System software

Loader

Loader

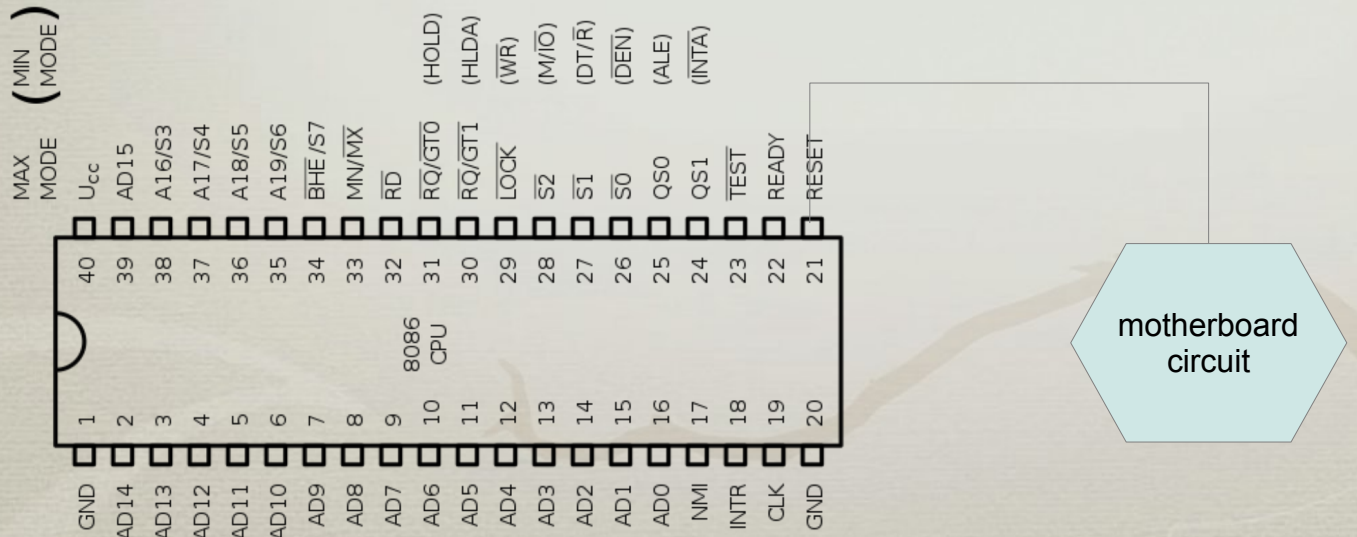
- Loader (*nalagalnik*)
  - Program which from a storage device (disk) **loads** another program into the main memory on a given address and starts its execution.
  - bootloader (z*agonski nalagalnik*)
- Chain loading (verižno nalaganje)
  - loader loads a loader which loads a loader etc.
  - Who loads the first loader?

Boostraping a PC

- An overview of steps
  - motherboard
  - processor
  - BIOS / UEFI
  - boot device
  - boot sector (first sector of boot device)
  - first-stage loader
  - second-stage loader
  - etc.

Boostraping a PC

- Motherboard
  - pressing the power button
  - power supply
  - an electronic circuit on the motherboard
  - signal to the processor via the RESET pin



Vaje SPO, © JM

Boostraping a PC

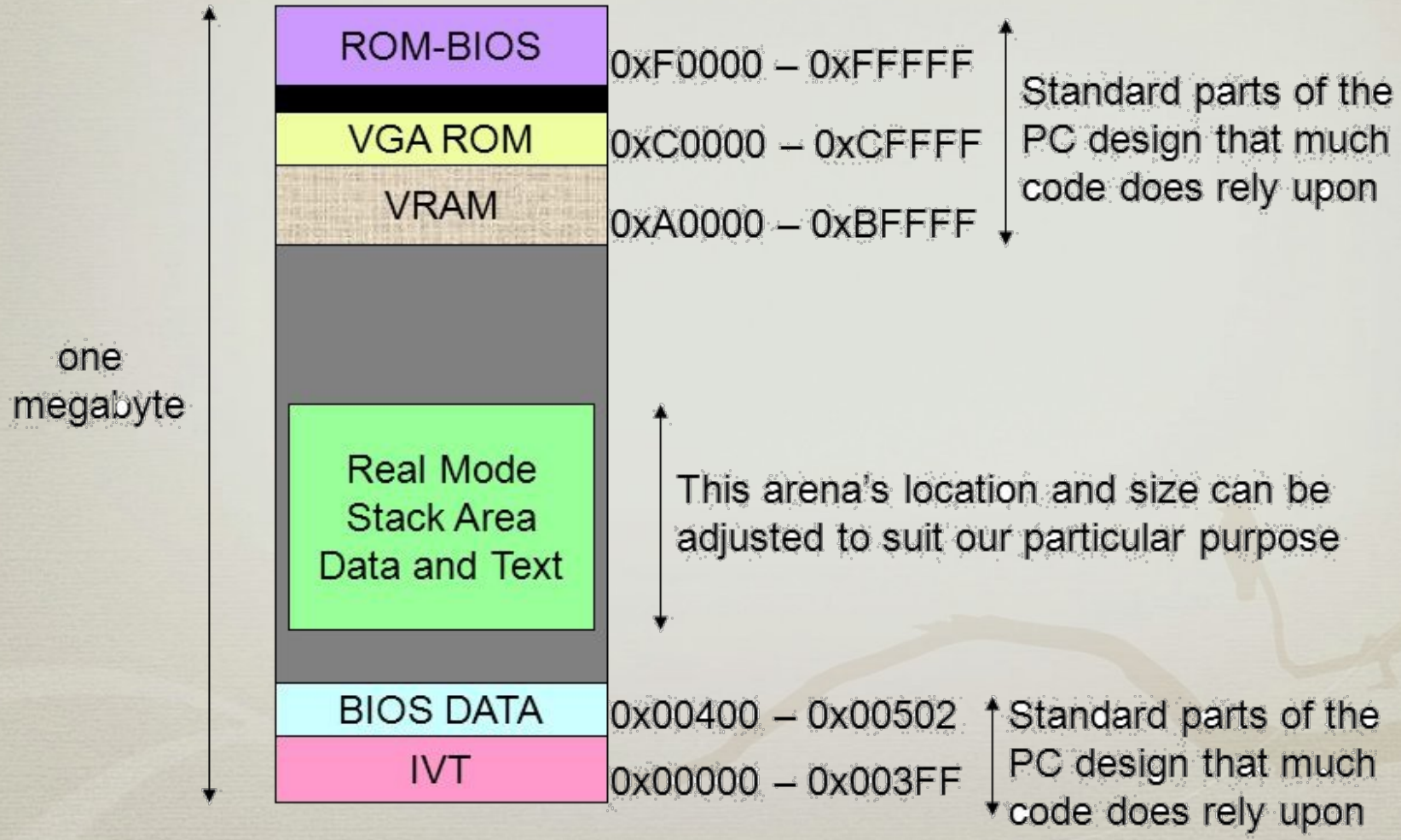
- Processor: 8086, 8088, 80186
  - *unusual* segmented memory scheme
    - address = segment * 16 + offset
    - segment registers
      - CS (code segment)
        - used together with IP (instruction pointer = program counter)
        - jump instructions
      - DS (data segment)
        - used for data access
        - load and store instructions
      - SS (stack segment)
        - used together with SP (stack pointer)
        - stack instructions

Boostraping a PC

- Processor: 8086, 8088, 80186
  - 20 bit addressing (1 MB of memory)
    - no memory protection
    - 16 bit registers (data and addresses)
    - max_address = 0xFFFF * 0x10 + 0xFFFF = 0x10FFEF
      - 1 MB + 64 kB – 16 B = 1114096 B
      - almost 64 kB more than 1 MB
    - addressing modulo 1 MB
      - wrap around for addresses higher than 1 MB
      - 21st address line (A20) was simply ignored (not present)

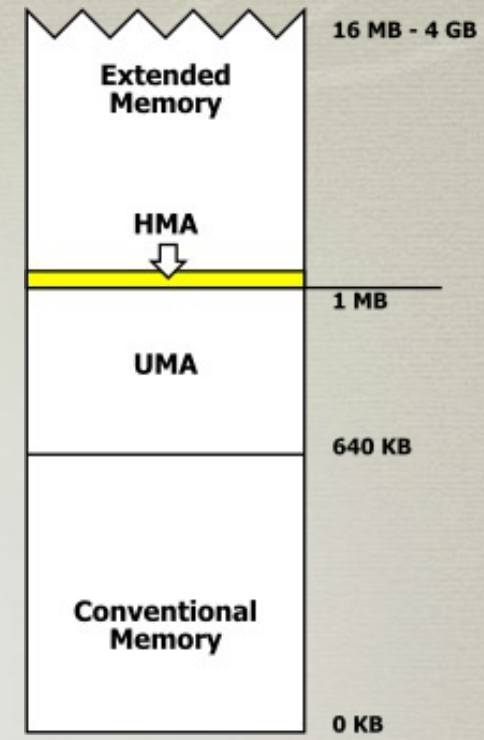
Boostraping a PC

The 8086 memory-map



Vaje SPO, © JM

Boostraping a PC

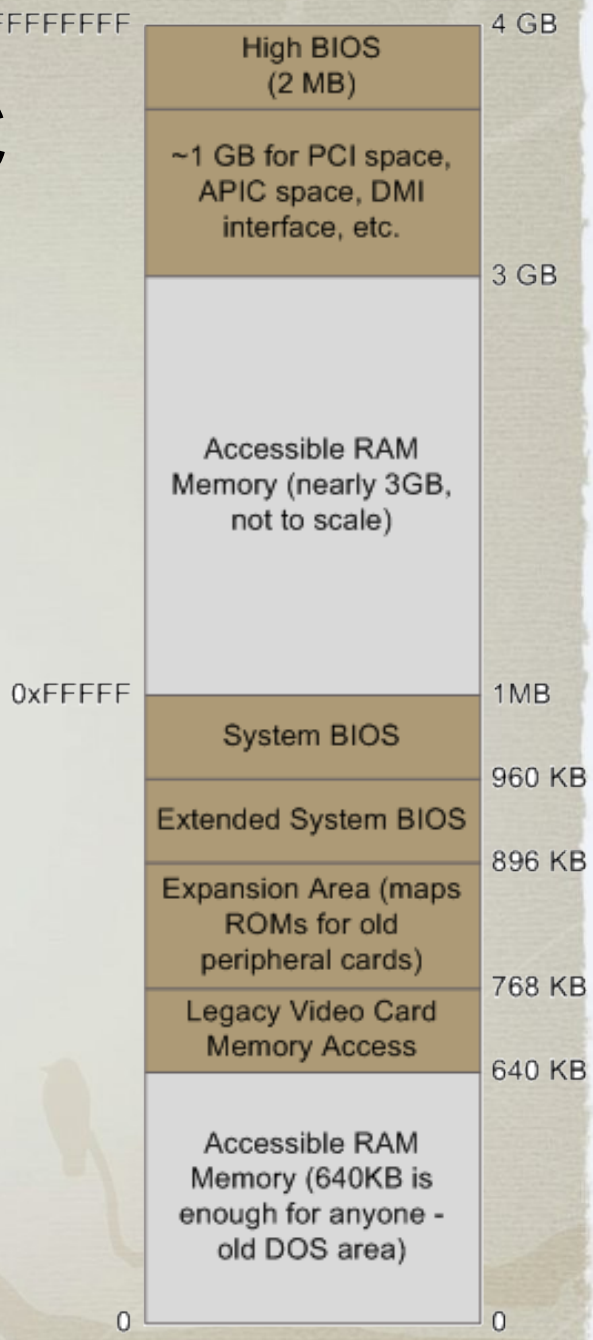


- Processor: 80286
  - 24 bits addresses, 16 MB
    - high memory area
      - no more wrap around, compatibility problems
      - 64 kB – 16 B of memory over 1 MB
    - A20 gate
      - a circuit on the motherboard placed between CPU and motherboard to force the 21st address line to zero
      - originally enabled or disabled programatically through keyboard controller
  - protected mode
    - but hard to use, so modern OSes were not developed
    - address = descriptors[segment] + offset

Vaje SPO, © JM

Boostraping a PC

- Processor: 80386 (i386, IA-32)
  - 32 bit data and addresses, 4 GB
  - real mode
    - compatibilty with 80186 are previous
  - protected mode
    - „easy“ to use modern features
    - virtual memory, memory protection
  - virtual 80186 mode
    - run *real mode* applications in *protected mode*



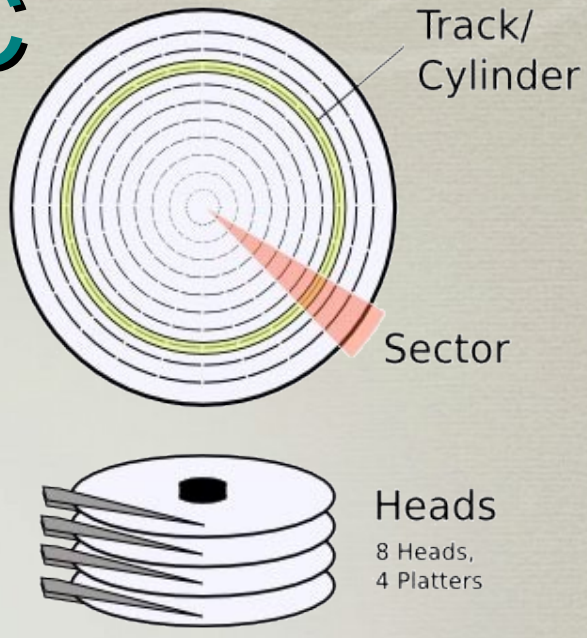
Boostraping a PC

- Processor: x86 family
  - resets and initializes
    - signal on the RESET pin
  - goes into the **real mode**
    - 1 MB of memory
  - reset vector (initial value of PC)
    - CS:IP = 0xFFFF:0x0000 = 0xFFFF0 … before 80286
    - this address containes JMP instruction (to BIOS routine)
    - 80286, 80386 and above initialize CS:IP a bit differently, but in real mode they start at the same 0xFFFF0 address

Boostraping a PC

- BIOS – *basic input/output system*
  - executes POST – *power-on self test*
    - detection, initailization and diagnostics of devices:
      - memory, buses, video, disks, etc.
  - proceeds to the next stage
    - finds a boot device
    - reads its first (boot) sector (512 bytes) to the main memory at 0x07C00
    - executes the code (sector) by jumping to the address 0x07C00

Boostraping a PC



- Boot sector
  - the first sector of the boot device
  - devices without partitioning
    - floppy disks
    - the first sector is called VBR – volume boot record
  - devices with partitions (partition table)
    - hard disks
    - the first sector is called MBR – master boot record
    - the first sector of each partition is called VBR

Boostraping a PC

- Boot sector: MBR and VBR
  - contains first-stage bootloader
  - and partition table (only MBR)

- boot loader and other info
  - bootstrap code, max 440 B
  - disk signature, 4 B
  - two null bytes, 2 B
- primary partition table, 4x 16 B
  - 1st partition
  - 2nd partition
  - 3rd partition
  - 4th partition
- signature: 0x55, 0xAA, 2 B

```
dd if=/dev/sda of=mbr.bin bs=512 count=1
hexdump -vC mbr.bin
ndisasm -o7C00h mbr.bin
```

Boostraping a PC

- First-stage bootloader
  - MBR bootloader
    - checks partition table and finds active partition
    - loads its first sector (VBR) and executes it
  - VBR bootloader
    - finds the second-stage bootloader
    - reads its image into the main memory
    - executes it

Boostraping a PC

- Second-stage bootloader
  - knows how to load OS (kernel)
  - may switch to protected mode
  - executes the loaded OS kernel
  - examples
    - LILO – LInux LOoader
    - NTLDR – NT LoaDeR
    - GRUB – GRand Unified Bootloader
    - BOOTMGR – Windows Boot Manager
  - boot manager
    - advanced options, interactive menus, …

System software

SIC/XE loader

SIC/XE loaders

- Loader in Java
  - e.g. „Load object file“ in the SicTools Simulator
- Loader in SIC/XE machine code
  - program in assembly
  - loads a program from a specific device

Type of loaders

- Absolute loader
  - program is loaded to a predefined address
  - directive START *address*
- Relative loader
  - program is loaded to an arbitray address
  - support for relocation (*prenaslavljanje*)
  - directive `START 0`

Format of `obj` file

- Textual (human-friendly) format (ASCII).
- sequence of records
  - each record is written in one line
  - record types: `H`, `T`, `E`, `M`, `D`, `R`.
- hexadecimal representation
  - one byte is represented with two hex digits
    - one hex digit is sometimes called nibble

B7

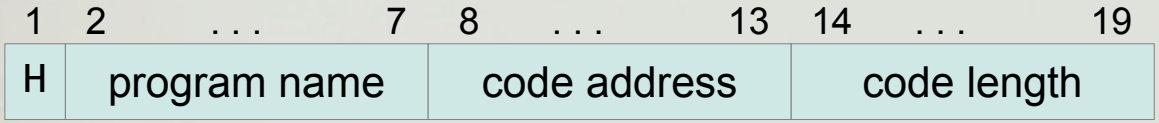
$11*2^4 + 7 = 183$

B7E3

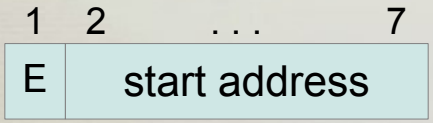
$11*2^{12} + 7*2^8 + 14*2^4 + 3 = 47075$

Format of `obj` file

- Records H (head) and E (end)
  - appear only once
  - H is the first record, E is the last



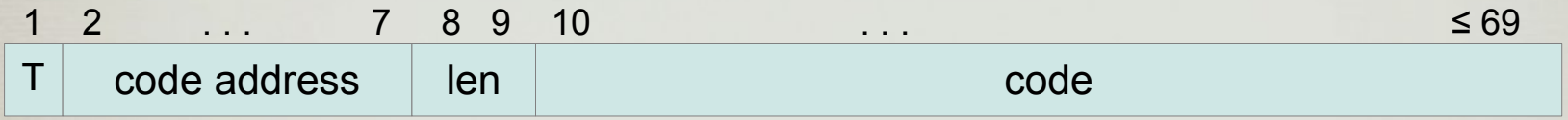
- Code address gives the location where the code is loaded.
- Address and length relate to the whole program.



- Start address is the address where the execution starts.
- After loading we jump to this address.

Format of `obj` file

- Record T
  - code, text



- Address and length relate to this record.
- Code must be part of the area specified with `H` record.
- Code is hexadecimaly encoded.

Format of obj file

- Records D and R
  - export and import of symbols

1	2 . . . 7	8 . . . 13	14 . . . ≤ 73
D	name	value	other pairs: name, address

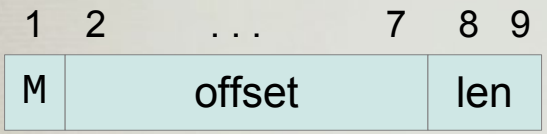
- Name and value of the symbol we are exporting.

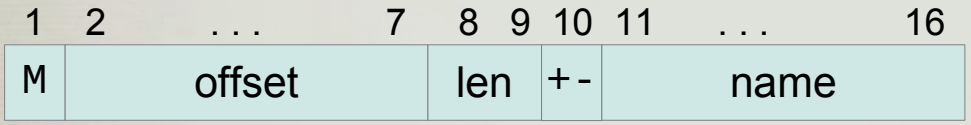
1	2 . . . 7	8 . . . 13	14 . . . ≤ 73
R	name	other names	

- Name of external symbol we are importing.

Format of `obj` file

- Record M
  - support for relocation





- Offset from the code start.
- Length is given in nibbles.
- Name of the symbol.