

# Языки описания схем

(mk.cs.msu.ru → Лекционные курсы → Языки описания схем)

Блок 16

Процессор:

Операционный автомат

Комбинационный управляющий автомат

лектор:

Подымов Владислав Васильевич

e-mail:

[valdus@yandex.ru](mailto:valdus@yandex.ru)

Осень 2018

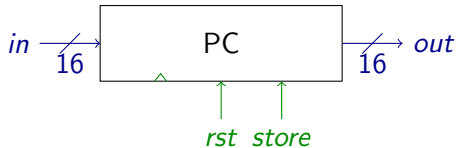
# Вступление

Чтобы спроектировать операционный автомат процессора, достаточно

- ▶ разделить порты его базовых функциональных блоков на **информационные** и **управляющие**
- ▶ описать способ соединения информационных портов (*операционный автомат*) отдельно для каждой инструкции
- ▶ совместить описанные операционные автоматы, поставив мультиплексоры в точки конфликта

## (П) Базовые блоки: данные и управление

Счётчик команд:

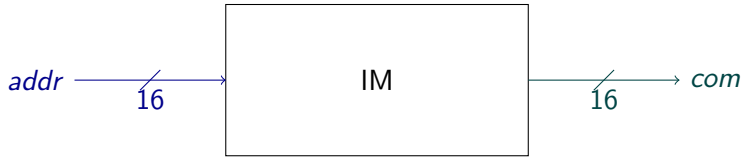


**Данные** — это адрес команды

**Управление** — это сигналы сброса и перезаписи адреса

## (П) Базовые блоки: данные и управление

Память инструкций:

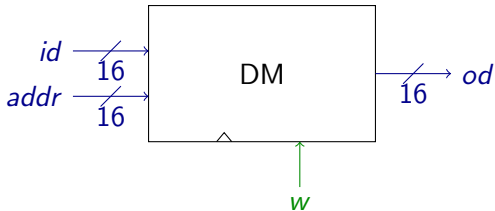


Так как адрес команды отнесён к данным, вход этого блока относится к **данным**

Выход содержит как **данные**, так и **управление**

## (П) Базовые блоки: данные и управление

Память данных:

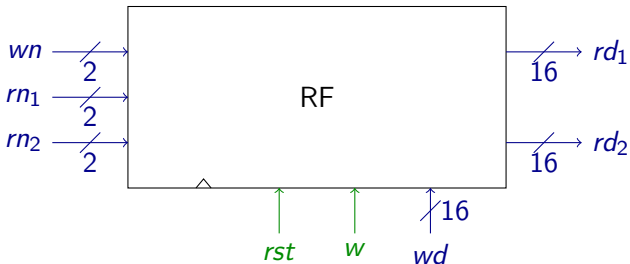


**Управление** — это сигнал перезаписи

**Данные** — это всё остальное

## (П) Базовые блоки: данные и управление

### Блок регистров:



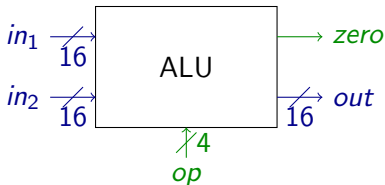
Номера регистров (*по аналогии с адресами ячеек в памяти*) — это **данные**

Значения регистров — тоже **данные**

Сигнал перезаписи значения регистра и сигнал сброса — это **управление**

## (П) Базовые блоки: данные и управление

Арифметико-логическое устройство:

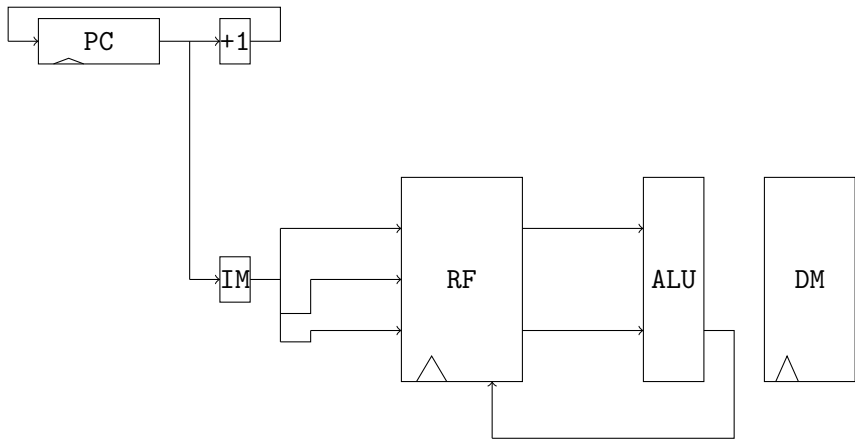


Входные и выходное арифметико-логические значения — это **данные**

Сигнал zero требуется для **управления** выбором следующего значения счётчика команд

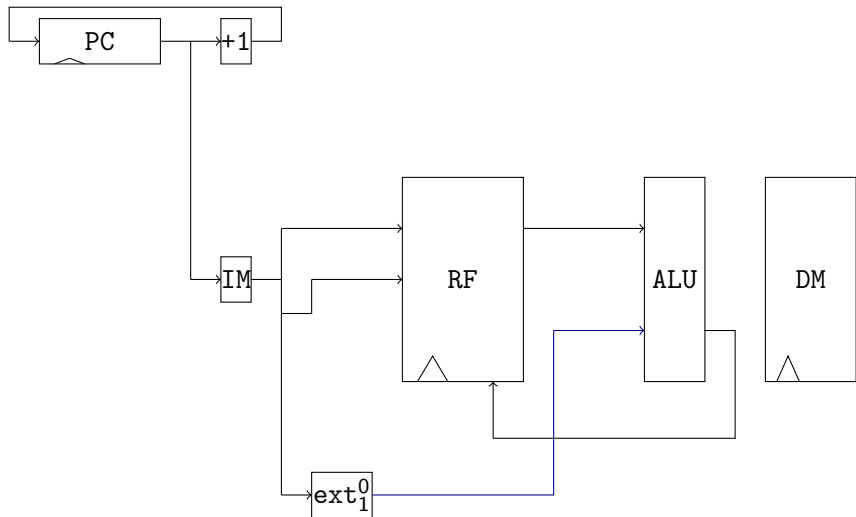
АЛУ используется для многих целей, не обязательно “явных” арифметико-логических преобразований, поэтому вход op нетривиально **управляет** режимом работы АЛУ

## (П) Операционный автомат: AR

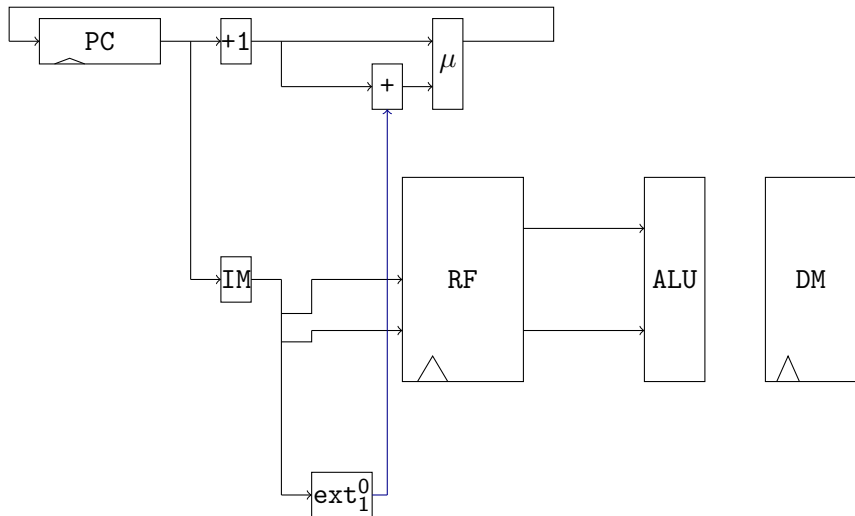




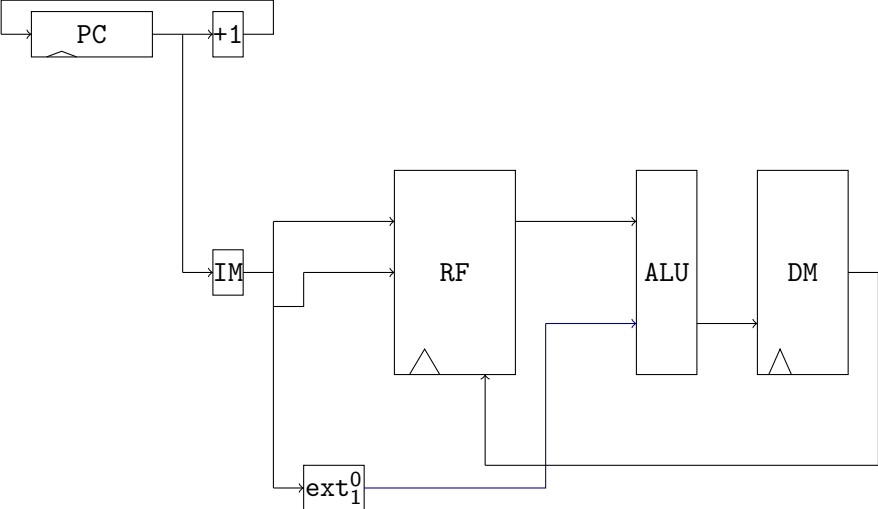
## (П) Операционный автомат: AI



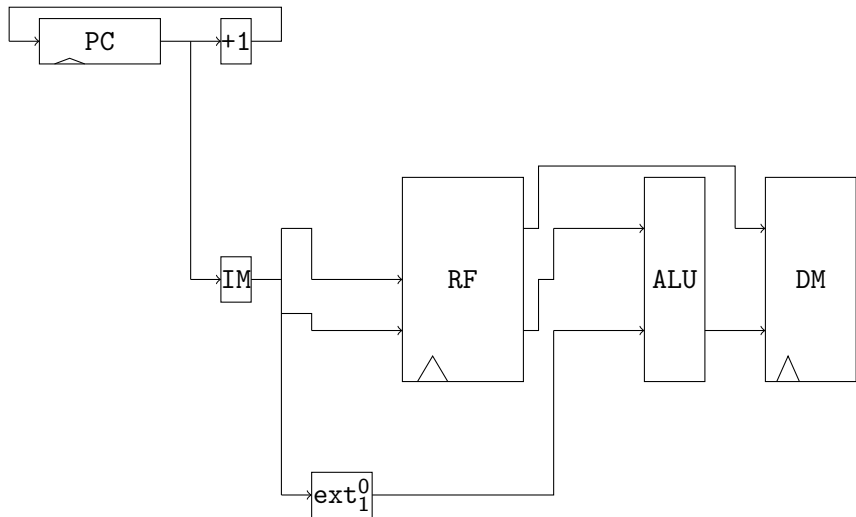
## (П) Операционный автомат: В1



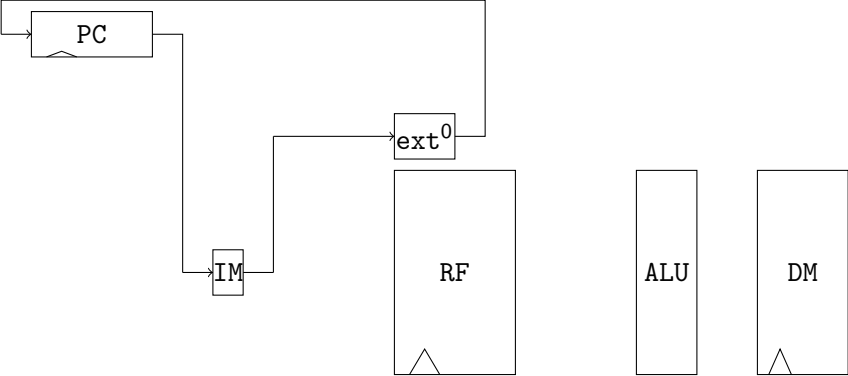
# (П) Операционный автомат: lw



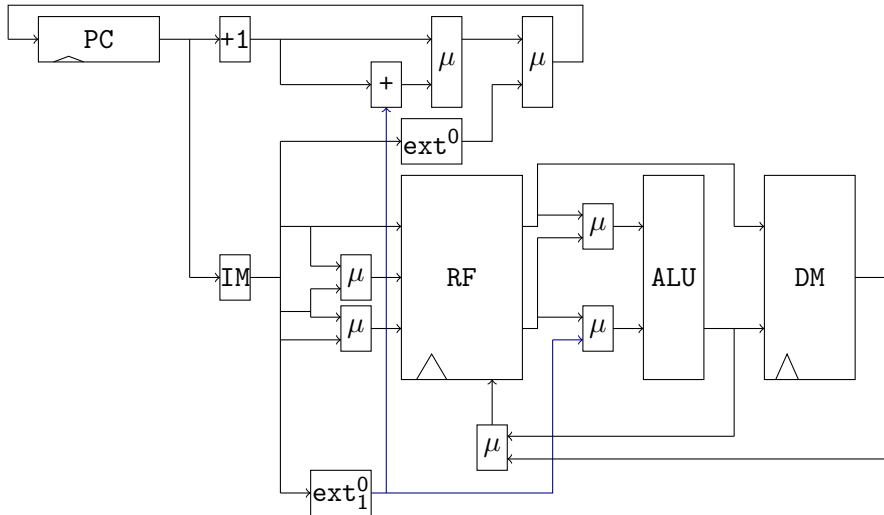
## (П) Операционный автомат: SW



# (П) Операционный автомат: J



## (П) Операционный автомат: всё вместе



## (П) Коротко об управляющем автомате

Сигнал сброса, подаваемый в процессор извне, напрямую передаётся в счётчик команд и блок регистров

Управляющий автомат процессора

- ▶ принимает на вход
  - ▶ код операции команды на выходе IM
  - ▶ код арифметико-логической операции команды на выходе IM (или *“мусор”*, если тип команды отличен от AR)
  - ▶ сигнал zero с выхода АЛУ
- ▶ на основании значений входных сигналов решает, какие сигналы должны быть поданы
  - ▶ в счётчик команд для загрузки
  - ▶ в память данных и блок регистров для перезаписи значения
  - ▶ в АЛУ для выбора требуемой операции
  - ▶ на управляющие входы всех мультиплексоров для выбора значения, пересылаемого в точках конфликта согласно текущей операции

## (П) Коротко об управляющем автомате

Самый простой управляющий автомат требуемого вида имеет комбинационный функционал:

- ▶ перезапись счётчика команд требуется всегда (*то есть по каждому переднему фронту тактового сигнала*)
- ▶ по коду операции, коду арифметико-логической операции и сигналу zero можно **однозначно** сказать,
  - ▶ требуется ли перезапись значения регистра
  - ▶ требуется ли перезапись ячейки памяти
  - ▶ какая операция должна быть выполнена в АЛУ
  - ▶ согласно какому из “простых” операционных автоматов должны выбираться значения в точках конфликта

Процессор с таким простым автоматом называется

**однотактовым**: каждая команда выполняется ровно за один такт, и этот такт целиком посвящён выполнению этой команды



## (П) Коротко об управляющем автомате

Каждый функциональный блок распространяет сигнал с некоторой задержкой

Основные функциональные блоки выстроены последовательно: значения входных сигналов на следующем блоке становятся актуальными относительно текущей команды только после распространения их предыдущими блоками

## (П) Коротко об управляющем автомате

Из-за такой последовательной обработки данных одноктактовый процессор оказывается **крайне** неэффективным:

- ▶ функциональные блоки “простаивают” в ожидании актуализации сигнала, хотя могли бы выполнять полезную работу
- ▶ для инструкций, задействующих много функциональных блоков, требуется большой тактовый период (*маленькая тактовая частота*)
- ▶ так как тактовая частота едина для всех инструкций, на той же (*маленькой*) тактовой частоте будут работать и инструкции, задействующие мало функциональных блоков

Основное назначение одноктактового процессора — служить отправной точкой при изучении устройства процессоров в целом