Introduction of Zenbleed

Akira Kawata
https://akawashiro.com/

Introduction (XMM, YMM, ZMM)

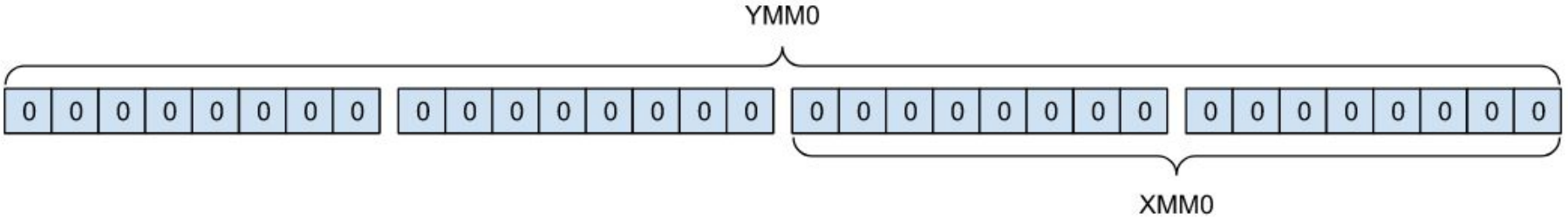
- All x86-64 CPUs have a set of 128-bit vector registers called the XMM registers.
- The 256-bit extended registers are called YMM, and the 512-bit registers are ZMM.
- They’re even used by standard C library functions, like strcmp, memcpy, strlen and so on.

Introduction (strlen in asm)

```
(gdb) x/20i __strlen_avx2
...
   <__strlen_avx2+9>:    vpxor   xmm0,xmm0,xmm0
...
   <__strlen_avx2+29>:   vpcmpeqb ymm1,ymm0,YMMWORD PTR [rdi]
   <__strlen_avx2+33>:   vpmovmskb eax,ymm1
...
   <__strlen_avx2+41>:   tzcnt   eax,eax
   <__strlen_avx2+45>:   vzeroupper
   <__strlen_avx2+48>:   ret
```

Introduction (Store zero to ymm0)

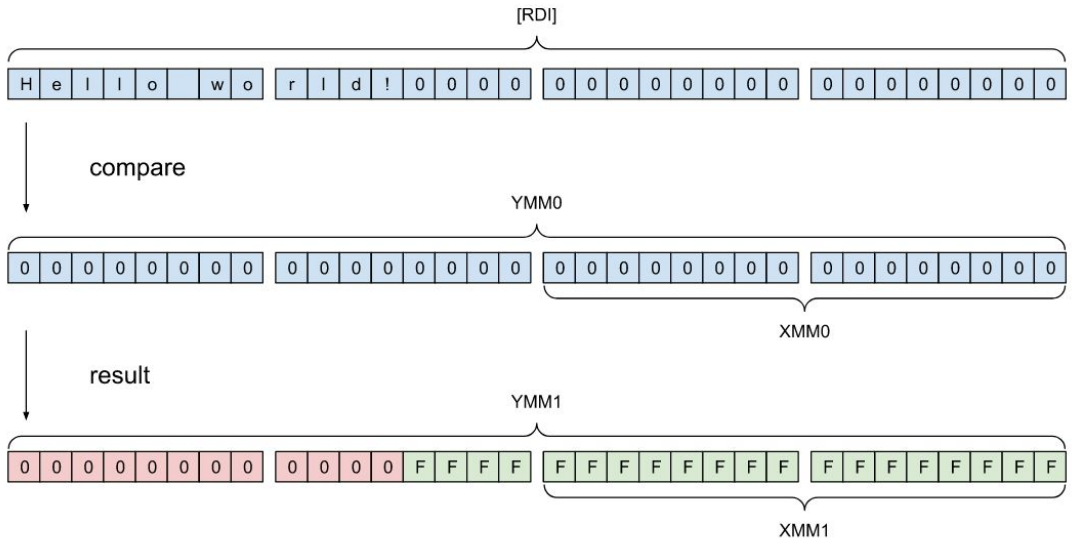
- Initialize ymm0 to zero




```
> vpxor xmm0, xmm0, xmm0
  vpcmpeqb ymm1, ymm0, [rdi]
  vpmovmskb eax, ymm1
  tzcnt eax, eax
  vzeroupper
```

Introduction (vpcmpeqb)

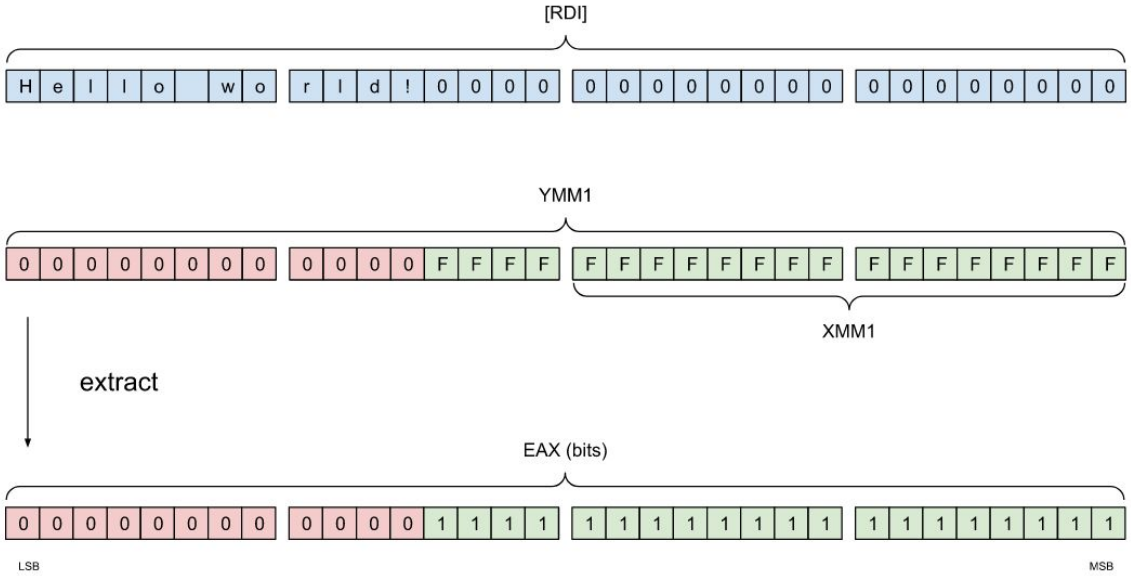
- rdi contains a pointer to our string, so vpcmpeqb will check which bytes in ymm0 match our string, and stores the result in ymm1




```
    vpxor xmm0, xmm0, xmm0
  > vpcmpeqb ymm1, ymm0, [rdi]
    vpmovmskb eax, ymm1
    tzcnt eax, eax
    vzeroupper
```

Introduction (vpmovmskb)

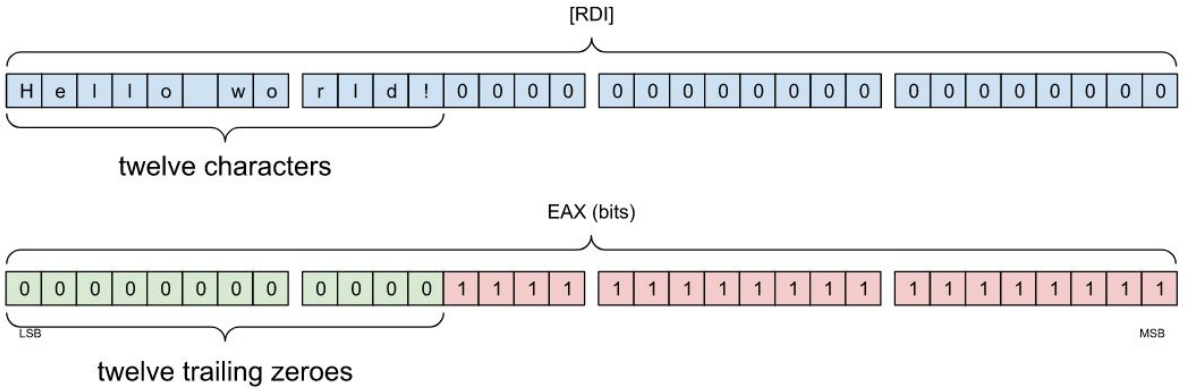
- Extract the result into a general purpose register like eax with vpmovmskb.




```
    vpxor xmm0, xmm0, xmm0
    vpcmpeqb ymm1, ymm0, [rdi]
  > vpmovmskb eax, ymm1
    tzcnt eax, eax
    vzeroupper
```

Introduction (tzcnt)

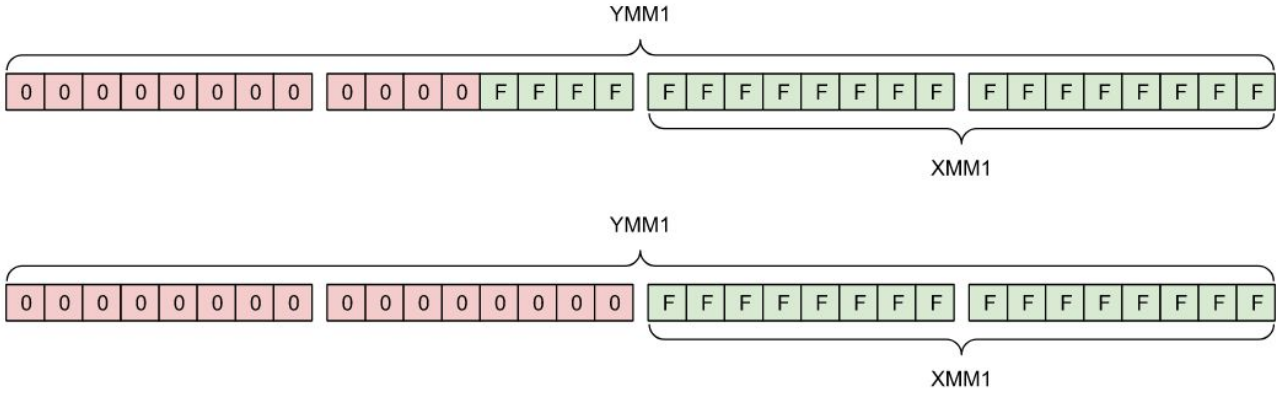
- Get length of “Hello world”.




```
    vpxor xmm0, xmm0, xmm0
    vpcmpeqb ymm1, ymm0, [rdi]
    vpmovmskb eax, ymm1
  > tzcnt eax, eax
    vzeroupper
```

Zeroing Registers

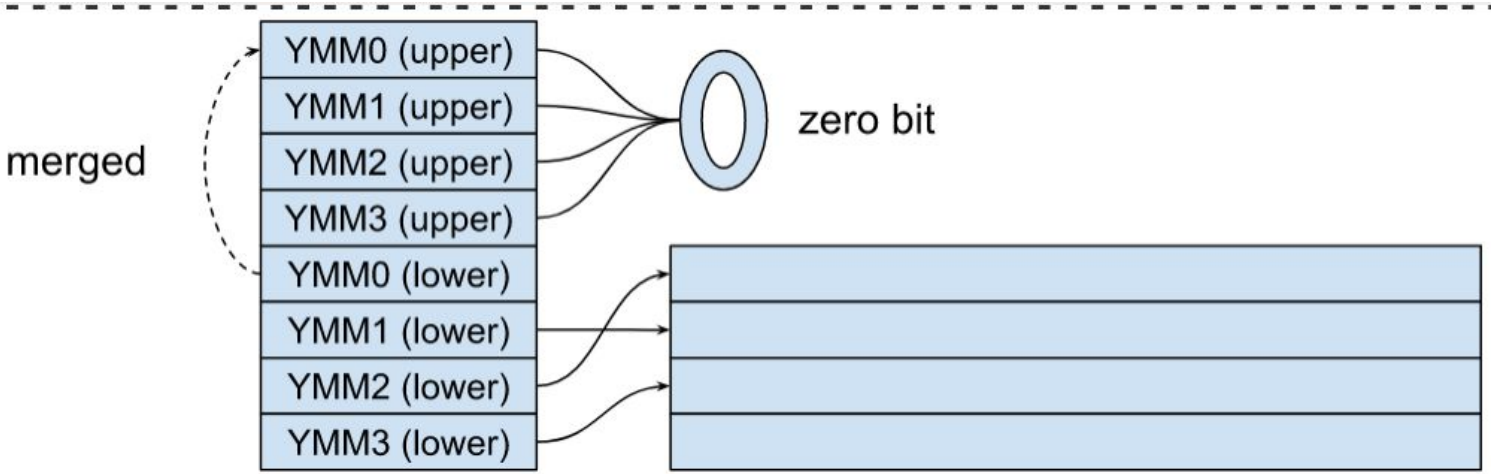
- vzeroupper will zero the upper bits of the vector registers.




```
        vpxor xmm0, xmm0, xmm0
        vpcmpeqb ymm1, ymm0, [rdi]
        vpmovmskb eax, ymm1
        tzcnt eax, eax
      > vzeroupper
```

The Vector Register File

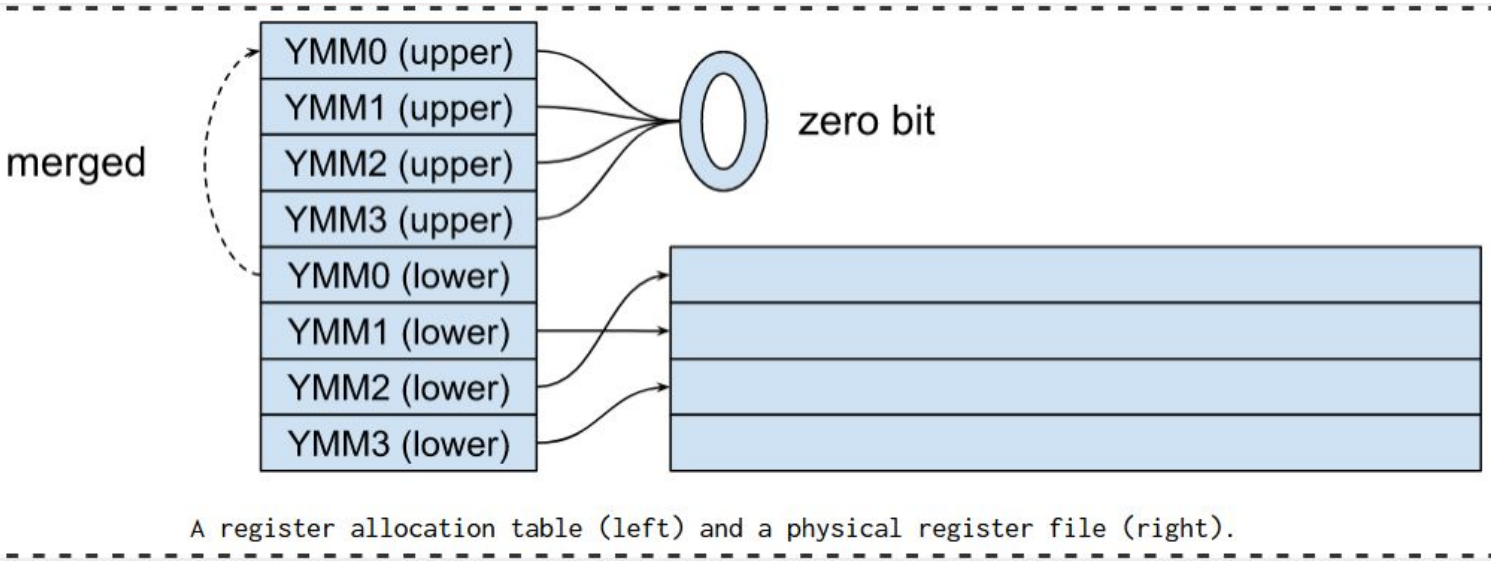
- Your processor doesn’t have a single physical location where each register lives, it has what’s called a Register File and a Register Allocation Table.



A register allocation table (left) and a physical register file (right).

The Vector Register File

- vzeroupper can simply set the z-bit and then release any resources assigned to it in the register file.

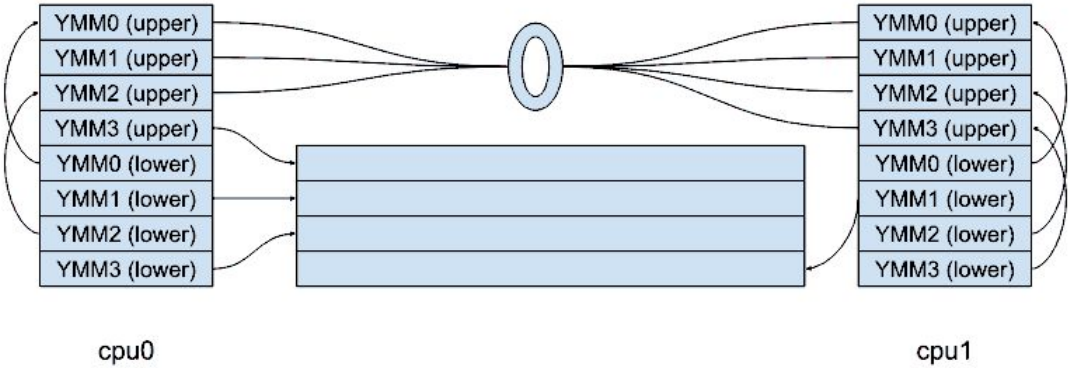


A register allocation table (left) and a physical register file (right).

Speculation

- Modern processors use speculative execution, so sometimes operations have to be rolled back.
- How to roll back vzeroupper?
  - maybe we can just unset that z-bit?
- What happen the roll back isn’t enough?
  - **Boom!**💥

Mechanism of vulnerability



Mechanism of vulnerability

1. cpu0がレジスタファイルを確保
2. 投機実行されたvzeroupperで開放
3. cpu2が開放されたレジスタファイルを確保
4. cpu2がレジスタファイルに機密を書き込む
5. cpu1が2で行った投機実行が巻き戻される
6. 巻き戻しが不完全なためcpu2が書き込んだデータが読める

Exploitation

```
      vcvtsi2s{s,d}    xmm, xmm, r64
      vmovdqa          ymm, ymm
      jcc              overzero
      vzeroupper
overzero:
      nop
```

vzeroupper will be mispredicted and rolled back.

Youtube [Zenbleed (CVE-2023-20593)]