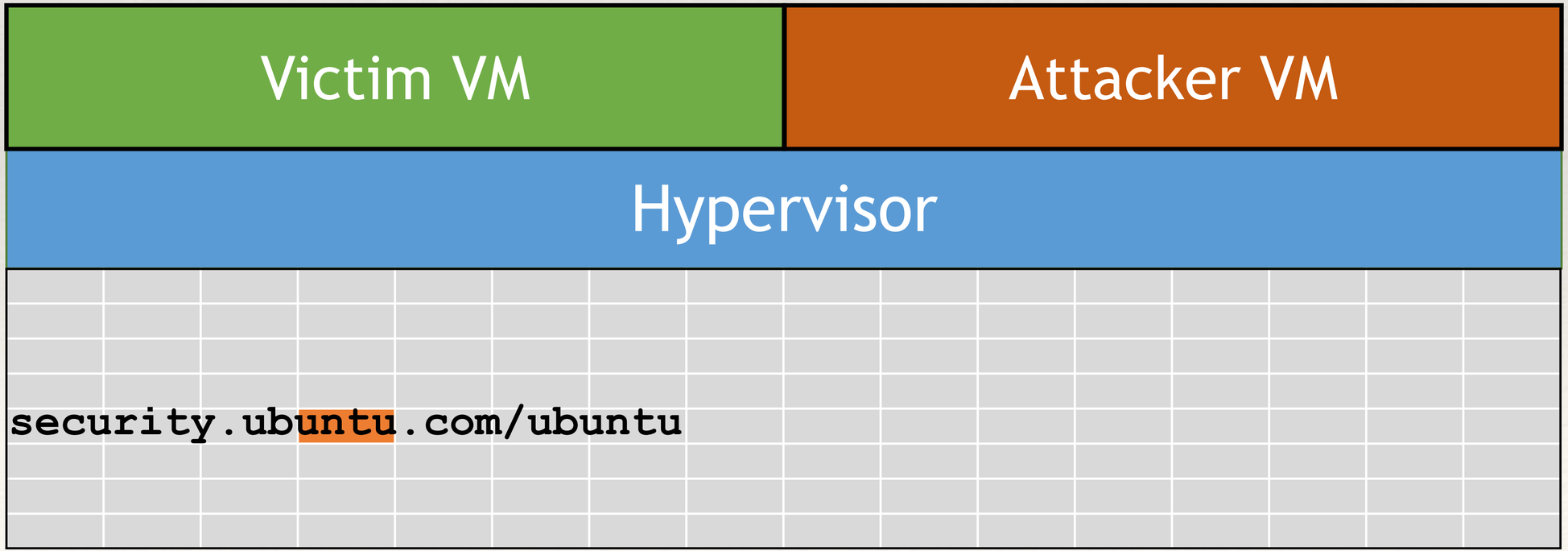
- Flipping a bit changes the public/private key pair
- New public key is *easy* to factorize
- Attacker computes the new private key and gets SSH access



Reproduce the bit flip

Cryptographic keys + domain names

Domain name used for **`apt-get upgrade`**

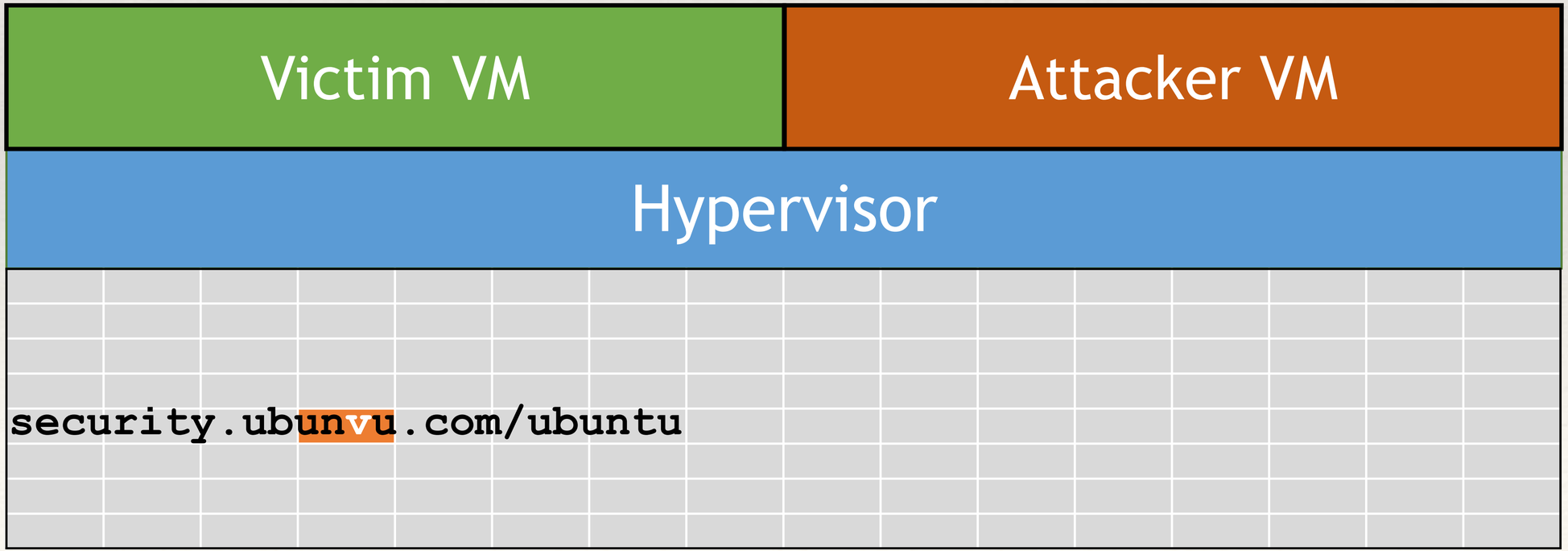


Reproduce the bit flip

Cryptographic keys + domain names

Domain name used for `apt-get upgrade`

- Flipping a bit changes the domain name used to pull updates
- Attacker hosts malicious `ls` command at **`ubunvu.com`**

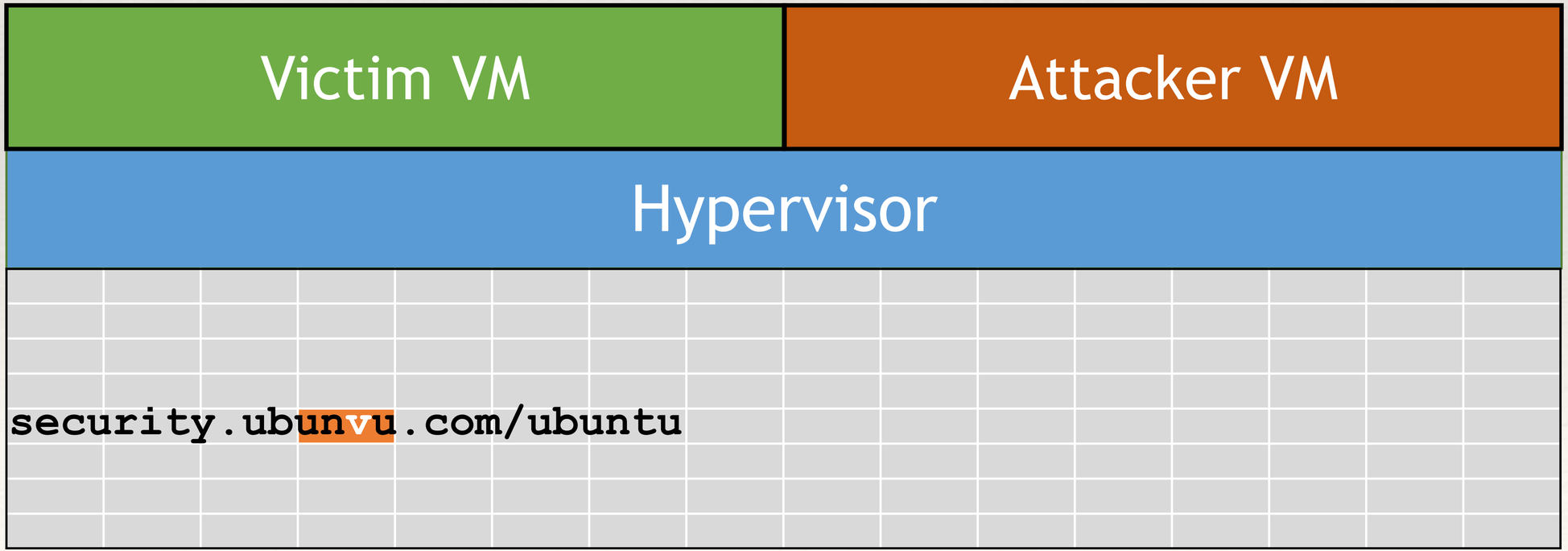


Reproduce the bit flip

Cryptographic keys + domain names

Domain name used for `apt-get upgrade`

- Flipping a bit changes the domain name used to pull updates
- Attacker hosts malicious `ls` command at **`ubunvu.com`**
- Packages are signed: also flip a bit in the GPG keychain

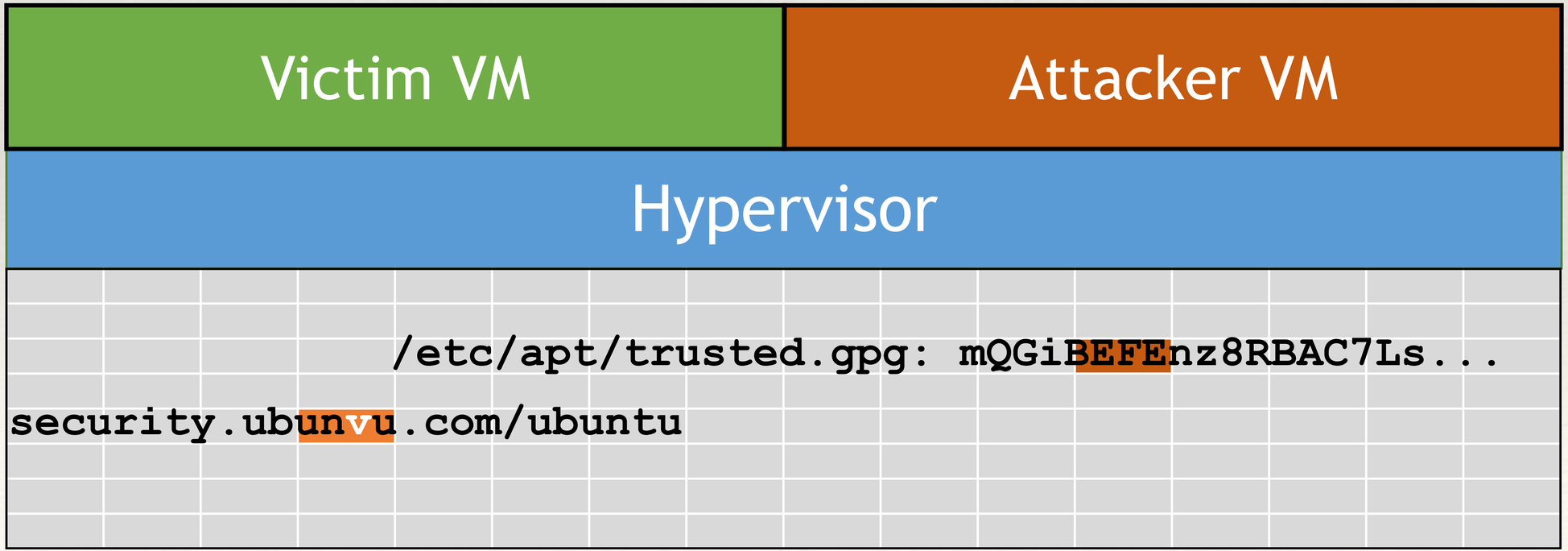


Reproduce the bit flip

Cryptographic keys + domain names

Domain name used for `apt-get upgrade`

- Flipping a bit changes the domain name used to pull updates
- Attacker hosts malicious `ls` command at **`ubunvu.com`**
- Packages are signed: also flip a bit in the GPG keychain

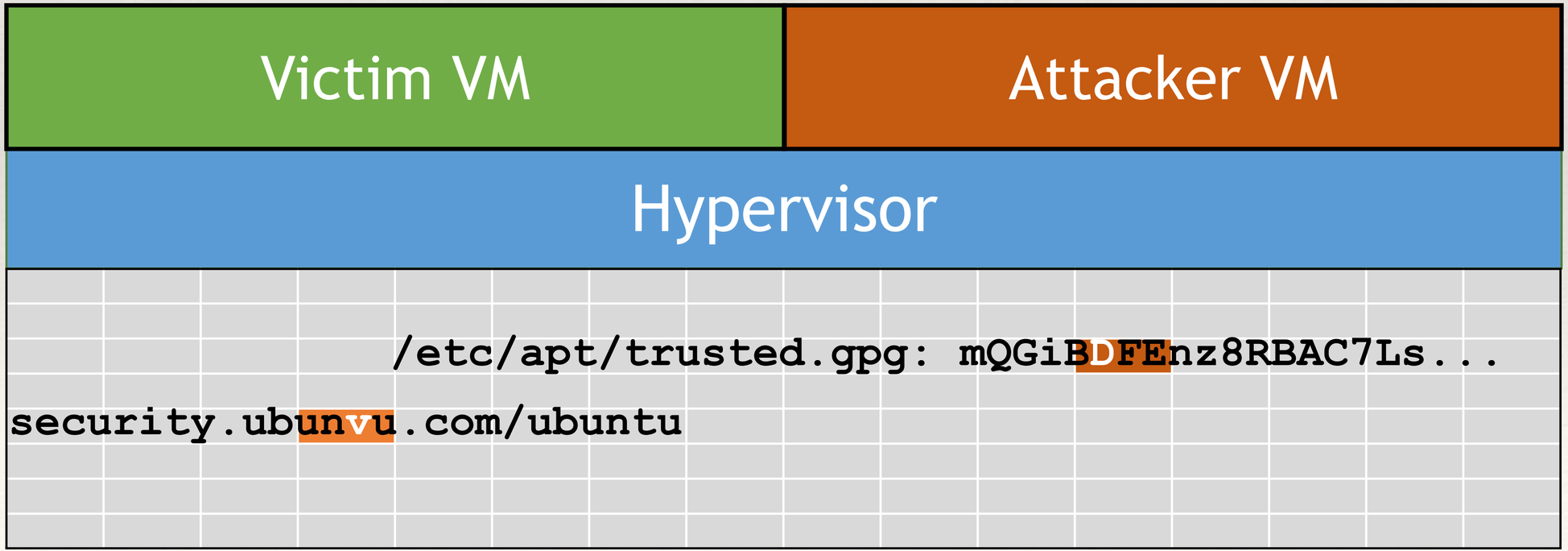


Reproduce the bit flip

Cryptographic keys + domain names

Domain name used for **`apt-get upgrade`**

- Flipping a bit changes the domain name used to pull updates
- Attacker hosts malicious **`ls`** command at **`ubunvu.com`**
- Packages are signed: also flip a bit in the GPG keychain

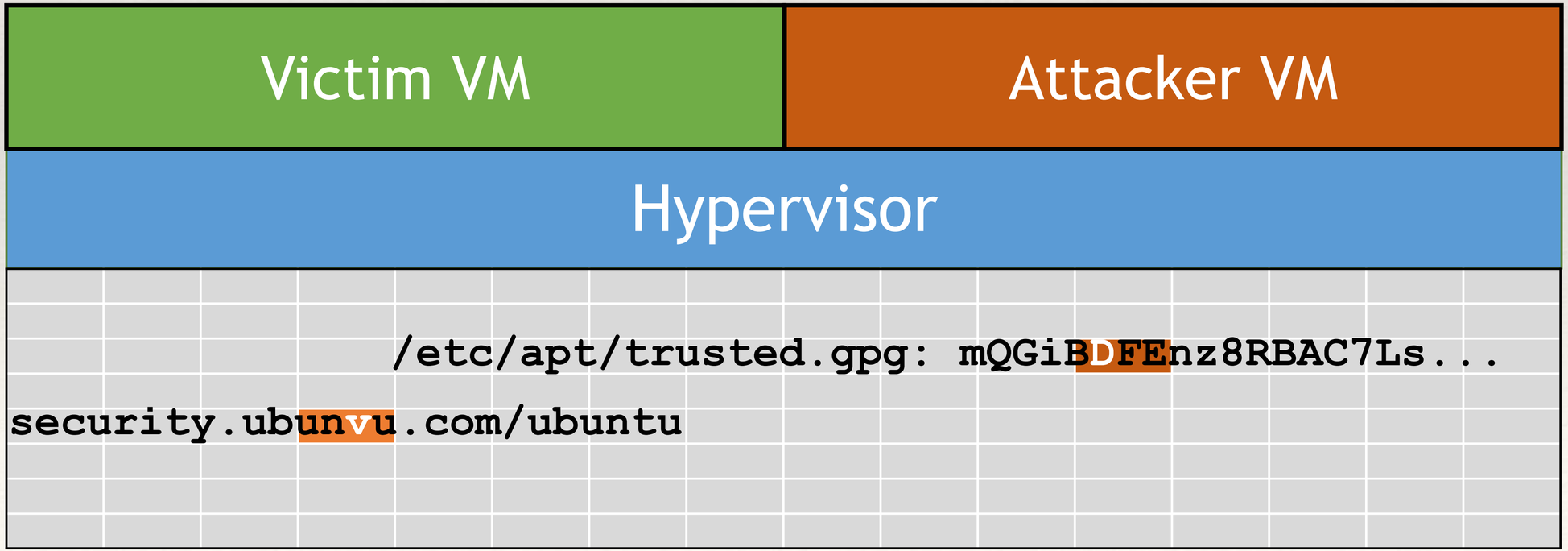


Reproduce the bit flip

Cryptographic keys + domain names

Domain name used for **`apt-get upgrade`**

- Flipping a bit changes the domain name used to pull updates
- Attacker hosts malicious **`ls`** command at **`ubunvu.com`**
- Packages are signed: also flip a bit in the GPG keychain
- Attacker computes the new GPG key and signs his backdoored **`ls`**

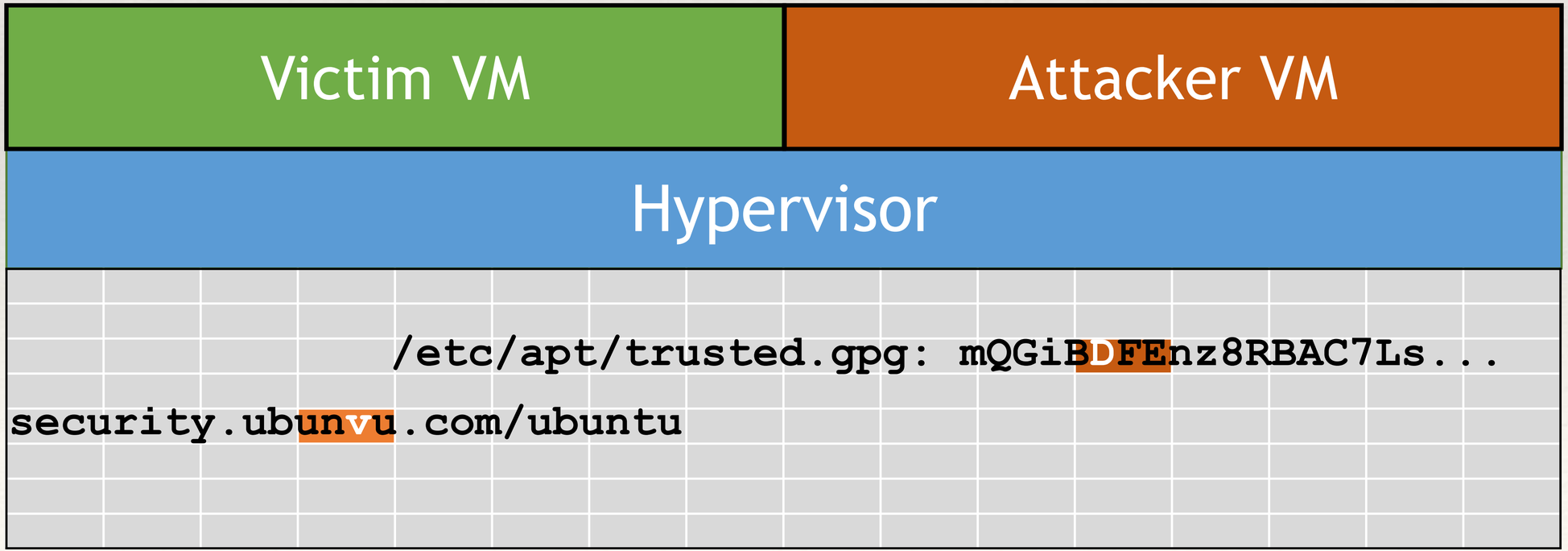


Reproduce the bit flip

Cryptographic keys + domain names

Domain name used for `apt-get upgrade`

- Flipping a bit changes the domain name used to pull updates
- Attacker hosts malicious `ls` command at **`ubunvu.com`**
- Packages are signed: also flip a bit in the GPG keychain
- Attacker computes the new GPG key and signs his backdoored `ls`
- Wait for victim to apt-get upgrade



Exploitable Domains

- ***ubunvu.com***
- ***ubuftu.com***
- ***ubunte.com***
- ***ubuntt.com***
- ***ubuntw.com***
- ***ubun4u.com***
- ***ubunuu.com***
- ***dabian.org***
- ***ddbian.org***
- ***debiaf.org***
- ***debicn.org***
- ***debial.org***
- ***debiqn.org***
- ***debien.org***

And more…

Demo Time

This Year's Blackhat Pwnie Awards

- Flip Feng Shui Nominated For Best Cryptographic Attack Pwnie

nominations for the pwnie awards

Nominees for the Pwnie Awards 2017

Pwnie for Best Cryptographic Attack (new for 2016!)

Awarded to the researchers who discovered the most impactful cryptographic attack against real-world systems, protocols, or algorithms. This isn't some academic conference where we care about theoretical minutiae in obscure algorithms, this category requires actual pwnage.

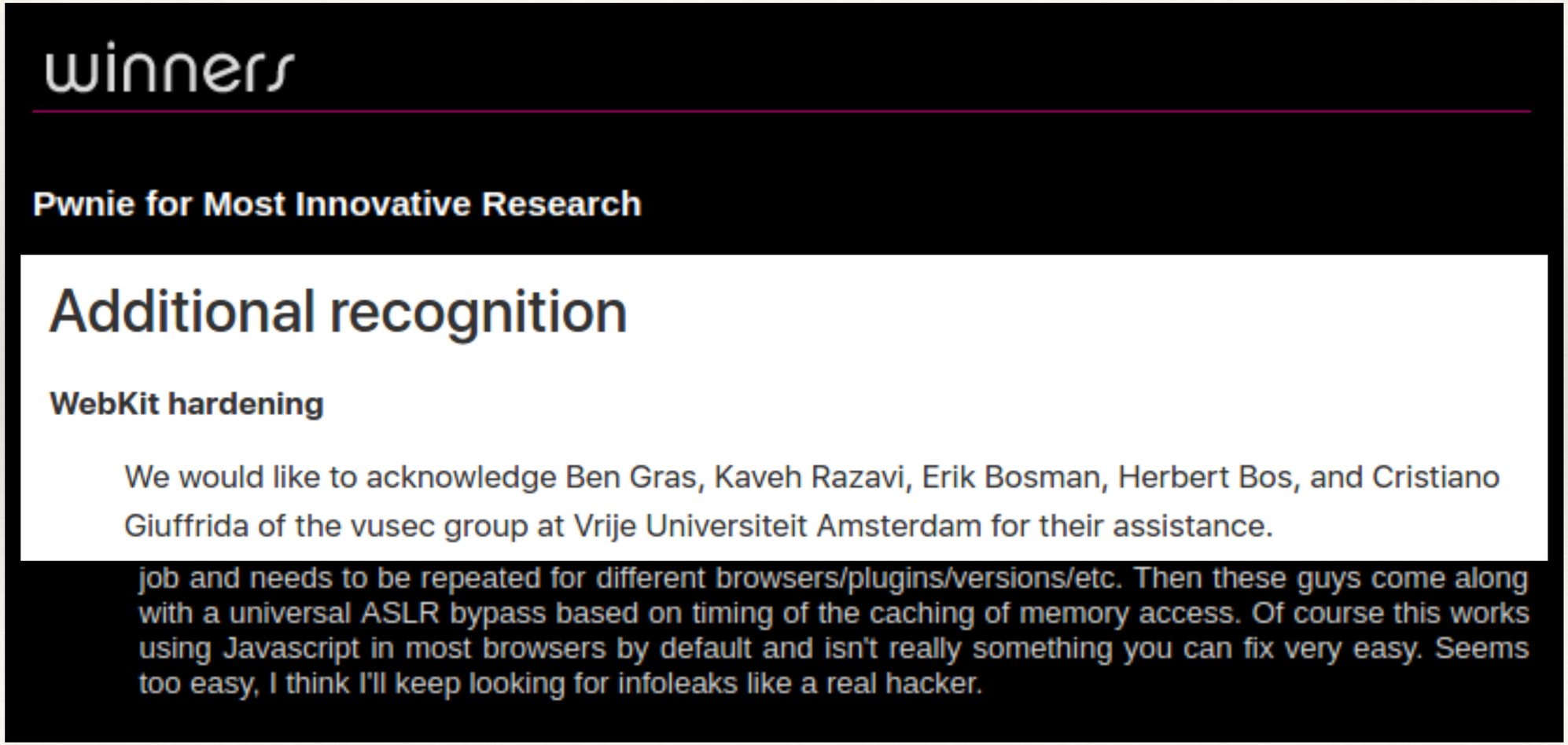
- Flip Feng Shui: Hammering a Needle in the Software Stack

  Credit: Kaveh Razavi, Ben Gras, Erik Bosman, Bart Preneel, Cristiano Giuffrida, Herbert Bos

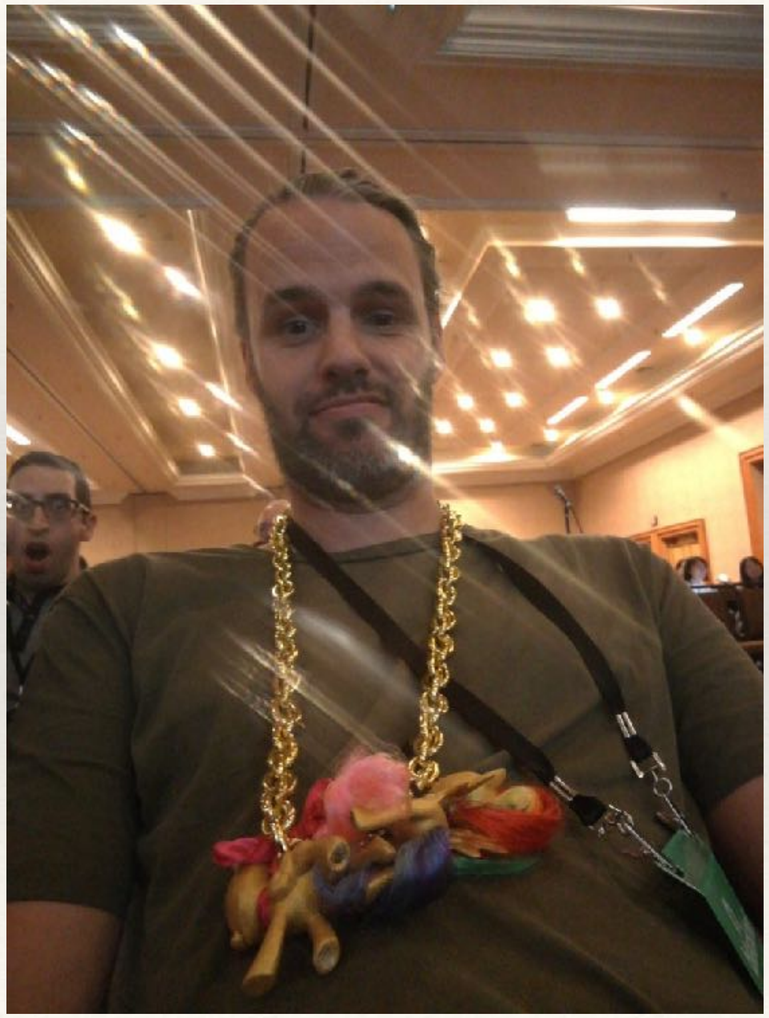
  We introduce Flip Feng Shui (FFS), a new exploitation vector which allows an attacker to induce bit flips over arbitrary physical memory in a fully controlled way. FFS relies on hardware bugs to induce bit flips over memory and on the ability to surgically control the physical memory layout to corrupt attacker-

This Year's Blackhat Pwnie Awards

- AnC Won Most Innovative Research Pwnie



This Year's Blackhat Pwnie Awards



- Victor went to Vegas to collect them for us

Conclusion

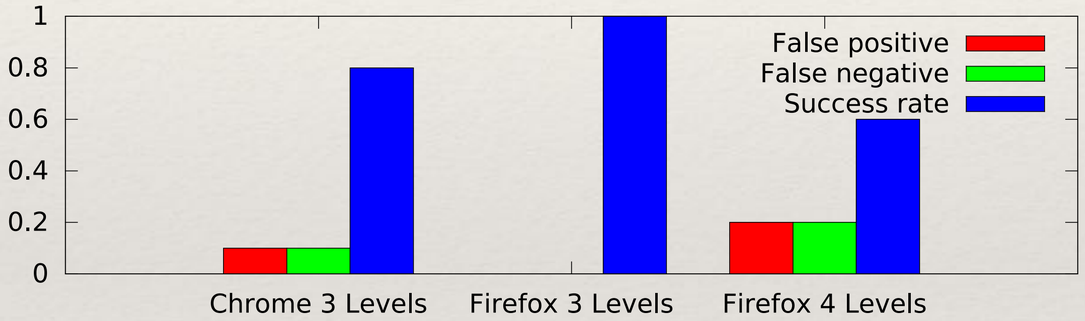
- MMU cache side channel breaking ASLR in sandboxed environments
- Targeting Rowhammer to compromise SSH and GPG
- Hardware is not a flawless abstraction! it can break and leak.

- AnC: www.vusec.net/projects/anc/
- FFS: www.vusec.net/projects/flip-feng-shui/
- `@vu5ec, @gober, @bjg`

Thank you

- Thank you for your attention

Results: reliability



Results: noise

- Repeat measurements vs confidence margin