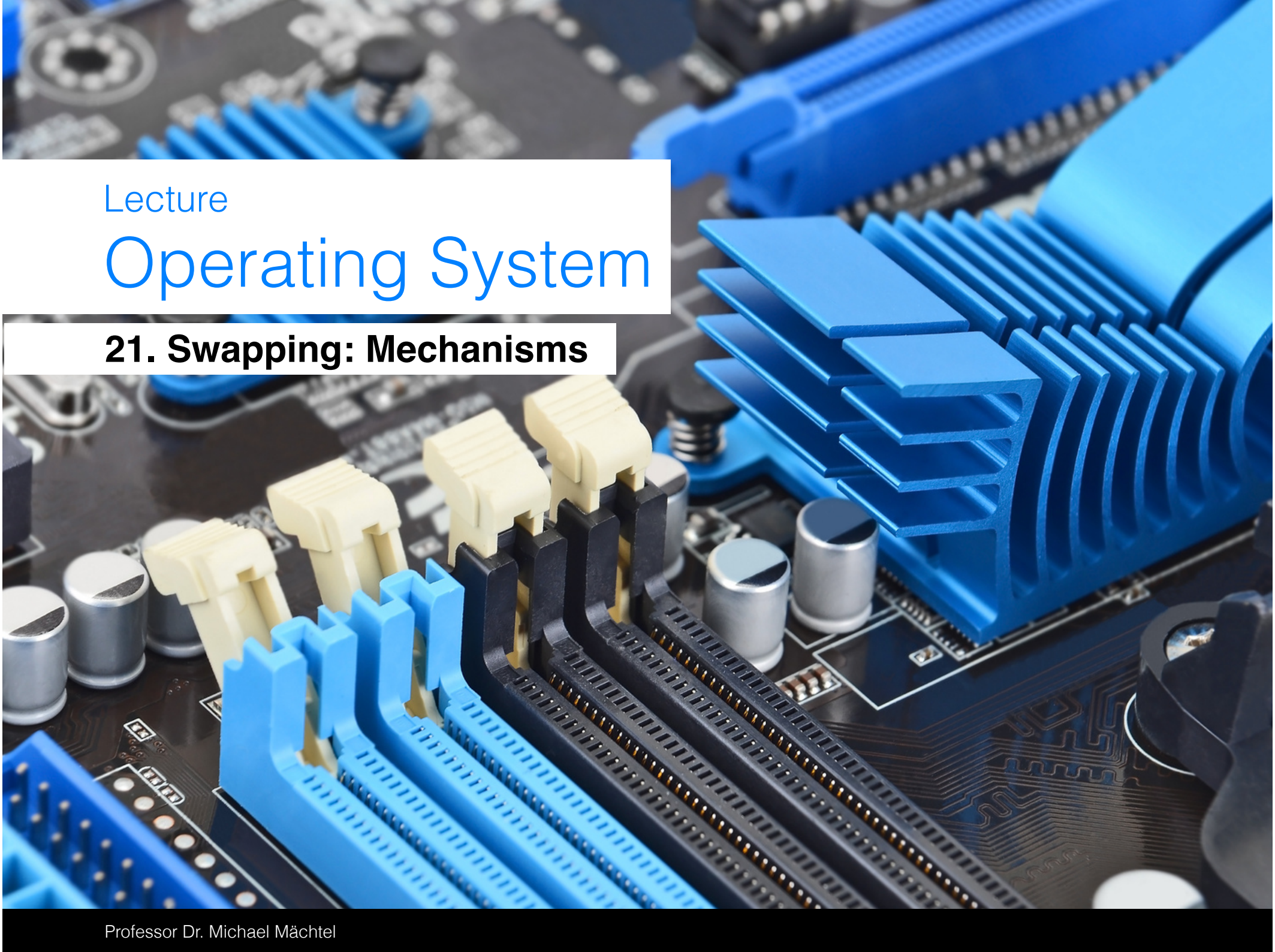


Lecture

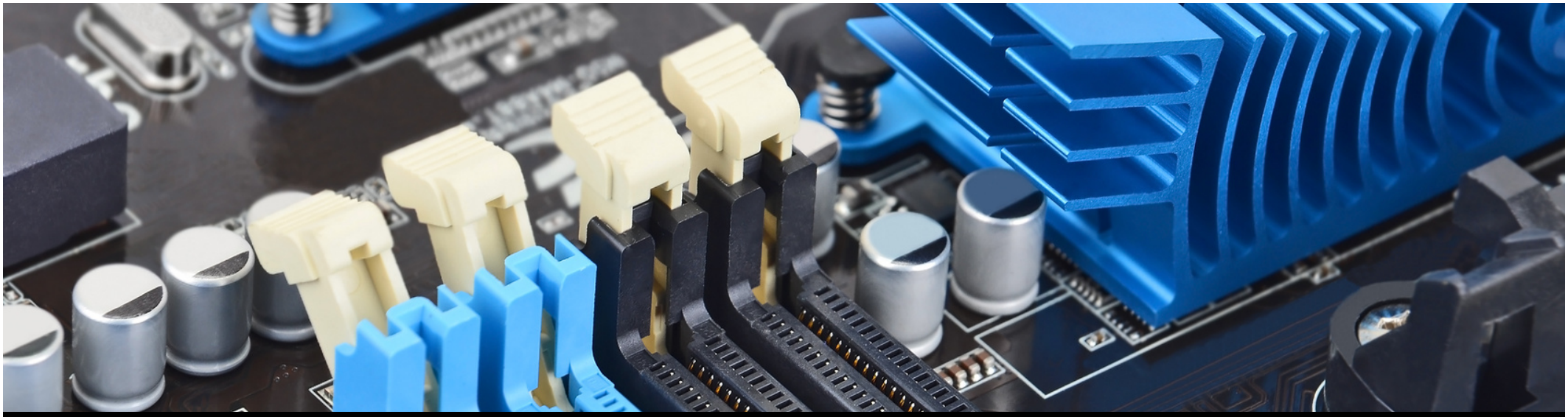
Operating System

21. Swapping: Mechanisms

Professor Dr. Michael Mächtel

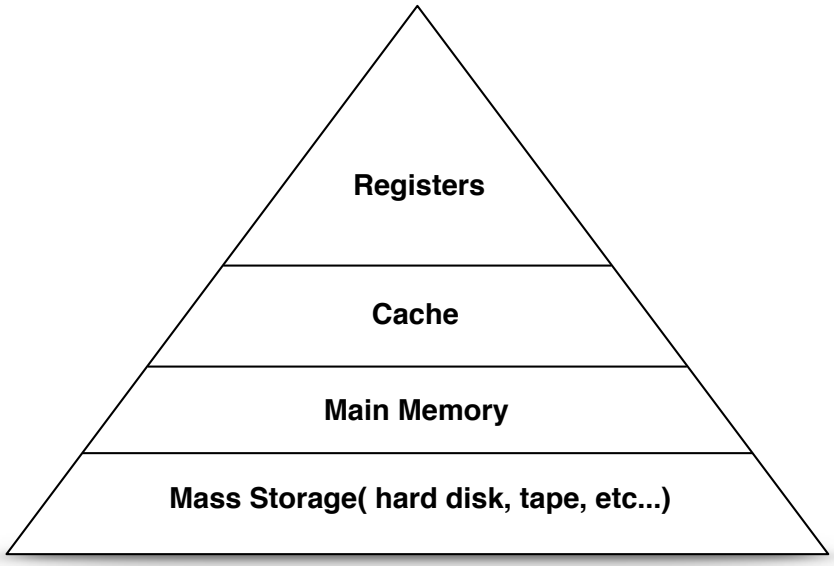
21. Swapping: Mechanisms

1. **To support large address spaces, the OS will need a place to stash away portions of address spaces (e.g. on discs) that currently aren't in great demand**
2. **How can the OS make use of a larger, slower device to transparently provide the illusion of a large virtual address space?**



Beyond Physical Memory: Mechanisms

- Require an additional level in the **memory hierarchy**.
  - OS need a place to stash away portions of address space that currently aren't in great demand.
  - In modern systems, this role is usually served by a **hard disk drive**
- Memory Hierarchie in modern systems:

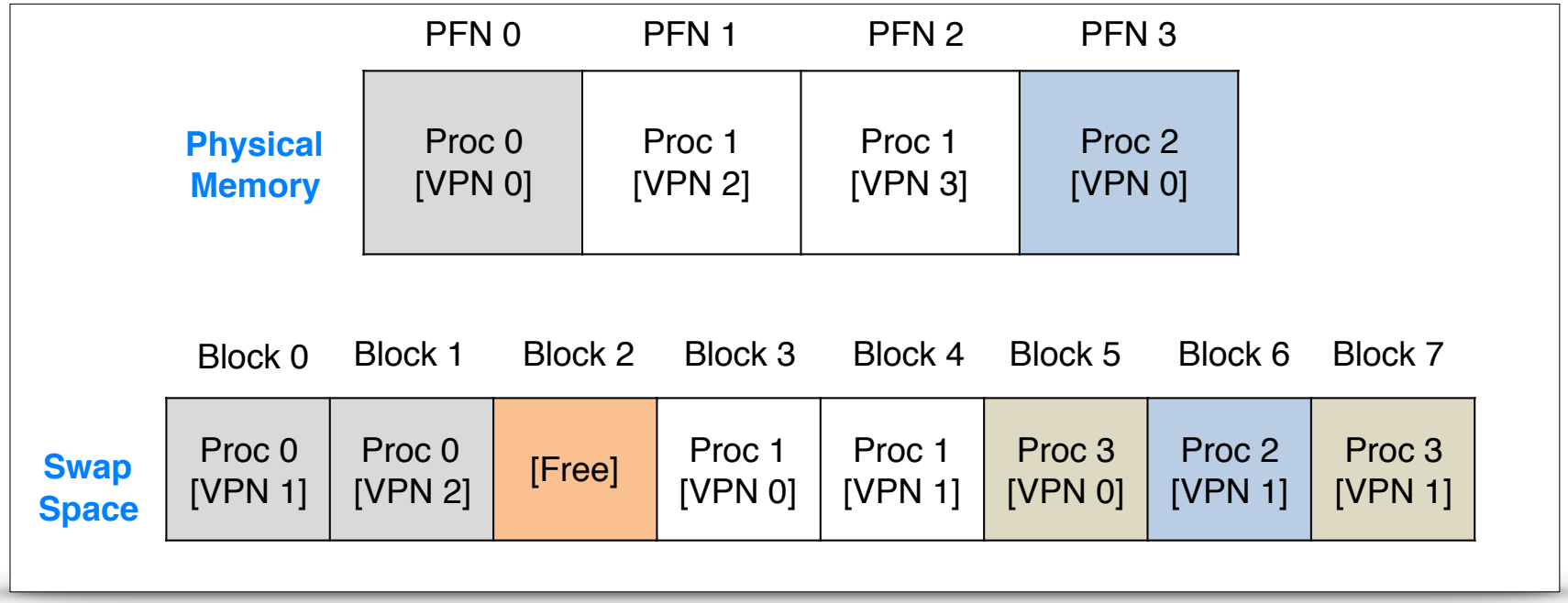


Single large address for a process

- Always need to first arrange for the code or data to be in memory when before calling a function or accessing data.
- To Beyond just a **single process**.
  - The addition of **swap space** allows the OS to support the illusion of a large virtual memory for multiple concurrently-running process

Swap Space

- Reserve some space on the disk for moving pages back and forth.
- OS need to remember to the swap space, **in page-sized unit**



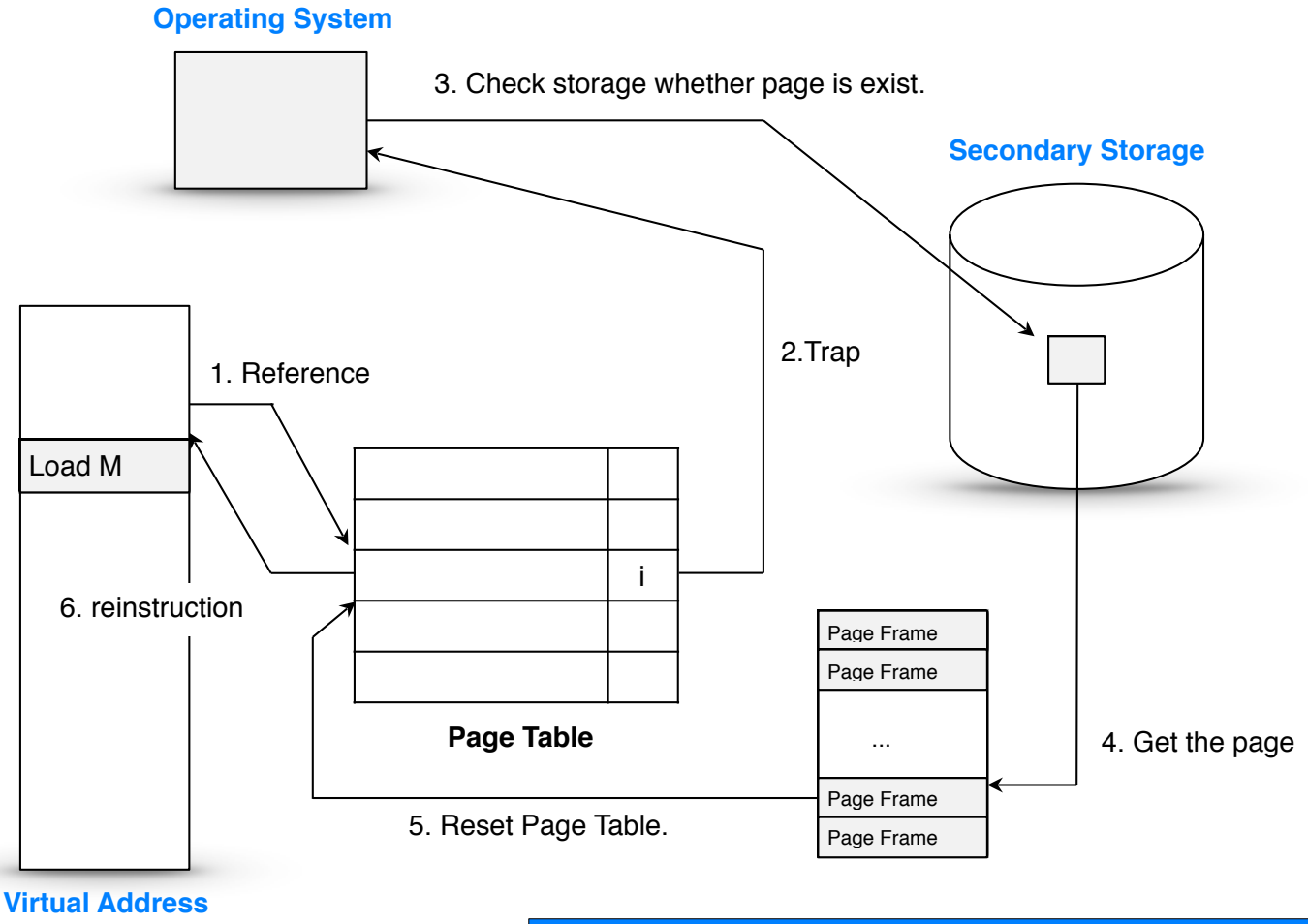
Physical Memory and Swap Space

Present Bit

- Add some machinery higher up in the system in order to support swapping pages to and from the disk.
  - When the hardware looks in the PTE, it may find that the page is not present in physical memory.
- Accessing page that is **not in physical memory**.
  - If a page is **not present** and has been swapped disk, the OS need to swap the page into memory in order to service the **page fault**.

Value	Meaning
1	page is present in physical memory
0	The page is not in memory but rather on disk.

Page Fault Control Flow



When the OS receives a page fault, it looks in the PTE and issues the request to disk.

Page Fault Control Flow

Hardware

```
1:  VPN = (VirtualAddress & VPN_MASK) >> SHIFT
2:  (Success, TlbEntry) = TLB_Lookup(VPN)
3:  if (Success == True) // TLB Hit
4:      if (CanAccess(TlbEntry.ProtectBits) == True)
5:          Offset = VirtualAddress & OFFSET_MASK
6:          PhysAddr = (TlbEntry.PFN << SHIFT) | Offset
7:          Register = AccessMemory(PhysAddr)
8:      else RaiseException(PROTECTION_FAULT)
9:  else // TLB Miss
10:     PTEAddr = PTBR + (VPN * sizeof(PTE))
11:     PTE = AccessMemory(PTEAddr)
12:     if (PTE.Valid == False)
13:         RaiseException(SEGMENTATION_FAULT)
14:     else
15:         if (CanAccess(PTE.ProtectBits) == False)
16:             RaiseException(PROTECTION_FAULT)
17:         else if (PTE.Present == True)
18:             // assuming hardware-managed TLB
19:             TLB_Insert(VPN, PTE.PFN, PTE.ProtectBits)
20:             RetryInstruction()
21:         else if (PTE.Present == False)
22:             RaiseException(PAGE_FAULT)
```

page was both present and valid
so simple grap PFN from PTE

page is not present
so PageFault Handler must run

Page Fault Control Flow

Software

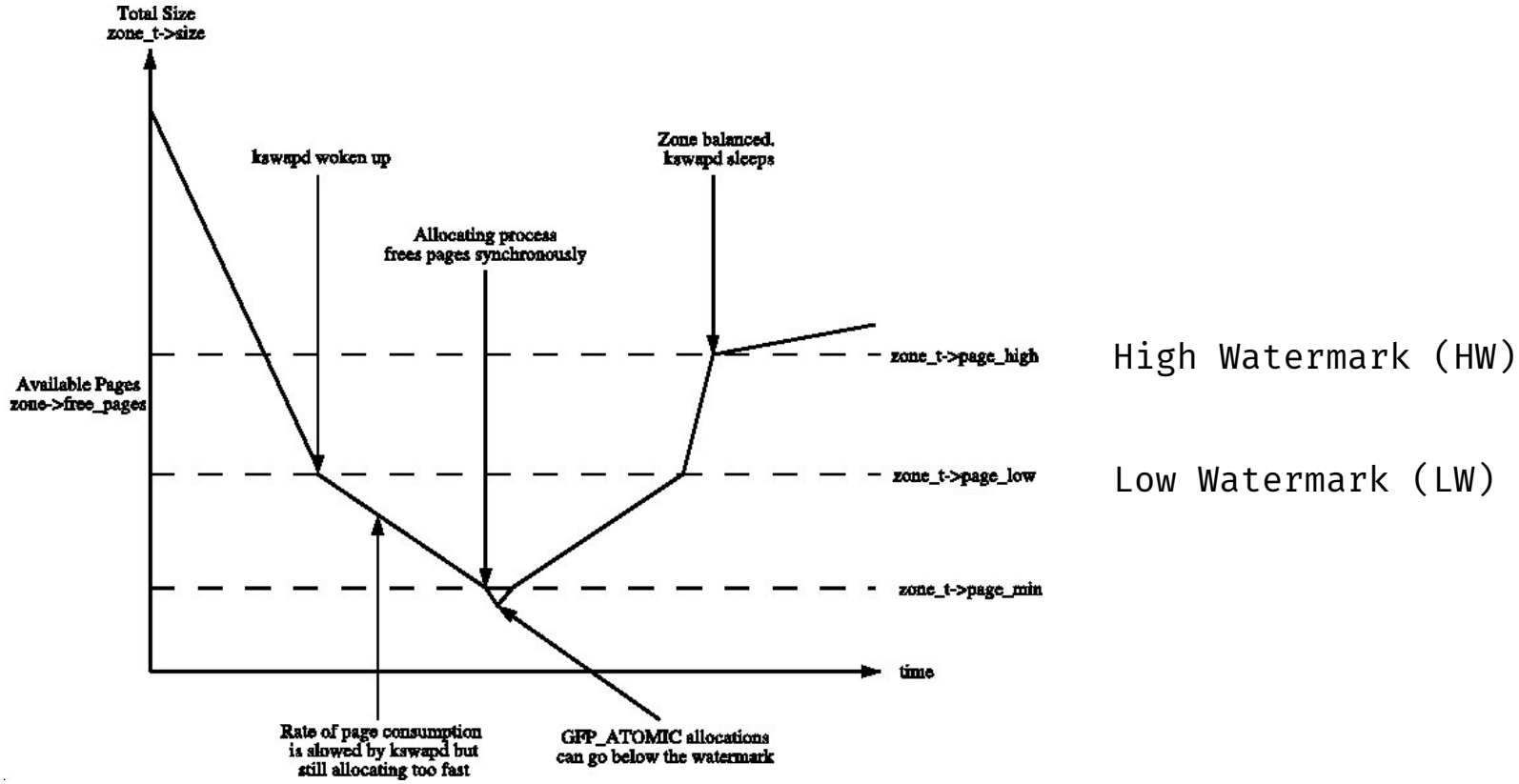
```
1:   PFN = FindFreePhysicalPage()
2:   if (PFN == -1)                  // no free page found
3:      PFN = EvictPage()            // run replacement algorithm
4:   DiskRead(PTE.DiskAddr, pfn)     // sleep (waiting for I/O)
5:   PTE.present = True              // update page table with present
6:   PTE.PFN = PFN                   // bit and translation (PFN)
7:   RetryInstruction()              // retry instruction
```

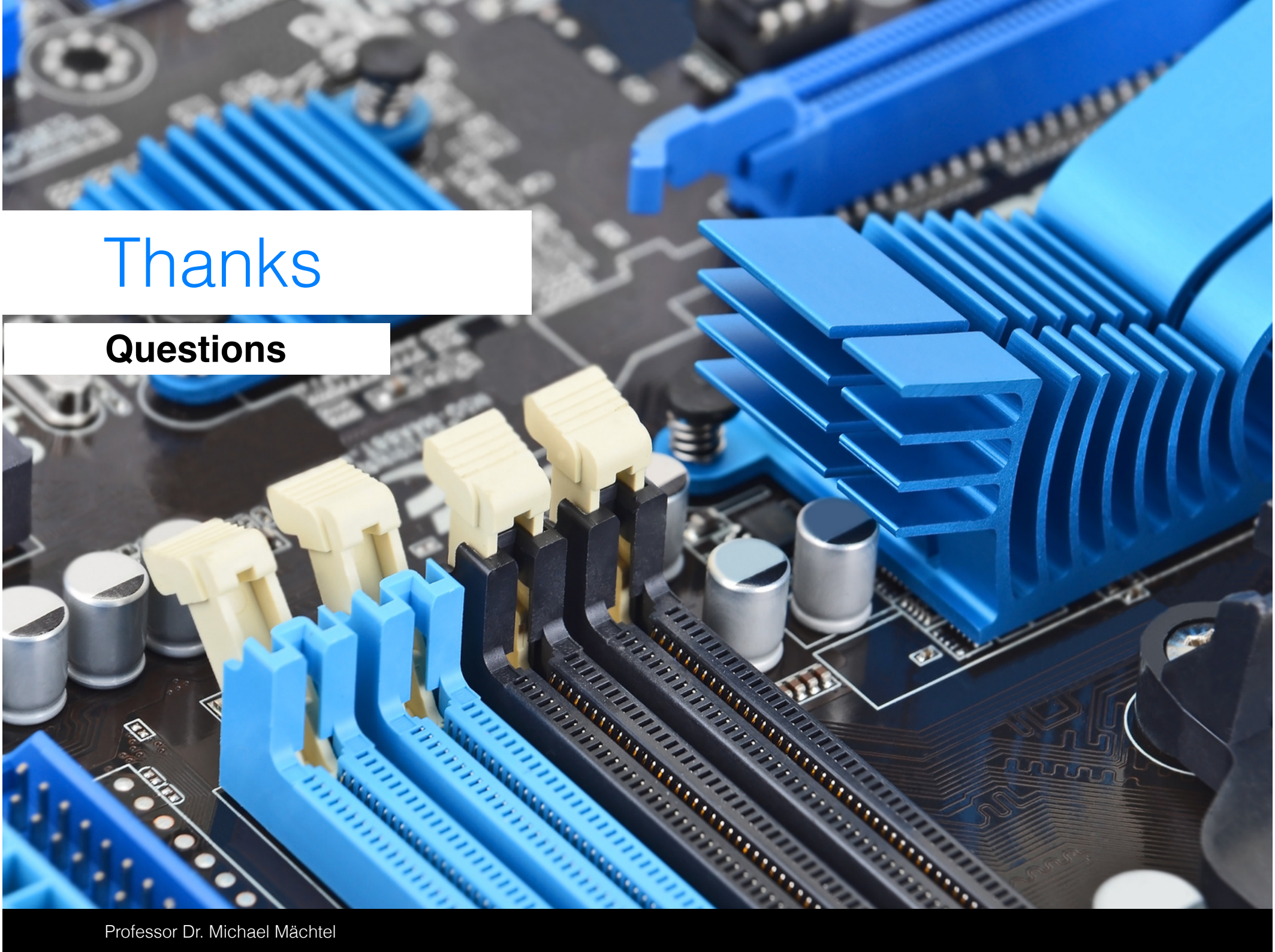
Page Replacement

- The OS like to page out pages to make room for the new pages the OS is about to bring in.
  - The process of picking a page to kick out, or replace is known as page-replacement policy
- OS waits until memory is entirely full, and only then replaces a page to make room for some other page
  - This is a little bit unrealistic, and there are many reason for the OS to keep a small portion of memory free more proactively.

Swap Daemon, Page Daemon

- There are fewer than **LW pages** available, a background thread that is responsible for freeing memory runs.
- The thread evicts pages until there are **HW pages** available.





Thanks

Questions