

OS, IO, DMA & Network

Discussion 14,

CS 110: Computer Architecture, Spring 2021

Zihao Diao

Recap: What Does OS Do?

Three Roles

- As referee - 「裁判」 : Allow fair sharing of resources among applications:
 - e.g., Scheduler: Fair share of CPU, disk & network;
- As illusionist - 「魔术师」 : Provide the application with illusion of infinite resources:
 - e.g., VM & scheduler: illusion of having dedicated CPU and all memory;
- As glue - 「胶水」 : Provide the application with standard service interface:
 - e.g., System Calls & File system: standard interface for disk access.

Recap: Input & Output

How to Do IO?

- ISA Support for IO:
 - Special instructions for IO: ISA provide dedicated instruction for IO
 - e.g., Intel x86
 - Memory-mapped IO: CPU *maps* a device to an memory address
 - Use memory read&write instructions for IO
 - e.g., RISC-V: <https://riscv.org/wp-content/uploads/2017/05/riscv-privileged-v1.10.pdf> section 3.5

Recap: Input & Output

Polling & Interrupt

- Polling:

- Consistently check the device for the data to read or write;
- Control register and data register;

- Input: Read from keyboard into a0

```
li    t0, 0xffff0000 #ffff0000
Waitloop: lw    t1, 0(t0)    #control
        andi  t1, t1, 0x1
        beq   t1, zero, Waitloop
        lw    a0, 4(t0)    #data
```

- Output: Write to display from a0

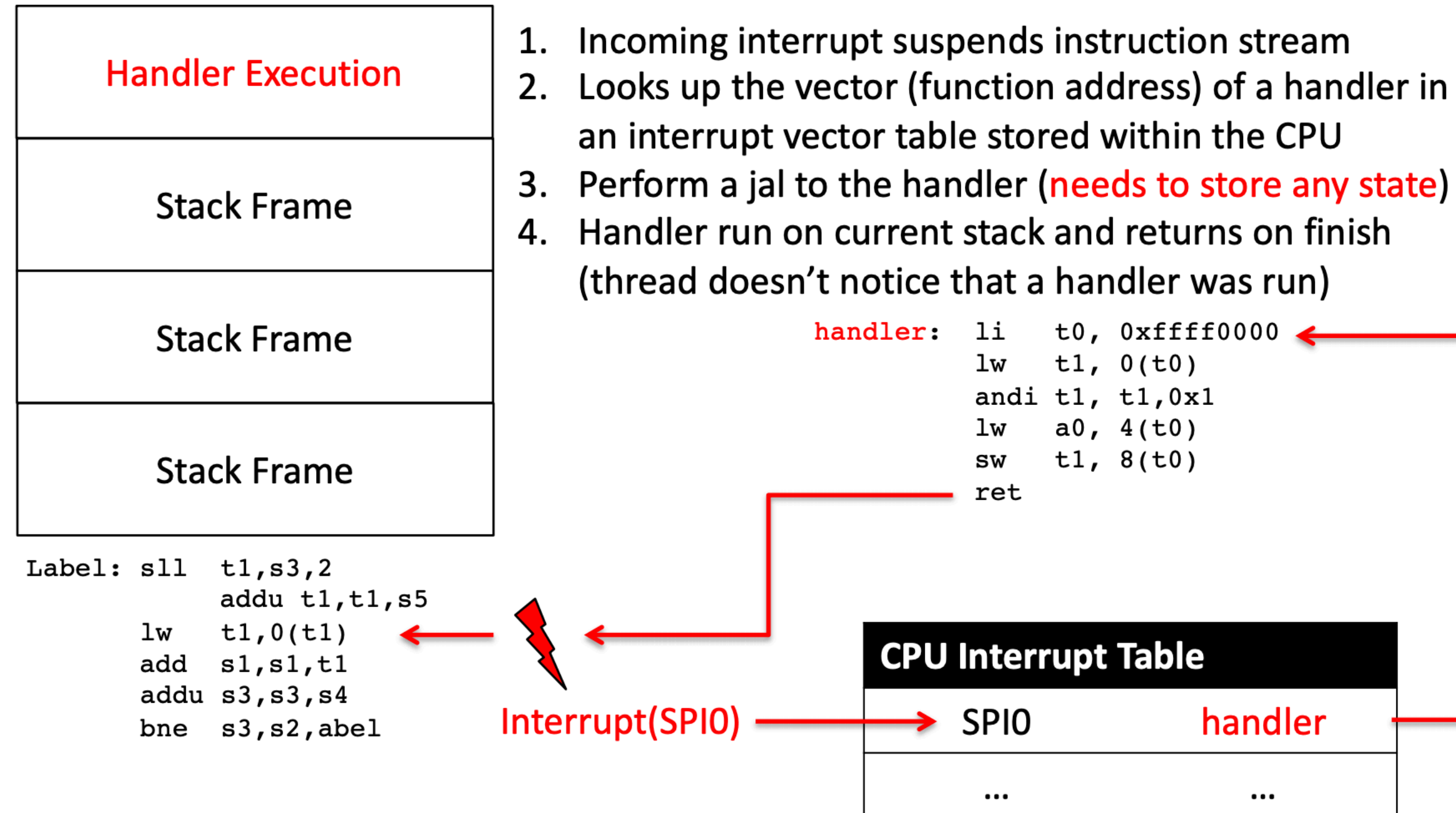
```
li    t0, 0xffff0000 #ffff0000
Waitloop: lw    t1, 8(t0)    #control
        andi  t1, t1, 0x1
        beq   t1, zero, Waitloop
        sw   a0, 12(t0)    #data
```

- Interrupt

Recap: Input & Output

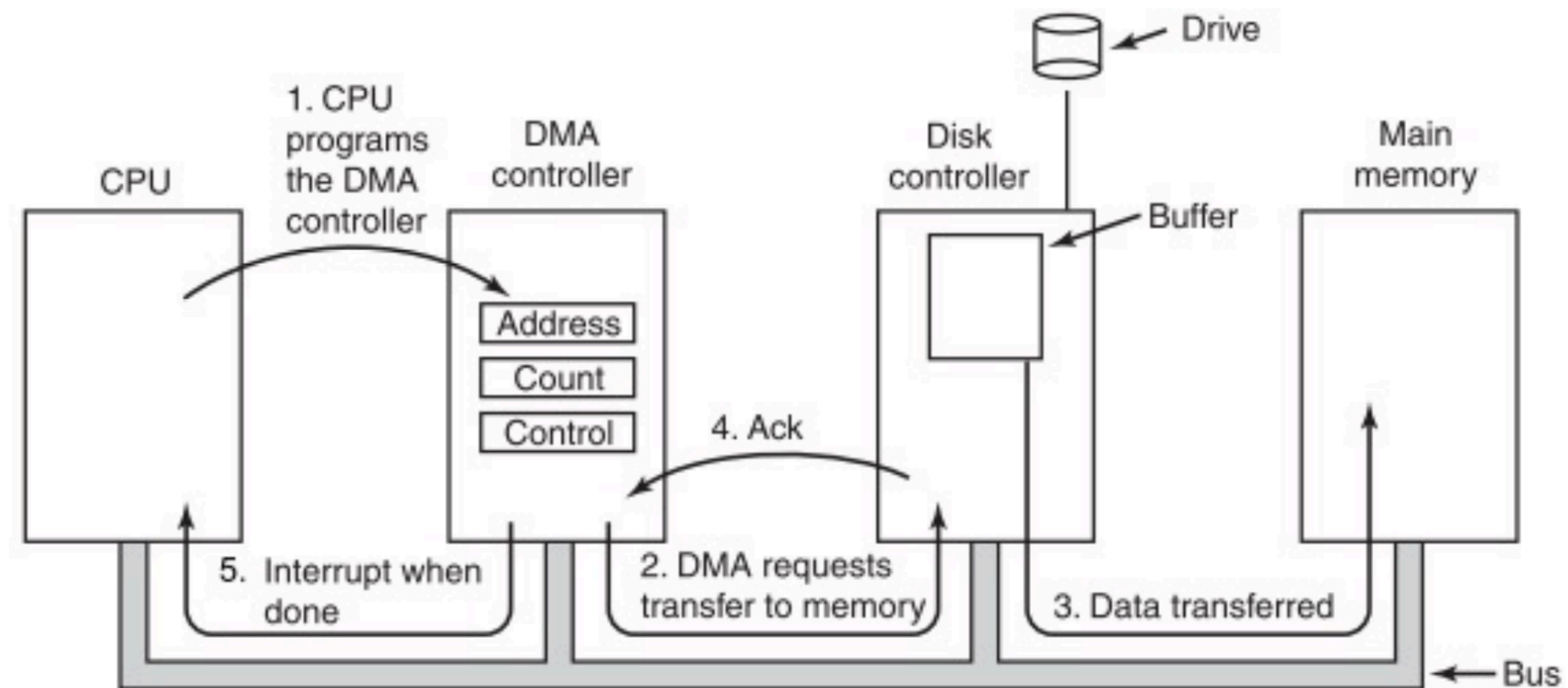
Polling & Interrupt

- Polling
- **Interrupt:** Let the device notify the CPU when a data is ready.



Direct Memory Access

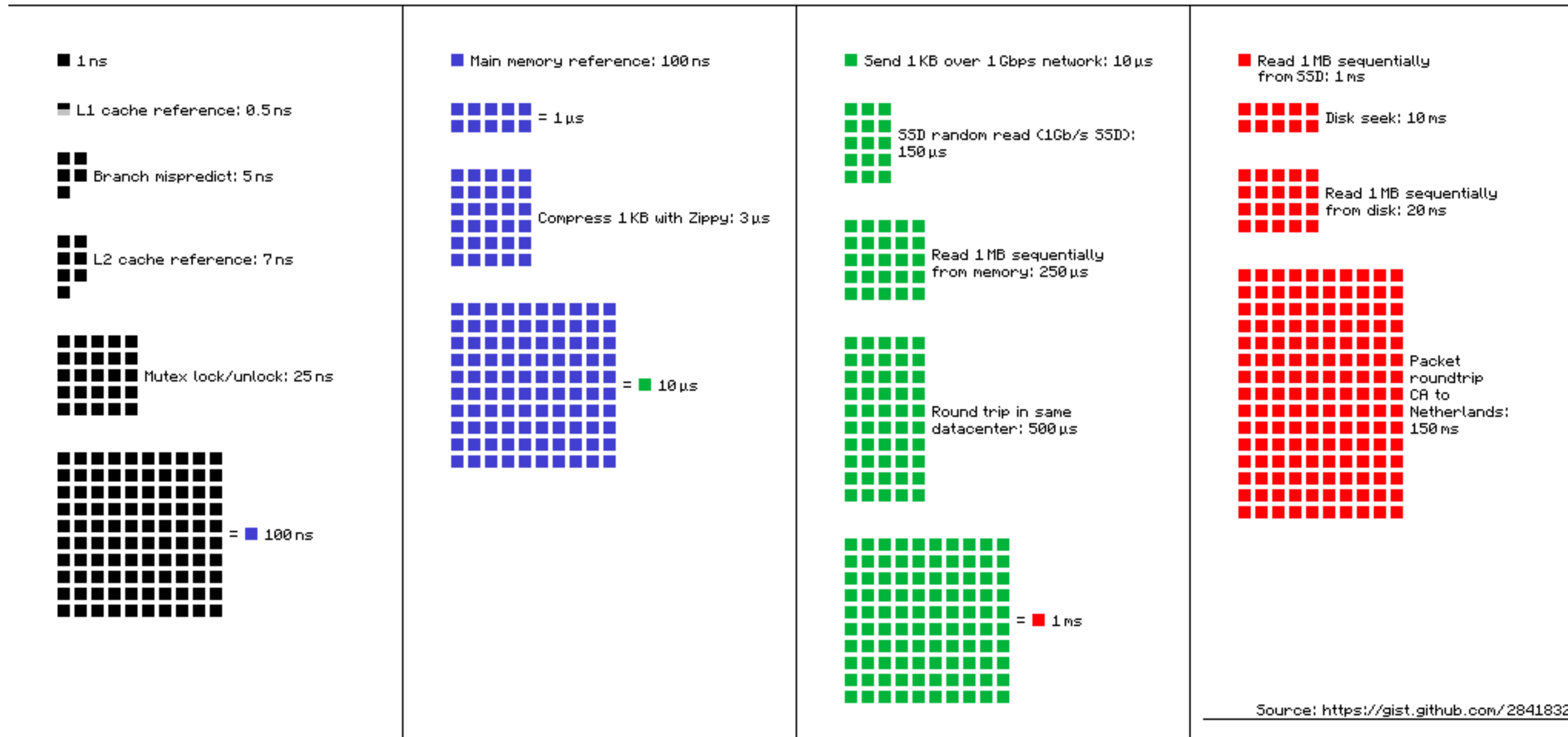
- As of now: CPU will do the read/write
 - IO is slow... (more later)
 - DMA: let the device do the IO instead of the CPU
 - CPU could do other things.



Takeaway: IO is Slow

- IO is very very slow: Compared to Human Scale

Latency Numbers Every Programmer Should Know



<https://i.imgur.com/k0t1e.png>

Source: <https://gist.github.com/2841832>

Takeaway: IO is Slow

- IO is very very slow: Compared to Human Scale
 - Some interesting figure:
 - If one CPU cycle (~ 0.5 ns) is $1/60$ sec - a frame in video...
 - Read from SSD (~ 50 us) will take 30 min, at best;
 - Read from HDD (~ 1 ms) will take half a day, at best;
 - Internet RTT from Shanghai to Boston (~ 300 ms) will take half a year.
 - **In conclusion: We must free CPU from waiting for them...**

Practice

True or False?

- Memory-mapped IO only works with polling.

Practice

True or False?

- Memory-mapped IO only works with polling.

False. Memory-mapped IO also works for interrupt. Polling/Interrupt and MMIO/IO instructions are two sets of orthogonal concepts.

Practice

True or False?

- Interrupt has lower latency than polling in general.

Practice

True or False?

- Interrupt has lower latency than polling in general.

False. Polling has lower latency than interrupt: polling will take fewer steps and interrupt will require the RW request to be queued before the CPU actually do it.

Practice

True or False?

- Interrupt provides higher throughput under the same CPU load.

Practice

True or False?

- Interrupt provides higher throughput under the same CPU load.

True. Despite its higher latency, interrupt provides higher throughput thus is more suitable for high volume data transfer.

Practice

True or False?

- User program can access OS routines with sys calls.

Practice

True or False?

- User program can access OS routines with sys calls.

True. Sys calls provides interface for user program to access some common interface from the OS: e.g., read from the disk.

Q&A