**1.1 Purpose of the base computer**

**The base computer is a simple hypothetical machine that has the typical features of many specific computers. Knowledge of the principles of construction and operation of this computer is a good basis for mastering microprocessor systems of any types and models, therefore it is called the basic computer. Naturally, it is advisable to start studying a computer with a similar machine.**

**low level, which can be done in practice, using its models built for different types of personal computers. When constructing the basic computer, the "Electronics 100" and "Saratov" computers, as well as similar computers of the PDP8 type, the single-chip IM6100 microprocessor and the DECmate 11 personal computer, were chosen as the prototype.**

1.1 Mục đích của máy tính cơ sở

Máy tính cơ sở là một máy giả định đơn giản có các tính năng điển hình của nhiều máy tính cụ thể. Kiến thức về nguyên lý cấu tạo và hoạt động của máy tính này là cơ sở tốt để làm chủ các hệ thống vi xử lý thuộc bất kỳ kiểu và mô hình nào, do đó nó được gọi là máy tính cơ sở. Đương nhiên, nên bắt đầu nghiên cứu một máy tính với một máy tương tự.

mức thấp, có thể được thực hiện trong thực tế bằng cách sử dụng các mô hình được xây dựng cho các loại máy tính cá nhân khác nhau. Khi chế tạo máy tính cơ bản, máy tính điện tử 100 và Saratov, cũng như các máy tính tương tự kiểu PDP8, bộ vi xử lý IM6100 một chip và máy tính cá nhân DECmate 11, được chọn làm nguyên mẫu.

**1.2 Structure of the base computer**

**In fig. B.1 shows a simplified structure of a basic computer. It is a battery-type unicast computer operating with 16-bit words. It implements two types of addressing: direct and indirect.**

1.2 Cấu trúc của máy tính cơ sở

Trong bộ lễ phục. B.1 cho thấy cấu trúc đơn giản của một máy tính cơ bản. Nó là một máy tính unicast loại pin hoạt động với các từ 16-bit. Nó thực hiện hai kiểu xưng hô: trực tiếp và gián tiếp.

**Let's consider the components of the basic computer, while not touching the input-output devices (IO) and the control panel (CP). Memory. Consists of 2048 cells (16-bit) with addresses 0, 1, ..., 2046, 2047. Eight memory cells with addresses from 008 to 00F are slightly different from the rest.**

**These cells are called index cells and they are intended for organizing auto-incremental addressing and are used in cyclic programs (section 1.5). CPU. Consists of a number of registers, an arithmetic logic unit and a control unit.**

Chúng ta hãy xem xét các thành phần của máy tính cơ bản, trong khi không chạm vào các thiết bị đầu vào - đầu ra (IO) và bảng điều khiển (CP). Ký ức. Gồm 2048 ô (16-bit) có địa chỉ 0, 1, ..., 2046, 2047. Tám ô nhớ có địa chỉ từ 008 đến 00F hơi khác so với phần còn lại.

Các ô này được gọi là ô chỉ mục và chúng được dùng để tổ chức địa chỉ tự động tăng dần và được sử dụng trong các chương trình tuần hoàn (phần 1.5). CPU. Bao gồm một số thanh ghi, một đơn vị logic số học và một đơn vị điều khiển.

**An arithmetic logic unit (ALU) can perform arithmetic operations such as addition and addition taking into account the carry resulting from the previous operation, the operation of logical multiplication and inversion. All transfers between registers in the computer are also performed through the ALU.**

Một đơn vị logic số học (ALU) có thể thực hiện các phép toán số học như phép cộng và phép cộng có tính đến giá trị mang theo từ phép toán trước, phép toán phép nhân và phép nghịch đảo logic. Tất cả các chuyển đổi giữa các thanh ghi trong máy tính cũng được thực hiện thông qua ALU.

**Command counter (SC) - a register that stores the address of a memory cell containing the next executable program command. Since commands can be located in any of 2048 = 2^11 memory cells, the CK has 11 bits.**

Bộ đếm lệnh (SC) - thanh ghi lưu địa chỉ của ô nhớ chứa lệnh chương trình thực thi tiếp theo. Vì các lệnh có thể được đặt trong bất kỳ ô nào trong số 2048 = 2^11 ô nhớ, nên CK có 11 bit.

**The address register (RA) is an 11-bit register that serves to organize calls to memory cells and contains the address of the memory cell to which the processor accesses.**

Thanh ghi địa chỉ (RA) là một thanh ghi 11 bit được sử dụng để tổ chức các cuộc gọi đến các ô nhớ và chứa địa chỉ của ô nhớ mà bộ xử lý truy cập.

**Command register (RK) is a 16-bit register used to store the code of the command currently being executed.**

Thanh ghi lệnh (RK) - Thanh ghi 16-bit dùng để lưu mã của lệnh đang được thực thi.

**Data register (RD) - 16-bit register for temporary storage of 16-bit words when exchanging information between the memory and the processor.**

Thanh ghi dữ liệu (RD) - thanh ghi 16 bit để lưu trữ tạm thời các từ 16 bit khi trao đổi thông tin giữa bộ nhớ và bộ xử lý.

**Buffer register (BR) - 17-bit register for temporary storage of calculation results in ALU. Also this register is used when performing shifts.**

Thanh ghi đệm (BR) - Thanh ghi 17 bit để lưu trữ tạm thời các kết quả tính toán trong ALU. Ngoài ra thanh ghi này được sử dụng khi thực hiện ca.

**Accomulator (A) - 16-bit register, which is one of the main elements of battery-type processors. The machine can simultaneously perform arithmetic and logical operations with only one or two operands. One of the operands is in the accumulator, and the second (if there are two of them) is in the data register. The result is ultimately placed in A.**

**When performing operations on the accumulator, the computer processor saves some signs of the result in special one-bit registers:**

Pin (A) - Thanh ghi 16-bit, là một trong những yếu tố chính của bộ xử lý loại pin. Máy có thể thực hiện đồng thời các phép toán số học và logic chỉ với một hoặc hai toán hạng. Một trong các toán hạng nằm trong bộ tích lũy và toán hạng thứ hai (nếu có hai trong số chúng) nằm trong thanh ghi dữ liệu. Kết quả cuối cùng được đặt trong A.

Khi thực hiện các hoạt động trên bộ tích lũy, bộ xử lý máy tính lưu một số dấu hiệu của kết quả trong các thanh ghi một bit đặc biệt:

**The carry flag (C) acts as a continuation of the accumulator and is filled with overflow A. When performing arithmetic and shift operations, the 16th bit of the buffer register gets into it.**

**The zero (Z) flag stores information about whether the content is equal battery zero, is filled when performing operations on the battery.**

**The sign flag (N) preserves the sign of the number in the accumulator, effectively duplicating its 15th bit.**

Cờ mang (C) hoạt động như một phần tiếp theo của bộ tích lũy và được lấp đầy bởi tràn A. Khi thực hiện các phép toán số học và dịch chuyển, bit thứ 16 của thanh ghi đệm sẽ đi vào nó.

Cờ zero (Z) lưu trữ thông tin về việc nội dung có bằng nhau hay không

pin không, được làm đầy khi thực hiện các hoạt động trên pin.

Cờ dấu (N) giữ nguyên dấu hiệu của số trong bộ tích lũy, sao chép hiệu quả bit thứ 15 của nó.

**1.3. Base computer command system**

**Классификация команд. БЭВМ способна исполнять точно определенный набор команд. При составлении программы пользователь ограничен этими командами. В зависимости от того, к каким блокам базовой ЭВМ обращается команда или на какие блоки она ссылается, команды можно разделить на три**

**группы:**

**• адресные команды;**

**• безадресные команды;**

**• команды ввода-вывода.**

**Classification of teams. A computer is capable of executing a precisely defined set of commands. When compiling a program, the user is limited to these commands. Depending on which blocks of the base computer the command refers to or which blocks it refers to, the commands can be divided into three groups:**

**• address commands;**

**• unaddressed commands;**

**• input-output commands.**

**Адресные команды предписывают машине производить действия с ячейкой памяти, адрес которой указан в адресной части команды.**

**Address commands instruct the machine to perform actions with the memory cell, the address of which is indicated in the address part of the command.**

Các lệnh địa chỉ hướng dẫn máy thực hiện các hành động với ô nhớ, địa chỉ của ô nhớ được chỉ ra trong phần địa chỉ của lệnh.

**Безадресные команды выполняют различные действия без ссылок на ячейку памяти. Например, команда CLA (табл. В.1) предписывает ЭВМ очистить аккумулятор (записать в А код нуля). Это команда обработки операнда, расположенного в конкретном месте, "известном" машине. Другой пример безадресной команды -команда HLT.**

**Unaddressed commands perform various actions without reference to a memory cell. For example, the CLA command (Table B.1) instructs the computer to clear the battery (write the zero code in A). This is a command to process an operand located at a specific location "known" to the machine. Another example of an addressless command is the HLT command.**

Các lệnh chưa được giải quyết thực hiện các hành động khác nhau mà không cần tham chiếu đến ô nhớ. Ví dụ, lệnh CLA (Bảng B.1) hướng dẫn máy tính giải phóng pin (viết mã số 0 trong A). Đây là lệnh xử lý một toán hạng nằm tại một vị trí cụ thể mà máy "biết". Một ví dụ khác về lệnh không có địa chỉ là lệnh HLT.

**Команды ввода-вывода управляют обменом данными между процессором и внешними устройствами ЭВМ.**

**Полный перечень команд базовой ЭВМ приведен в таблице В.1.**

**Input-output commands control data exchange between the processor and external computer devices.**

**A complete list of host computer commands is given in Table B.1.**

Các lệnh I / O kiểm soát việc trao đổi dữ liệu giữa bộ xử lý và các thiết bị máy tính bên ngoài.

Danh sách đầy đủ các lệnh máy tính chủ được đưa ra trong Bảng B.1.

**Форматы команд и способы адресации. Разработчики БЭВМ выбрали три формата 16-битовых (однословных) команд с 4-битовым кодом операции (рис. В.2).**

**Command formats and addressing methods. The computer designers have chosen three formats of 16-bit (one-word) instructions with a 4-bit operation code (Fig. B.2).**

Các định dạng lệnh và phương pháp đánh địa chỉ. Các nhà phát triển máy tính đã chọn ba định dạng của lệnh 16 bit (một từ) với mã hoạt động 4 bit (Hình B.2).

1. How many times there is a memory access in unaddressed commands:

Basically, this happens 1 time for almost everyone - on the instruction fetch cycle, but there are also commands related to the Stack, when you need to access memory again - they start from 0800, in which case there will be 2 memory accesses

1. What operations can ALU perform?

Forming zero, +1, inversion, logical multiplication

1. Все про флаги NZVC

N - negative Z - zero V - overflow C - carry If we have C after some operation, then it will be until we have another arithmetic operation, which changes the carry bit to 0 (It's not accurate) V put like this - take the transfer from the 14th digit to 15 (C\_14) and the transfer from the 15th (the usual C), and then they are xor'ed and the result is added to V

1. Как отличить команду с прямой адресацией от команды с относительной? How to distinguish a command with direct addressing from a command with a relative one? Làm thế nào để phân biệt một lệnh có địa chỉ trực tiếp với lệnh có địa chỉ tương đối?

At the eleventh bit, in the address command it is equal to one

1. What is a buffer for?

The buffer is needed for intermediate values, for example, to add one to IP, on the first cycle IP is fed to the right input of the ALU, and then to the buffer and AR at the same time, on the next cycle, one is added to BR and written to IP. So for logical "or" buffer is also used (~ A & ~ B) -> BR BR-> ALU (~ BR) -> AC "

1. How to directly load the operand into the command? Làm thế nào để nạp trực tiếp toán hạng vào lệnh?

For commands with direct loading of the operand, the third tetrad is F

1. Several ways to change the sign of a number

You can load zero into the accamulater and subtract the number, or use the NEG command, which changes the sign of the accamulater

1. How are V and C bits set?

C - is there a carry from the sign bit, V - if there is simultaneously or simultaneously no carry from the sign and most significant bits

Overflow can only happen when adding two numbers of the same sign and

getting a different sign. So, to detect overflow we don't care about

any bits except the sign bits. Ignore the other bits.

When adding two binary values, consider the binary carry coming into

the leftmost place (into the sign bit) and the binary carry going out

of that leftmost place. (Carry going out of the leftmost [sign] bit

becomes the CARRY flag in the ALU.)

**The rules for turning on the carry flag in binary/integer math are two:**

The carry flag is set if the addition of two numbers causes a carry

out of the most significant (leftmost) bits added.

The carry (borrow) flag is also set if the subtraction of two numbers

requires a borrow into the most significant (leftmost) bits subtracted.

In unsigned arithmetic, watch the carry flag to detect errors.

In signed arithmetic, the carry flag tells you nothing interesting.

**The rules for turning on the overflow flag in binary/integer math are two:**

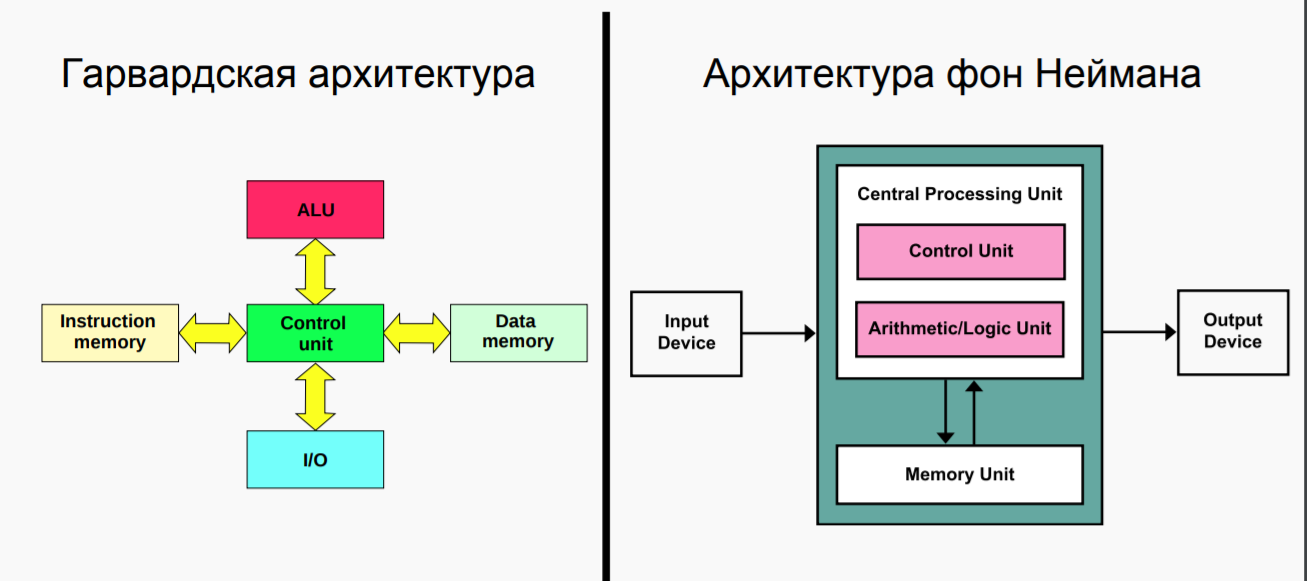
If the sum of two numbers with the sign bits off yields a result number

with the sign bit on, the "overflow" flag is turned on.

If the sum of two numbers with the sign bits on yields a result number

with the sign bit off, the "overflow" flag is turned on.

1. Как БЭВМ ставит V?(формула и что она значит)

****