Oprea Sergiu-Daniel, Gr 30322

Homework-Lab2

Answer the following questions:

1. What is a microprocessor?

A microprocessor is an integrated circuit that can be programmed to process data and send messages and directions to other electronic systems.

1. What is a register?

Registers are temporary storage locations inside the CPU that hold data and addresses.

1. Which are the x86 16 bit registers?

The 16 bit registers are AX, BX, CX, DX.

1. Which are the x86 32 bit registers?

The X86 architecture contains 8 general purpose registers on 32 bits each, as follows: EAX, EBX, ECX, EDX, ESI, EDI, EBP, ESP. The first four are mostly used in arithmetic and logic operations (for keeping operand values), while the next four are used for keeping and calculating memory addresses for operands.

1. Which are the general registers?

Register **EAX** – **accumulator** in most arithmetic and logic instructions, usually keeps one operand and afterwards the result; for multiply and divide instructions, this register is used as default for the first operand and result.

Register **EBX** – **base** register, used in arithmetic and logic instructions and also for calculating the base address when used.

Register **ECX** – **counter,** used in arithmetic and logic instructions, as default in loop instructions.

Register **EDX** – **data,** used in arithmetic and logic instructions, for ports in in/out instructions and also as an extension of register EAX for multiply and divide operations.

Registers **ESI** and **EDI** – **source** and **destination index,** used for indexes for arrays, when calculating the indexed addresses.

Register **EBP** – **base pointer,** used for calculating the addresses in base addressing.

Register **ESP** – **stack pointer,** for addressing the stack of the running program.

1. Which are the special registers?

Register **PC** – **program counter** **–** keeps the address of the following instruction, is not directly addressable (by name), but its contents can be modified by executing jump instructions

Register **PSW** – **program status word –** keeps status indicators/flags of the CPU

**Segment registers** – used for calculating the addresses of operands and instructions, they divide the memory in segments and there are 6 segment registers:

* + Register CS – **code segment,** used for addressing instructions, it contains the start address of the code segment (the source code of the program)
  + Register DS – **data segment,** used for addressing the operands from the memory
  + Register SS – **stack segment,** used for addressing the memory of the program stack
  + Register ES – **extra-data segment,** also used for data, like the DS

Registers FS and GS – similar to ES

1. What are the flags? Give some examples

Flags are an important component of microprocessors as they register the outcomes of calculations and actions. They are important as they keep status of the results from the operations made by the CPU.

* + Flags that store the result of an instruction
    - ZF – zero result
    - SF – the sign of the result
    - OF – overflow of quantity/size in the last arithmetic instruction
    - PF – parity of the result
    - CF – carry flag
    - AC – auxiliary carry flag (after the first 4 bits)
  + Flags that control the work procedure of the CPU:
    - IF – interrupt flag, if set to 1, then all maskable interrupts are blocked; otherwise all are validated

DF – direction flag, shows the travel direction on arrays (ascending or descending memory addresses)

1. Which is the role of segment registers?

Segment registers are used for calculating the addresses of operands and instructions, they divide the memory in segments and there are 6 segment registers.

1. How is the memory organized in real mode? How is the physical address calculated?

In this type of addressing, the maximum amount of RAM is 1 MB. This memory is divided into fixed-size segments of 64 KB each. The physical memory location is calculated as the sum between the segment address and the offset address. The segment address is obtained by multiplying the content of the register with 16. The offset address is calculated based on the addressing type and the address contained in the operand of the instruction. The result of the sum is the physical address on 20 bits, enough for addressing a total memory space of 1 MB.

• Maximum size for the RAM is 1 MB

• A segment must start at an address multiple of 16

• A segment can have a maximum size of 64 KB

• Segments can overlay partially or completely

• The same physical address location can be expressed through several pairs of addresses (segment : offset)

• Few possibilities for protecting the memory zones

• Any program can address any memory location, no restrictions imposed (multitasking is not possible)

1. How is the memory organized in protected mode? How is the physical address calculated?

The address calculation is similar to the previous addressing mode, but with some improvements:

• A segment register stores a segment selector

• The segment selector points to a segment descriptor

• The segment descriptor is a data structure on 32 bit that contains the segment address, segment length, access indicators, etc.

• The offset address is on 32 bit

These design modifications produce several improvements over the real-mode addressing, such as:

• Maximum addressing space is extended to 4 Gb (232)

• A data segment has variable length, from 1 byte to 4 GB

• 3 levels of protection (0 is the highest priority)

• A data segment is accessible only to the assigned task and maybe to the operating system

• Some data segments can be blocked during the write operation (e.g. code segment)