**Sổ ghi Thí nghiệm 2 – OPD**

Mục tiêu: Nguyên cứu kĩ thuật làm việc trên máy tính cơ bản và trình tự thực hiện các lệnh

Bài tập:

Chuẩn bị:

* Thiết bị và hệ thống các lệnh của máy tính cơ bản (Phụ lục B, đoạn 1.1 -1.6)
* Trình tự thực hiện lệnh của máy (1.9)
* Hướng dẫn làm việc với mô-đen của máy (Phụ lục G)
* Cách biểu diễn các giá trị số và logic trong máy

Trình tự làm việc:

Nội dung báo cáo công việc:

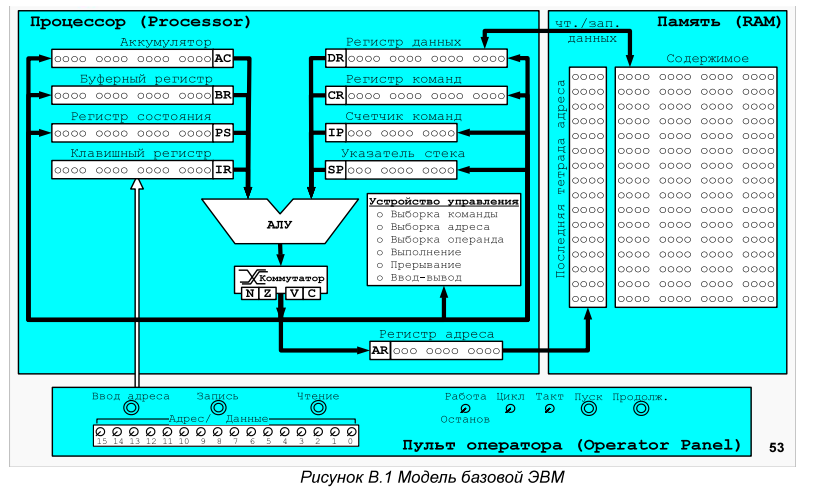
**Phần chuẩn bị:**

**Phụ lục B:**

* 1. **Mục đích của máy tính cơ sở:**

Базовая электронная вычисленная машина (БЭВМ)- это простая машина, облада.щая типичными свойствами многиз конкретных ЭВМ.

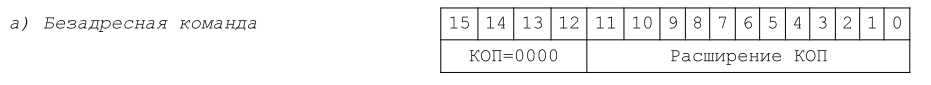
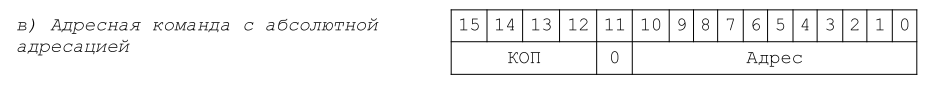
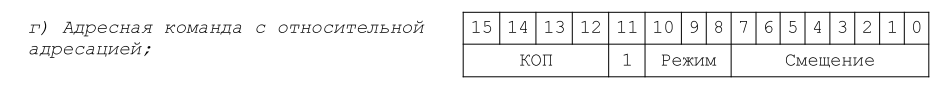
*Máy tính điện tử cơ bản (BEVM) là một loại máy đơn giản có các đặc tính điển hình của nhiều máy tính cụ thể.*

* 1. **Cấu trúc của Mtcb:**
* Bộ nhớ (**RAM**): Gồm 2048 ô- 16 bit với các địa chỉ 0, 1, 2, ..., 2047. Các ô từ 0 đến 0xF chứa véc tơ ngắt (вектор прерывания)
* Bộ xử lí **CPU** (процессор): Gồm một số thanh ghi (регистры), thiết bị logic- số học (АЛУ), công tắc (коммутатор), khối lập dấu cho kết quả, thiết bị điều khiển.
  + Thiết bị điều khiển (**УУ**- устройство управления)(**CU**- control unit)(**МПУ**- микропрограммное устройство) thực hiện các lệnh máy của CPU
  + Thiết bị logic- số học (**АЛУ**- арифметико- логическое устройство ) thực hiện các phép tính số học: cộng, cộng với kết quả trước, nhân logic, nghịch đảo. Đầu ra của ALU được nối với công tắc.
  + Công tắc (**коммутатор**): là thiết bị nhận kết quả 18 bit từ ALU (16 bit kết quả và các bit dùng để định dạng dấu). Thông tin được gửi từ công tắc đến bus dữ liệu để ghi vào các thanh ghi và khối thiết lập dấu kết quả.
  + Khối thiết lập dấu kết quả (блок установки признаков результата) bao gồm:
    - Cờ vận chuyển (C- Carry) (флаг переноса) được lấp đầy khi kết quả vượt quá ranh giới 16 bit
    - Cờ tràn (V- oVerflow) (флаг переполнения) báo hiệu sự tràn lưới bit trong các hoạt động với số có dấu
    - Cờ không (Z- Zero) (флаг нуля) bảo quản thông tin về việc, nội dung AC có bằng 0 hay không
    - Cờ dấu (N- Negative) (флаг знака) bảo quản dấu của số trong AC
  + Các thanh ghi:
    - AC- **Accumulator**: thanh ghi 16 bit, lưu trữ kết quả của các phép toán logic- số học
    - BR- Buffer Register: thanh ghi 16 bit lưu trữ dữ liệu trung gian trong quá trình hoạt động
    - PS- Program State: thanh ghi 16 bit, lưu trữ các bit điều khiển hoạt động của máy tính và các đặc tính (nzvc) của kết quả
    - IR- Input Register: thanh ghi 16 bit, nằm ở 1 phần của bảng điều khiển máy tính và dùng để nhập địa chỉ của chương trình
    - DR- Data Register: thanh ghi 16 bit lưu trữ tạm thời các từ 16 bit trong quá trình RAM và CPU (bộ xử lí) trao đổi
    - CR- Command Register: thanh ghi 16 bit lưu trữ mã lệnh hiện đang thực thi
    - IP- Instruction Pointer: thanh ghi 11 bit chứa địa chỉ ô RAM, trong đó chứa địa chỉ lệnh chương trình thực thi tiếp theo. Vì các lệnh có thể đặt bất kì trong số 2048=2^11 ô nên IP có 11 bit.
    - SP –Stack Pointer: con trỏ ngăn xếp 11 bit, trỏ đến đầu ngăn xếp của 1 RAM đặc biệt; lưu trữ địa chỉ trả về và tham số của chương trình con và ngắt
    - AR- Address Register: thanh ghi 11 bit tổ chức quyền truy cập vào các ô nhớ và chứa địa chỉ ô nhớ được truy cập (bởi CPU)
  1. **Hệ thống lệnh của Mtcb:**

Phân loại lệnh: Chia thành 4 nhóm:

* Lệnh không có địa chỉ ( безадресная команда)
* Lệnh vào/ ra (команда ввода- вывода)
* Lệnh địa chỉ (адресная команда)
* Lệnh nhánh (команда ветвления)

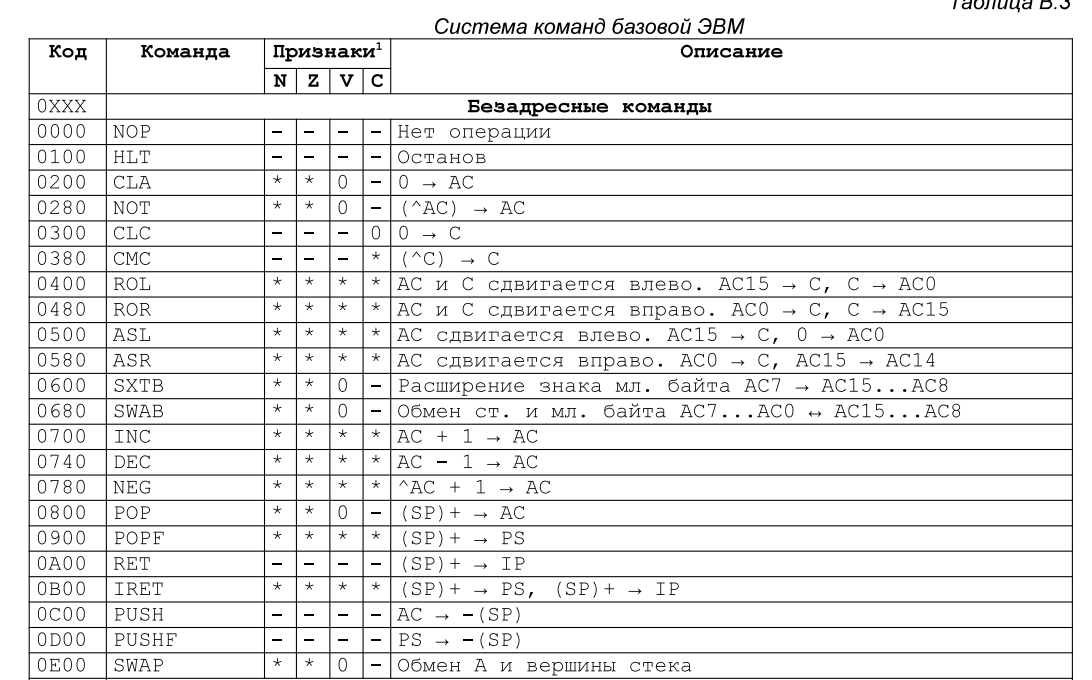
Việc lựa chọn kiểu lệnh được thiết bị điều khiển (МПУ) phân tích qua 4 bit cao nhất của mã lệnh (bit 12 -15 ) được gọi là mã hoạt động ( КОП- код операции).

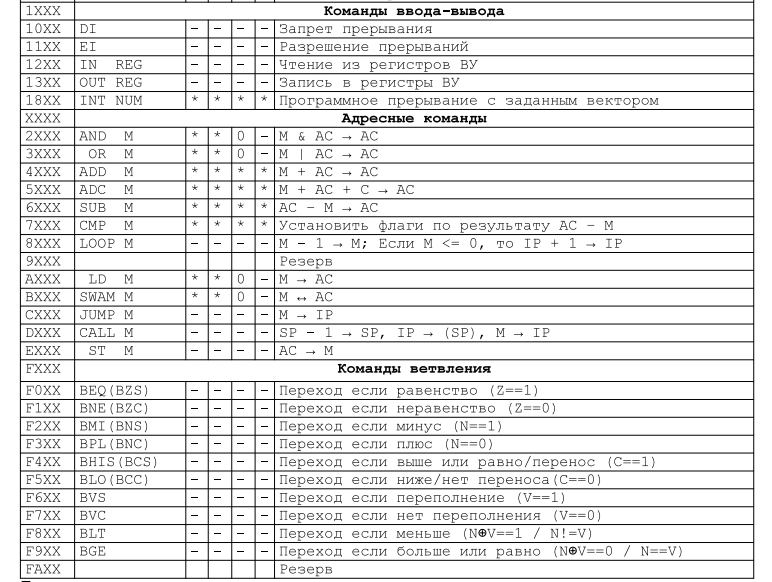
1. Lệnh không địa chỉ: thực hiện hành động không cần tham chiếu tới vị trí RAM
2. Các lệnh I/O kiểm soát việc trao đổi giữa CPU và các thiết bị máy tính bên ngoài
3. Lệnh địa chỉ: hướng dẫn máy thực hiện các hành động với 1 ô nhớ; Địa chỉ của ô nhớ được xác định trên phần địa chỉ của lệnh (bao gồm 12 bit từ bit 0 đến bit 11 ). Mã hoạt động (КОП) (12- 13- 14- 15) lấy các giá trị từ 0x2 đến 0xE và chỉ định thao tác
   * Với địa chỉ trực tiếp tuyệt đối *(С прямой абсолютной адресацией*): bit 11 luôn là “0”; Phần địa chỉ (từ bit 0 đến bit 10) ghi giá trị tuyệt đối của địa chỉ toán hạng (tức là số ô trong không gian địa chỉ trong RAM). Khi thực hiện 1 thao tác, lệnh sẽ xử lí trực tiếp địa chỉ đã cho = cách chọn/ viết 1 toán hạng
   * Với địa chỉ tương đối (*С относительной адресацией*): bit 11 là “1”; Bit 8 -> 10 là chế độ địa chỉ; Các bit 0 -> 7 chứa phần bù để tính địa chỉ của toán hạng trong RAM: Thêm phần bù với giá trị IP. Phần bù có thể >0 hoặc <0, cho phép trỏ đến 127 ô trước hoặc 128 ô sau lệnh hiện tại trong RAM. Lưu ý phần bù =0 sẽ trỏ đến ô liền kề lệnh (bởi vì thời điểm tính toán toán hạng, RAM đã tăng lên 1 do lệnh được thực thi).

Có các chế độ của địa chỉ tương đối:

* + 1. Tương đối trực tiếp (mã 0xE)
    2. Tương đối gián tiếp (mã 0x8)
    3. Gián tiếp tự động tăng (0xA)
    4. Gián tiếp tự động giảm (0xB)
    5. Phần bù tương ứng với SP (0xC)
    6. Tải trực tiếp toán hạng (0xF)

1. Các lệnh phân nhánh: cho phép bạn tiếp tục quá trình tính toán từ 1 địa chỉ chương trình khác, tùy thuộc vào trạng thái của các thuộc tính kết quả N-Z-V-C

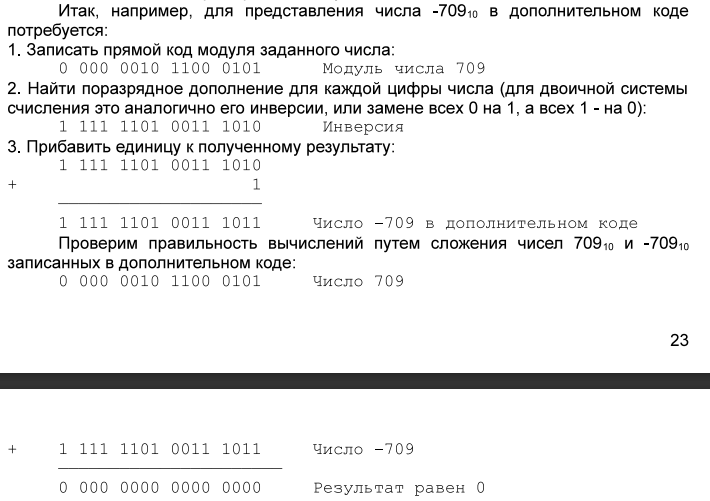


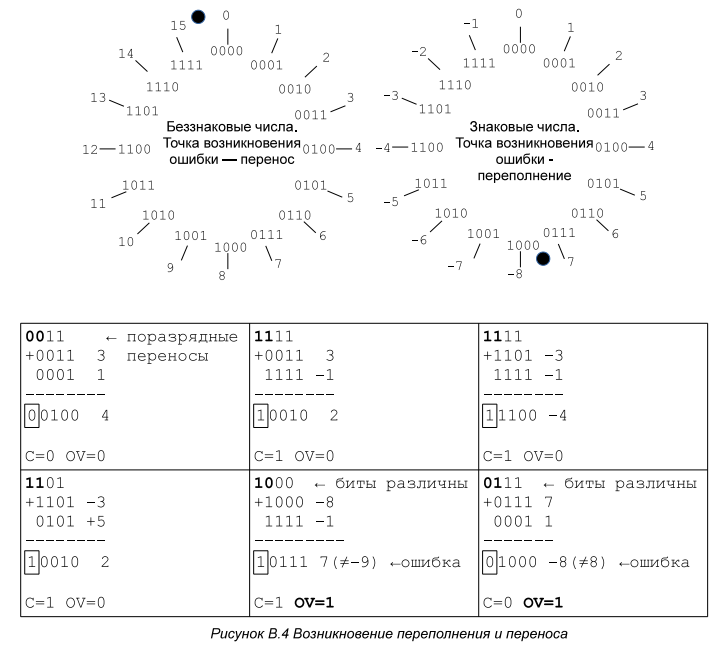


**Признаки1**- cho biết: “-”- lệnh không ảnh hưởng tới thuộc tính; “0”- thuọc tính sẽ được đặt lại; “\*”- giá trị thuộc tính phụ thuộc vào kết quả của thao tác

* 1. **Biểu diễn thông tin (số) trong Mtcb:**

Khi biểu diễn số có dấu , mã bù (дополнительный код) được sử dụng, trong đó bit cao nhất xác định dấu của số: 0 cho số dương và 1 cho số âm.



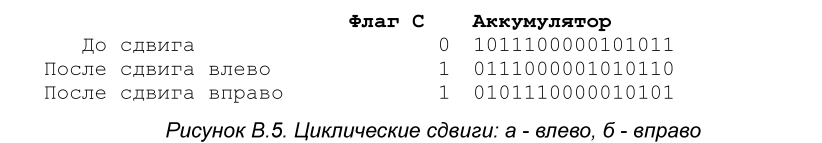


* 1. **Các phép toán RAM và số học:**

Mtcb phụ thuộc vào bộ xử lí kiểu tích lũy, trong đó 1 toán hạng nằm trong ô nhớ và toán hạng thứ 2 nằm trong thanh ghi AC.

* Tải 1 giá trị để xử lí (Загрузка значения для обработки): LD
* Lưu kết quả (Сохранение результата): **ST**
* Trao đổi nội dung giữa các ô RAM và AC (Обмен содержимого ячейки памяти и аккумулятора): SWAM
* Trao đổi giữa bit cao và bit thấp (Обмен старшего и младшего байтов): SWAB
* Cộng các số nguyên nhị phân có dấu và không có dấu (Сложение целых двоичных чисел со знаком и без знака): ADD
* ADC?
* Tăng lên 1 (Увеличение на 1): INC
* Giảm xuống 1 (Уменьшение на 1): DEC
* Đổi dấu của số (Изменение знака): NEG=NOT+INC
* Mở rộng dấu (Расширение знака): SXTB
* Trừ (Вычитания (X-Y)): LD X, SUB Y
* So sách các số (Сравнение чисел): CMP
* Nhân và chia (Умножение и деление): ALU không hỗ trợ các hoạt động này, nên phải sử dụng các lệnh ASR và ASL (phép toán nhân và chia cho 2)
  1. **Sự dịch chuyển và các phép toán logic (Сдвиги и логические операции)**

Việc xử lí dữ liệu dữ liệu dựa trên bit được thực hiện trên Mtcb được thực hiện bởi các lệnh sau:

* Nhân logic (логическое умножение): AND- thực hiện trên từng bit của nội dung AC và nội dung ô nhớ được lệnh địa chỉ dẫn đến
* Cộng logic (логическое сложение): OR- giống Nhân logic
* Dịch chuyển tuần hoàn 1 bit (циклический сдвиг влево и вправо на 1 разряд): ROL và ROR- đóng AC và thanh ghi vận chuyển vào vòng, sau đó dịch chuyển tất cả các bit của vòng sang phải hoặc trái 1 bit
* Làm sạch AC (очистка аккумулятора): CLA
* Nghịch đảo bit hoặc phủ định nội dung của AC (побитная инверсия или отрицание содержимого аккумулятора): NOT/ CMA
* Làm sạch và nghịch đảo cờ vận chuyển C: (Очистка и инверсия флага переноса С) CLC và CMC

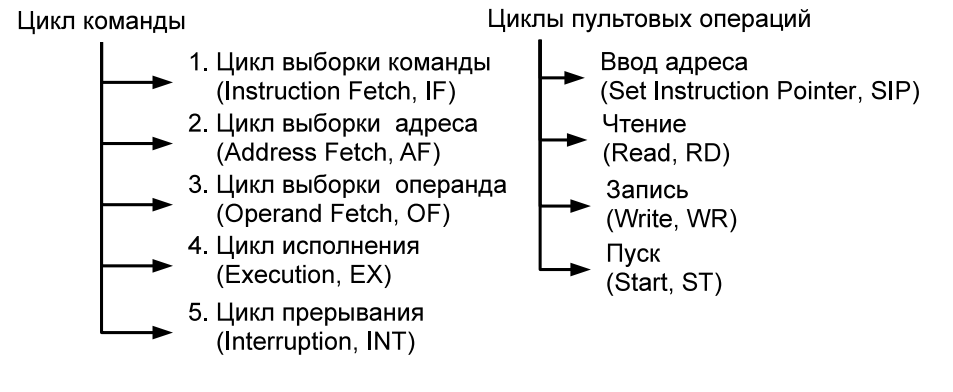
**Thực hiện lệnh máy (1.9)**

УУ thực hiện các hành động:

* chuyển các lệnh , toán hạng và kết quả trung gian giữa các thanh ghi
* phân tích các mã hoạt động và đia chỉ
* điều khiển ALU
* đọc/ghi nội dung RAM hoăc các thanh ghi của thiết bị I/O

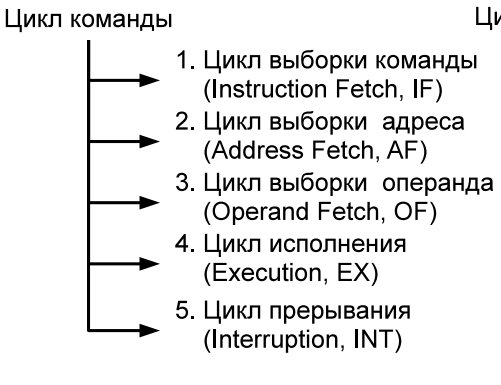
Chuỗi các hành động để thực hiện lệnh và thao tác, gọi là chu trình (цикл). Tập hợp các chu trình tạo nên mã chương trình МПУ, gọi là vi mô hoặc vi chương trình (микрокод/ микропрограмма).

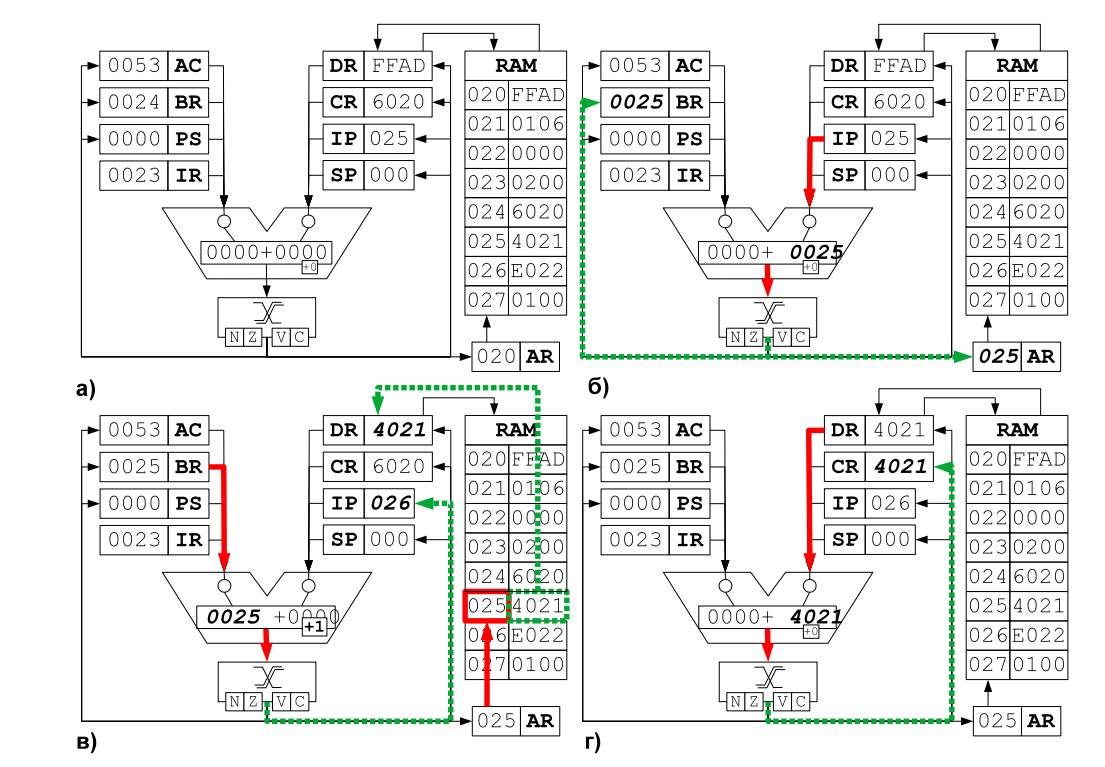
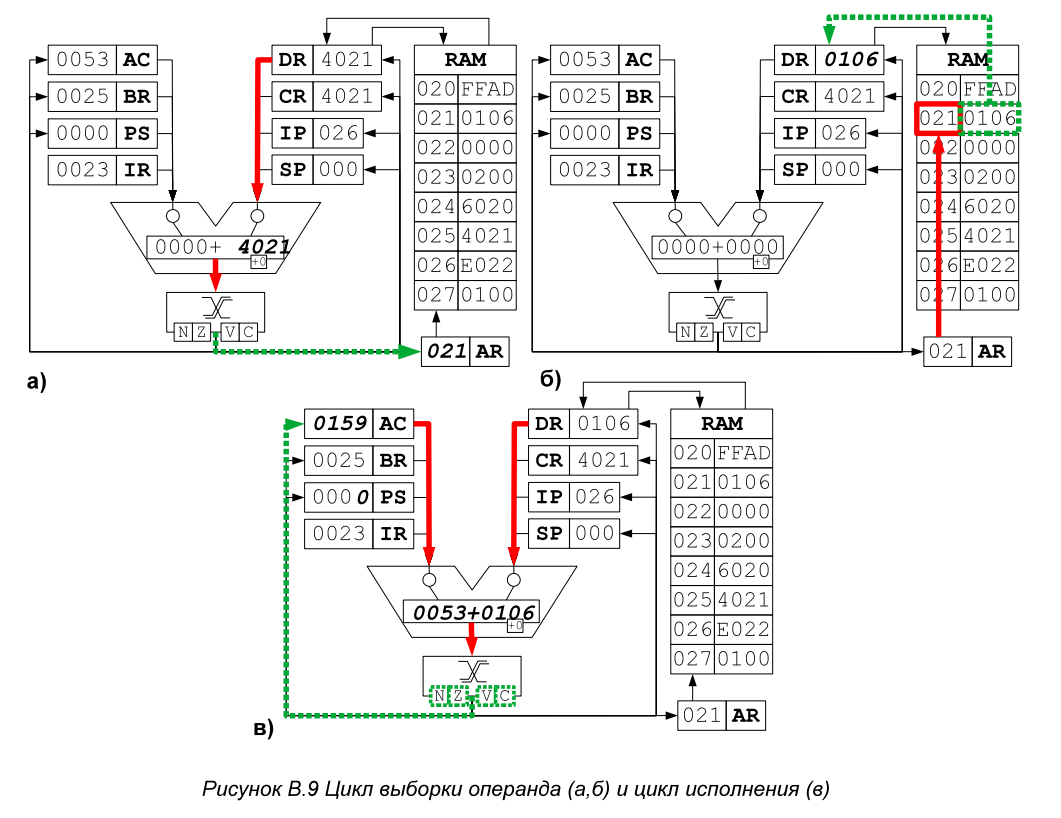
1. **Цикл пультовых операций (***Chu trình thao tác điều khiển***)**

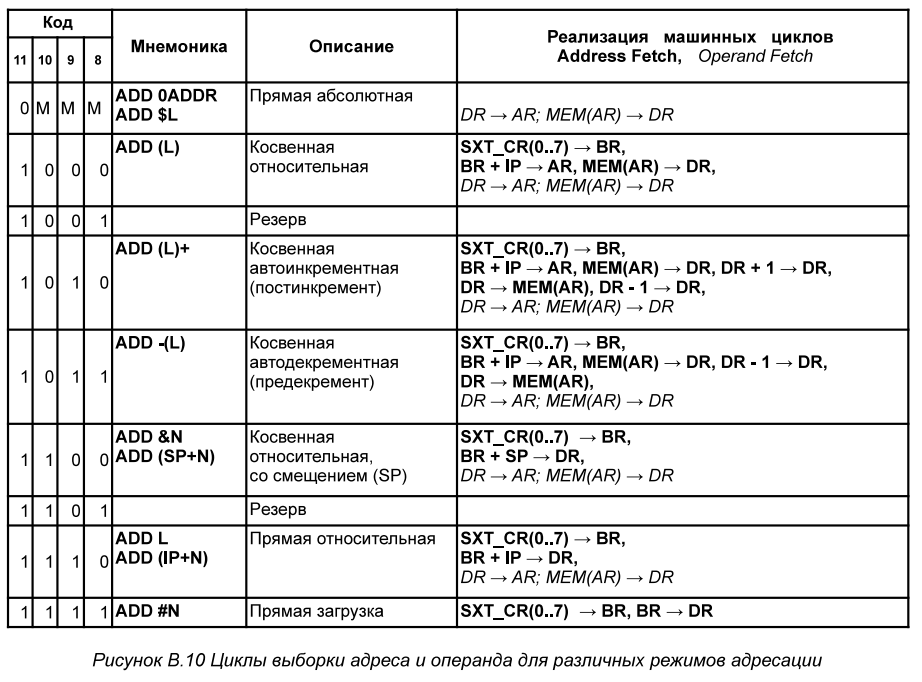
* **Ввод адреса** (SIP)- ghi nội dung của IR vào IP
* **Чтение** (RD)- đọc nội dung của ô nhớ (có địa chỉ thể hiện ở IP) vào DR; Sau đó tăng nội dung của IP lên 1 đơn vị (tức là chuyển sang ô nhớ tiếp theo)
* **Запись** (WR)- ghi nội dung của IR vào ô nhớ có địa chỉ thể hiện ở IP; Sau đó tăng nội dung của IP lên 1 đơn vị
* **Пуск** (ST)- Đặt lại nội dung của các thanh ghi DR, CR, SP, AC, BR, AR, đặt lại các dấu hiệu kết quả, tắt Ngắt; Nếu chế độ “Работа” được đặt, tiến hành thực hiện lệnh có địa chỉ trong IP.

Trên bảng vận hành có các nút điểu khiển РАБОТА/ОСТАНОВ, ТАКТ, Продолжение

1. **Цикл команды (***Chu trình lệnh***)**

* **Цикл выборки команды**: Lựa chọn lệnh- Trong chu trình máy hiện tại thực hiện đọc lệnh từ RAM và giải mã 1 phần của lệnh
* **Цикл выборки адреса** : không tồn tại trong lệnh có địa chỉ tuyệt đối, bởi địa chỉ được tạo đầy đủ sẽ ghi trực tiếp vào mã lệnh; Máy tính chuyển sang chu kì tiếp theo
* **Цикл выборки операнда**
* **Цикл исполнения**
* **Цикл прерывания**

**Цикл выборки команды**



**Làm việc với Mô-đen của MTCB**

Các phím sử dụng để di chuyển trên thanh ghi bàn phím

|  |  |
| --- | --- |
| RIGHT | Di chuyển con trỏ sang phải 1 vị trí |
| LEFT | Di chuyển con trỏ sang trái 1 vị trí |
| UP | Đảo ngược bit dựa trên vị trí con trỏ hiện tại |
| 1 | Nhập số “1” tại vị trí hiện tại và di chuyển nó tới vị trí tiếp theo |
| 0 | Nhập số “0” tại vị trí hiện tại và di chuyển nó tới vị trí tiếp theo |

Các phím sau cũng được sử dụng trong quá trình hoạt động

|  |  |
| --- | --- |
| F4 | Nhập địa chỉ: IR 🡪 IP |
| F5 | Ghi: IR 🡪 RAM (IP); IP+=1 |
| F6 | Đọc: RAM (IP) 🡪 DR; IP+=1 |
| F7 | Bắt đầu |
| F8 | Tiếp tục |
| F9 | Phím |
| F10 | Thoát khỏi MTCB |
| Shift+F4 | Thay đổi mask |

Làm việc với thiết bị bên ngoài

|  |  |
| --- | --- |
| F1, F2, F3 | tương ứng với các tbi bên ngoài là 1, 2, 3 |
| Tab | Chuyển sang chế độ nhập dữ liệu vô DR ВУ2 ВУ3 |

Làm việc với các lệnh vi mô

|  |  |
| --- | --- |
| Tab | Chuyển đổi đầu vào giữa bộ nhớ thông thường và bộ nhớ lệnh |
| Shift + F9 | Bật/ tắt chế độ TACT |