

# Тактовый сигнал, тактовая частота и разрядность

Луцив Дмитрий Вадимович

Кафедра системного программирования СПбГУ



## 1 Характеристики ЭВМ в целом

## 2 Тактовый сигнал

- Тактовые частоты
- Распространение тактового сигнала
- Виды синхронизации по тактовому сигналу
- Бестактовые процессоры

## 3 Разрядность

- Разрядность процессора
- Разрядность ОЗУ

## Характеристики ЭВМ в целом

- Характеристики процессора
- Объём оперативной памяти
- Объём и скорость устройств хранения данных
- Состав и характеристики интерфейсных устройств

- Качественная: система команд и архитектура в целом — об этом позже
- Количественная: тактовая частота
- Количественная и качественная: разрядность

## Тактовый сигнал

- Тактовые частоты
- Распространение тактового сигнала
- Виды синхронизации по тактовому сигналу
- Бестактовые процессоры

- **Тактовый сигнал** — периодический электрический сигнал, служащий для синхронизации электронных схем
- **Такт** — промежуток времени между тактовыми сигналами
- **Тактовая частота** — частота тактовых сигналов ( $1/\text{длинну такта}$ )
- **Тактовый генератор** — электронная схема, генерирующая тактовый сигнал

# Зачем нужен тактовый сигнал? (1)

- Для многих электронных компонент (например, сумматоров) задано максимальное время на операцию
- Время задаётся в тактах, например
  - Для Zilog Z80:
    - `push RR` (положить значение 16-битного регистра на стек) — 11 тактов
    - `pop RR` (снять со стека и сохранить в 16-битном регистре) — 10 тактов
    - `add a, R` (прибавить 8-битное значение в регистре R к a) — 4 такта
  - Для Intel 80386
    - `add eax, DWORD PTR [ebp-0x8]` (прибавить к 32-битному `eax` значение из памяти по адресу `ebp-8`) — 7 тактов
- Можно быть уверенным в том, когда операция завершена, а не проверять прогресс



# Зачем нужен тактовый сигнал? (1)

- Для многих электронных компонент (например, сумматоров) задано максимальное время на операцию
- Время задаётся в тактах, например
  - Для Zilog Z80:
    - `push RR` (положить значение 16-битного регистра на стек) — 11 тактов
    - `pop RR` (снять со стека и сохранить в 16-битном регистре) — 10 тактов
    - `add a, R` (прибавить 8-битное значение в регистре R к a) — 4 такта
  - Для Intel 80386
    - `add eax, DWORD PTR [ebp-0x8]` (прибавить к 32-битному `eax` значение из памяти по адресу `ebp-8`) — 7 тактов
- Можно быть уверенным в том, когда операция завершена, а не проверять прогресс
- На самом деле даже для Intel i80386 это не всегда так, а для более новых и подавно

## Зачем нужен тактовый сигнал? (2)

- Ясно, в какой момент можно начинать выполнять следующую команду
- Ясно, как синхронизировать разные стадии выполнения одной и той же команды
- Ясно, как синхронизировать различные узлы ЭВМ

При этом

- В ЭВМ обычно много тактовых генераторов и тактовых частот — несколько для ЦП, для ОЗУ (пониже), для шин — чем дальше от ядра процессора, тем ниже
- Тактовый сигнал не привязан к реальному времени: частота высокая, но может «плавать» или понижаться для экономии энергии

# Примеры значений тактовой частоты

**1 Гц** — ( $\text{с}^{-1}$ ), единица частоты периодических событий, 1 событие в секунду

У ЭВМ первого поколения типичное значение тактовой частоты было в пределах 100 КГц

Процессор	Год выпуска	Тактовая частота
Intel 4004	1971	740 КГц
Motorola 6800	1974	2 МГц
Zilog Z80	1976	2,5 МГц
Intel 80186	1982	6 МГц
Intel 80486 DX	1989	20 МГц
Intel 80486 DX4	1994	100 МГц
Pentium 4	2000	1,6 ГГц
Intel Xeon Westmere	2010	3,6 ГГц

# Примеры значений тактовой частоты

**1 Гц** — ( $\text{с}^{-1}$ ), единица частоты периодических событий, 1 событие в секунду

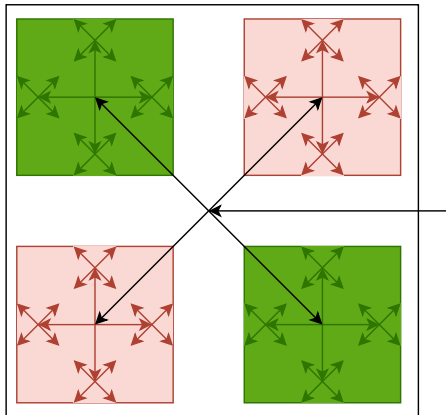
У ЭВМ первого поколения типичное значение тактовой частоты было в пределах 100 КГц

Процессор	Год выпуска	Тактовая частота
Intel 4004	1971	740 КГц
Motorola 6800	1974	2 МГц
Zilog Z80	1976	2,5 МГц
Intel 80186	1982	6 МГц
Intel 80486 DX	1989	20 МГц
Intel 80486 DX4	1994	100 МГц
Pentium 4	2000	1,6 ГГц
Intel Xeon Westmere	2010	3,6 ГГц

По идее, чем выше, тем «лучше», но у современного сложного процессора сигнал за 1 такт не успевает пройти даже от одной части кристалла к другой. Одно из косвенных решений — конвейеризация — позже

# Одновременная доставка тактового сигнала

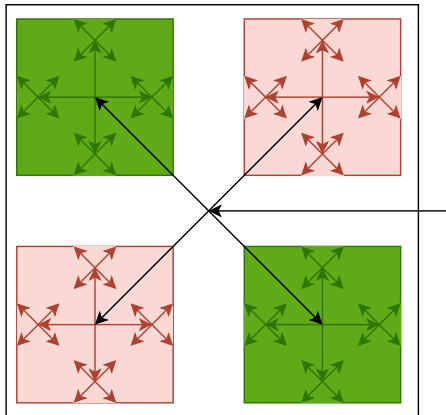
Желательно доставить тактовый сигнал во все блоки процессора одновременно



Тактовая сеть

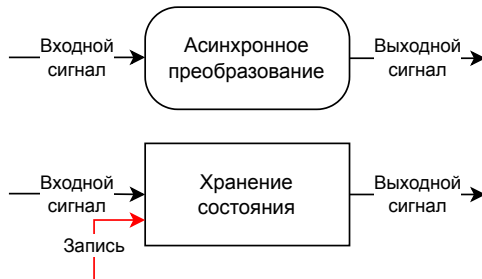
# Одновременная доставка тактового сигнала

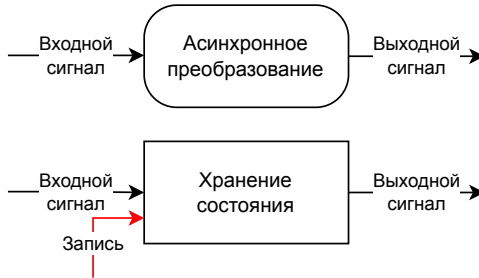
Желательно доставить тактовый сигнал во все блоки процессора одновременно



Тактовая сеть

Часть блоков могут быть отключены для экономии тактового сигнала (до 30% мощности процессора!) и потребляемой ими самими энергии



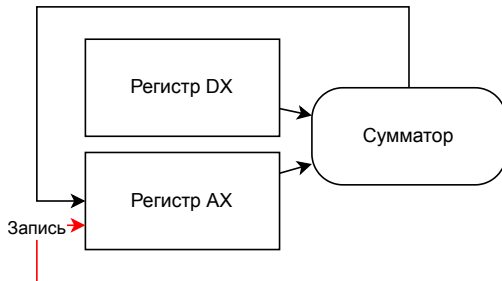


Примеры компонент:

- С состоянием — триггер, регистр
- Без состояния — сумматор, арифметико-логическое устройство (простое)



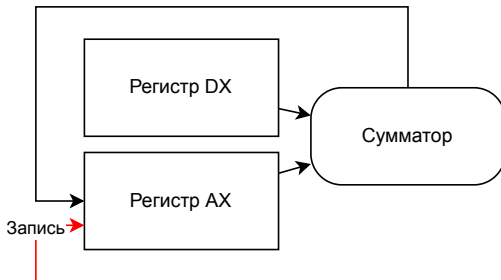
## Простой пример: сложение (I)



`add ax, dx`

Что здесь «плохо»?

## Простой пример: сложение (I)

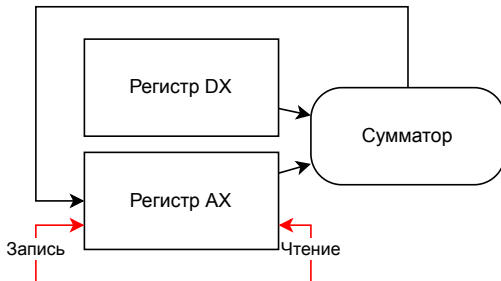


`add ax, dx`

Что здесь «плохо»?

Активный сигнал записи заставит данные «бегать по кругу» с неопределённой скоростью неопределённое число раз. Для корректной работы асинхронной схемы (сумматора) определённая скорость критична.

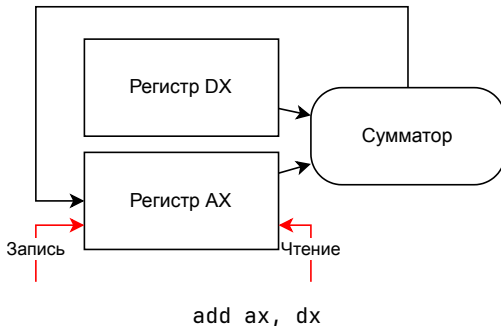
## Простой пример: сложение (2)



`add ax, dx`

В ранних компьютерах тактовый сигнал был двухфазный: для данного примера фаза  $\varphi_1$  — чтение из синхронных компонентов,  $\varphi_2$  — запись в синхронные компоненты, эти сигналы не пересекались по времени

## Простой пример: сложение (2)



В ранних компьютерах тактовый сигнал был двухфазный: для данного примера фаза  $\varphi_1$  — чтение из синхронных компонентов,  $\varphi_2$  — запись в синхронные компоненты, эти сигналы не пересекались по времени

Напоминает поочерёдную работу предсердий и желудочков сердца

Однофазный тактовый сигнал, компоненты синхронизируются по:

- Активному (высокому) значению
- Неактивному (низкому) значению
- Фронту импульса
- Спаду импульса

## Асинхронные ЭВМ

- Блок процессора / узел внутри ЭВМ подаёт сигнал по мере готовности результата
- Позволяют добиться большей производительности, но сложнее в проектировании и устройстве
- Соединённые компоненты либо работают за предсказуемое время, либо генерируют сигналы готовности друг для друга (очень упрощённо)

## Примеры (не экзотические)!

- ILLIAC I и II, GA144 (стековый, для Forth)
- Длинные асинхронные операции на современных процессорах, например, деление на RISC-процессорах

## Асинхронные ЭВМ

- Блок процессора / узел внутри ЭВМ подаёт сигнал по мере готовности результата
- Позволяют добиться большей производительности, но сложнее в проектировании и устройстве
- Соединённые компоненты либо работают за предсказуемое время, либо генерируют сигналы готовности друг для друга (очень упрощённо)

## Примеры (не экзотические)!

- ILLIAC I и II, GA144 (стековый, для Forth)
- Длинные асинхронные операции на современных процессорах, например, деление на RISC-процессорах
- Устройства расширения в «обычных» ЭВМ — выполняют длительные операции (например, с участием DMA), сообщают о выполнении команд и получают следующие по мере готовности

## Разрядность

- Разрядность процессора
- Разрядность ОЗУ



- **Разрядность** — обычно — количество бит в шине данных и в машинном слове
- **Машинное слово** — минимальная единица обмена данными между процессором и ОЗУ

# Понятие разрядности

- **Разрядность** — обычно — количество бит в шине данных и в машинном слове
- **Машинное слово** — минимальная единица обмена данными между процессором и ОЗУ

## А ещё обычно

- Количество бит в шине данных
- Количество бит в арифметических регистрах
- Размер целого числа, над которым аппаратно производится операция (машинное слово)

# Понятие разрядности

- **Разрядность** — обычно — количество бит в шине данных и в машинном слове
- **Машинное слово** — минимальная единица обмена данными между процессором и ОЗУ

## А ещё обычно

- Количество бит в шине данных
- Количество бит в арифметических регистрах
- Размер целого числа, над которым аппаратно производится операция (машинное слово)

## И иногда

- Количество бит в шине адреса и в адресных регистрах
- размер стандартного типа `int` в C (совсем не всегда, может зависеть от архитектуры, ОС и транслятора)

- **Внутренняя разрядность** — количество бит, из которых состоят регистры и шины между блоками процессора
- **Внешняя разрядность** — количество бит, из которых состоят шины компьютера

Обычно речь идёт об арифметических регистрах и шине данных, но понятия внутренней и внешней разрядности также применяются и к адресным регистрам и шине адреса

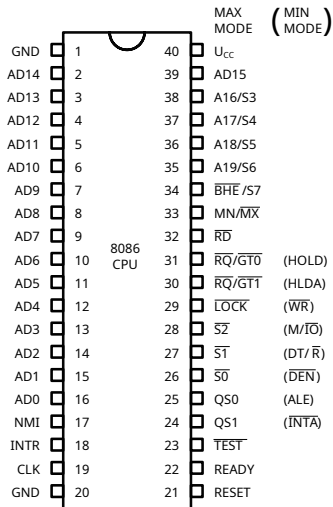
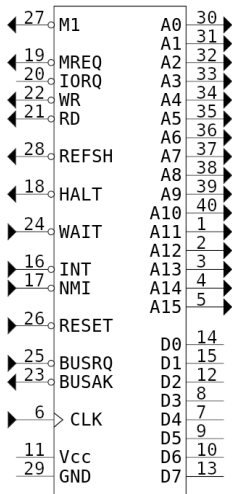
- Шина данных — 16 бит
- Арифметические регистры и операции — по 16 бит
  - Но `mul ax`, R/M считает 32-битный результат  $DX:AX \leftarrow AX * R/M$
- Адресные регистры — 16 бит (адресуют по 64 КиБ)
- Шина адреса — 20 бит (16-битный адрес складывается с адресом сегмента, это позволяет адресовать до 1 МиБ, об этом позже)

# Примеры, подтверждения и исключения (1): Intel 8086

- Шина данных — 16 бит
- Арифметические регистры и операции — по 16 бит
  - Но `mul ax`, R/M считает 32-битный результат  $DX:AX \leftarrow AX * R/M$
- Адресные регистры — 16 бит (адресуют по 64 КиБ)
- Шина адреса — 20 бит (16-битный адрес складывается с адресом сегмента, это позволяет адресовать до 1 МиБ, об этом позже)
- Intel 8088 (сделан позже 8086, первый процессор IBM PC)
  - Всё то же самое, но шина данных 8 бит

- Шина адреса — 16 бит
- Шина данных — 8 бит
- Арифметические регистры и операции — по 8 бит
  - Но `add hl, bc` считает 16-битный результат над парами регистров

### Примеры, подтверждения и исключения (3)

«Распиновка» Zilog Z80 [↗](#)

«Распиновка» Intel 8086 ↗




## Разрядность ОЗУ (1): зачем сделали Intel 8088?

Сделали позже 8086, а разрядность шины данных меньше.

# Разрядность ОЗУ (1): зачем сделали Intel 8088?


Сделали позже 8086, а разрядность шины данных меньше.

Оперативная память **30-контактные**  single in-line memory module — 8-битный

А тогда зачем сделали 16-битный 8086? Точнее, как он пользовался 8-битной памятью?

# Разрядность ОЗУ (1): зачем сделали Intel 8088?

Сделали позже 8086, а разрядность шины данных меньше.

Оперативная память **30-контактные**  single in-line memory module — 8-битный

А тогда зачем сделали 16-битный 8086? Точнее, как он пользовался 8-битной памятью?

- Один 16-битный модуль можно собрать из двух 8-битных. С 16-битными процессорами семейства x86 так и делали.
- В итоге «память вообще» получает номер «слова памяти», но слово может быть 8 или 16-битным, в зависимости от исполнения компьютера

Процессоры даны с внутренней / внешней разрядностью

- **SIMM 30-контактный** ↗ — 8 бит
  - X1 — i8088 16/8
  - X2 — i8086 16/16, i80186 16/16 , i80286 16/16, i386SX 32/16
- **SIMM 72-контактный** ↗ — 32 бит
  - X1 — i386 32/32, i486 32/32, i586 Overdrive 32/32 (специально на место 80486)
  - X2 — i586 32/64
- **DIMM (Dual in-line memory module)** ↗ 100-контактный — 64 бит
  - X1 — i586 \_32/64

## Способы экономии

- Если есть старая память или системная плата — можно поставить «урезанный» по внешней разрядности процессор — 8088, 386SX, 586 Overdrive
- Если есть старая память, на новую системную плату можно ставить старые модули меньшей разрядности парами (DIMM — 2xSIMM)

## Способы экономии

- Если есть старая память или системная плата — можно поставить «урезанный» по внешней разрядности процессор — 8088, 386SX, 586 Overdrive
- Если есть старая память, на новую системную плату можно ставить старые модули меньшей разрядности парами (DIMM — 2xSIMM)

## Способ повышения производительности

У некоторых процессоров (i586) внешняя разрядность выше внутренней для более производительного обмена данными с памятью. Про это позже в лекции про кэш.






# Параллельная ли шина адреса ОЗУ?

Или нет?..

# Параллельная ли шина адреса ОЗУ?

Или нет?..

Смешанная!

- **DRAM (SIMM 30)**  — память организована, как квадратная таблица  $2^{12} \times 2^{12}$ . Шина адреса 12 бит, адрес передаётся в два захода со вспомогательными сигналами *Row Address Strobe* и *Column Address Strobe*. Это позволяло упростить внутреннюю структуры микросхем и уменьшить количество выводов (а значит уменьшить корпуса)
- ...
- **EDO (SIMM 72)**  — зачатки конвейера, при чтении из памяти на шину данных выдаётся предыдущее значение, пока шина адреса передаёт номера строки и столбца. Несколько сигналов RAS и CAS, и возможность на разные биты шины данных одновременно выдавать данные с разных адресов 
- **SDRAM (DIMM)**  — уже целый несложный «протокол»: пока системная плата передаёт команды, память их исполняет
- ...
- **DDR 5**  — сложный протокол, внутренний конвейер команд



## Упражнения

- Найдите документацию по системной плате своего ПК, выясните, какие виды памяти и в каких сочетаниях в неё можно устанавливать
- Выясните разрядность шины адреса своего ПК — внутреннюю и внешнюю

## Вопросы

- Что такое тактовый сигнал, тактовая частота и тактовый генератор?
- Приведите примеры асинхронных операций, не управляемых тактами
- Что такое внутренняя и внешняя разрядность?
- Зачем Intel выпускали версии процессоров с пониженной внешней разрядностью?

# Вопросы



[EDU.DLUCIV.NAME](#) ↗