

# Память и ПЛИС

Луцив Дмитрий Вадимович

Кафедра системного программирования СПбГУ



1 Динамическая память

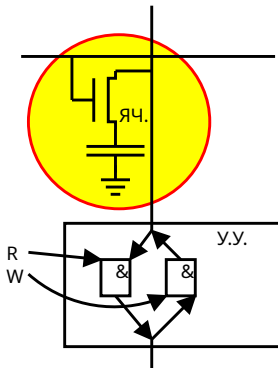
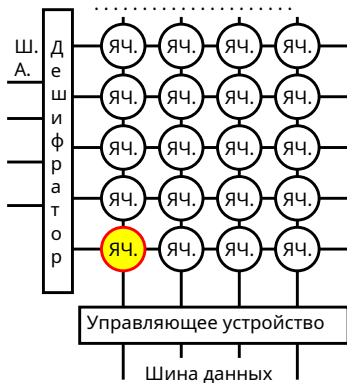
2 ПЛИС


# Динамическая память



- Статическая — на триггерах (обычно D). Применяется:
  - Где её нужно немного (микроконтроллеры, память настроек и часов и т.д.)
  - Где нужна память, не требующая дополнительных расходов времени (регистры, кэш)
- Динамическая — на схемах с [обычно кратковременной] памятью состояния
  - Ферритовые кольца или сердечники ↗ (могла быть энергонезависимой)
  - Беговая память ↗ — на принципе перемещения магнитных доменов в нанотрубках
  - Специальные ЭЛТ — Трубки Ульямса ↗
  - Конденсаторы



# DRAM: -1 приближение

В минус первом приближении (4-битный процессор, 1 слово в строке)



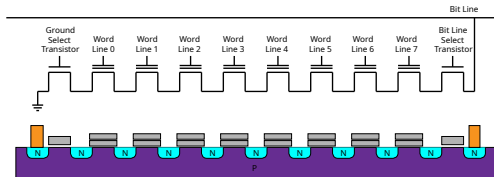
- Технологически конденсатор выполняется вместе с транзистором  и почти не требует дополнительного места: может быть использована ёмкость P-N перехода

- Технологически конденсатор выполняется вместе с транзистором  и почти не требует дополнительного места: может быть использована ёмкость P-N перехода
- По горизонтали — не шина данных с одним словом, а «широкая» строка , нужные участки которой управляющая схема мультиплексирует при чтении из памяти и демultipлексирует при записи (а «ненужные» просто перезаписывает)

- **Транзистор с плавающим затвором**  Эти полевые транзисторы могут при подаче *достаточно* высокого или *достаточно* низкого потенциалов (т.е. заметно больше 1 или заметно меньше 0) на управляющий вход запоминать своё состояние. После нескольких миллионов срабатываний транзистор необратимо портится (поэтому у Flash ограничено количество перезаписей). Этим эффектом, так же как и широкой петлёй гистерезиса (хотя природа этого совершенно иная), можно пользоваться для хранения данных.
- На основе таких транзисторов делается **флэш-память** 



# Flash-память: реализация NAND



- Для чтения на все слова, кроме читаемого, подаётся небольшой «приоткрывающий» потенциал. Ток течёт с соответствующих открытым транзисторам битовых линий в землю.
- Для программирования (открытия затвора, установки в бита 0) надо небольшим потенциалом «приоткрыть» все линии слов и подать сильный сигнал на пересечения нужных слова и бита.

- Из-за особенностей изготовления транзисторов, сброс (в 1) по словам или ещё большим блокам
- Для SSD введена операция **TRIM** [↗](#), которая говорит накопителю, что блок памяти свободен и может использоваться для оптимизации с целью увеличения ресурса перезаписи

- Из-за особенностей изготовления транзисторов, сброс (в 1) по словам или ещё большим блокам
- Для SSD введена операция **TRIM** [↗](#), которая говорит накопителю, что блок памяти свободен и может использоваться для оптимизации с целью увеличения ресурса перезаписи
- Альтернативная технология — **память на основе изменений фазового состояния халькогенидов** [↗](#) — обладает потенциально лучшими характеристиками, уже используется коммерчески (Например, кэш для ФС Intel Optane™), но тоже со временем деградирует

ПЛИС

# Что это и зачем?

## Что это?

- **Программируемые логические интегральные схемы (Programmable Logic Device)** — интегральные схемы, физический уровень которых (реализующий логику) можно задавать программно
  - программируются связи между компонентами схемы

## Зачем?

- Применяются для эффективной реализации (быстродействие — заметно медленнее серийных микросхем, существенно быстрее микропрограмм, принципиально быстрее обычной программной реализации) специфических задач
  - прототипирование, единичное или мелкосерийное производство
  - микропроцессоры встроенных ЭВМ

- Простые ПЛИС — логические функции, в т.ч. довольно сложная логика, но явно задаются через вентили «и», «или», «не»
- CPLD (Complex Programmable Logic Device) — несколько более высокоуровневые, содержат внутренние шины, более сложные элементы
- Программируемая пользователем вентиляционная матрица (Field-Programmable Gate Array, FPGA) — содержит готовые регистры, компоненты АЛУ и т.д.

## Пример: ПЛИС типа GAL (Gateway Array Logic)

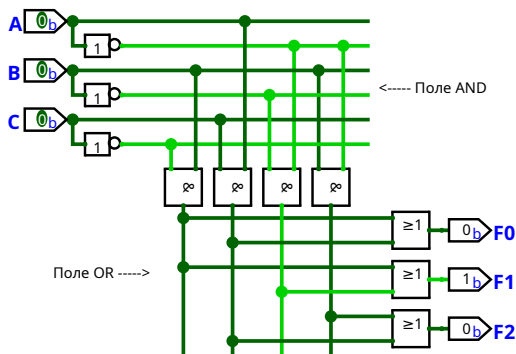


Схема на примере вычисляет функции, представленные в ДНФ:

$$F_0 = B \wedge \neg C \vee A \wedge C$$

$$F_1 = B \wedge \neg C \vee \neg A \wedge \neg B$$

$$F_2 = A \wedge C \vee \neg A \wedge B$$

- Входов у вентиля обычно больше, чем 2
- Как реализуются переключки в полях OR и AND?
  - «перезжиганием» (одноразовое программирование); мультиплексором с управлением статической памятью; транзистором с плавающим затвором



# Как программировать ПЛИС и чего можно добиться?

- Специальные языки, например Verilog Hardware Definition Language
- САПР (даже Logism)
  - Т.е. почти всё, что есть (было и будет) в данном курсе, можно сделать на ПЛИС
- **Hardware-Software CoDesign** — подход, подразумевающий совместную разработку специализированных ПО и оборудования
  - На кафедре системного программирования СПбГУ — Булычев, Медведев, Терехов:  
<https://scholar.google.com/scholar?q=HasCOL+SPbU>

- Как организована динамическая память?
- Как организована NAND-память?
- Что такое ПЛИС? Зачем они нужны и какие бывают?
- Опишите структуру ПЛИС типа GAL с полями AND и OR
- Как программируются ПЛИС?

# Вопросы



[EDU.DLUCIV.NAME](https://edu.dluciv.name) 