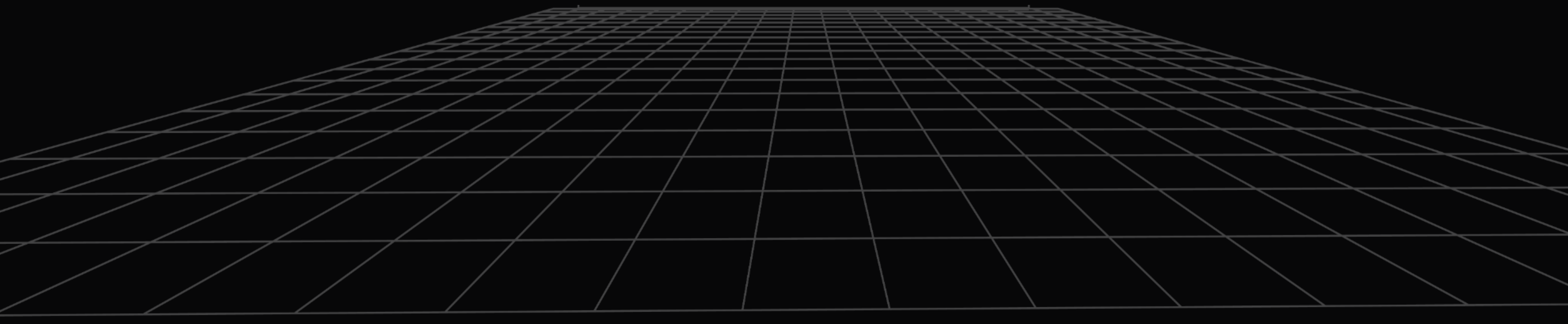


ОСНОВЫ COMPUTER SCIENCE

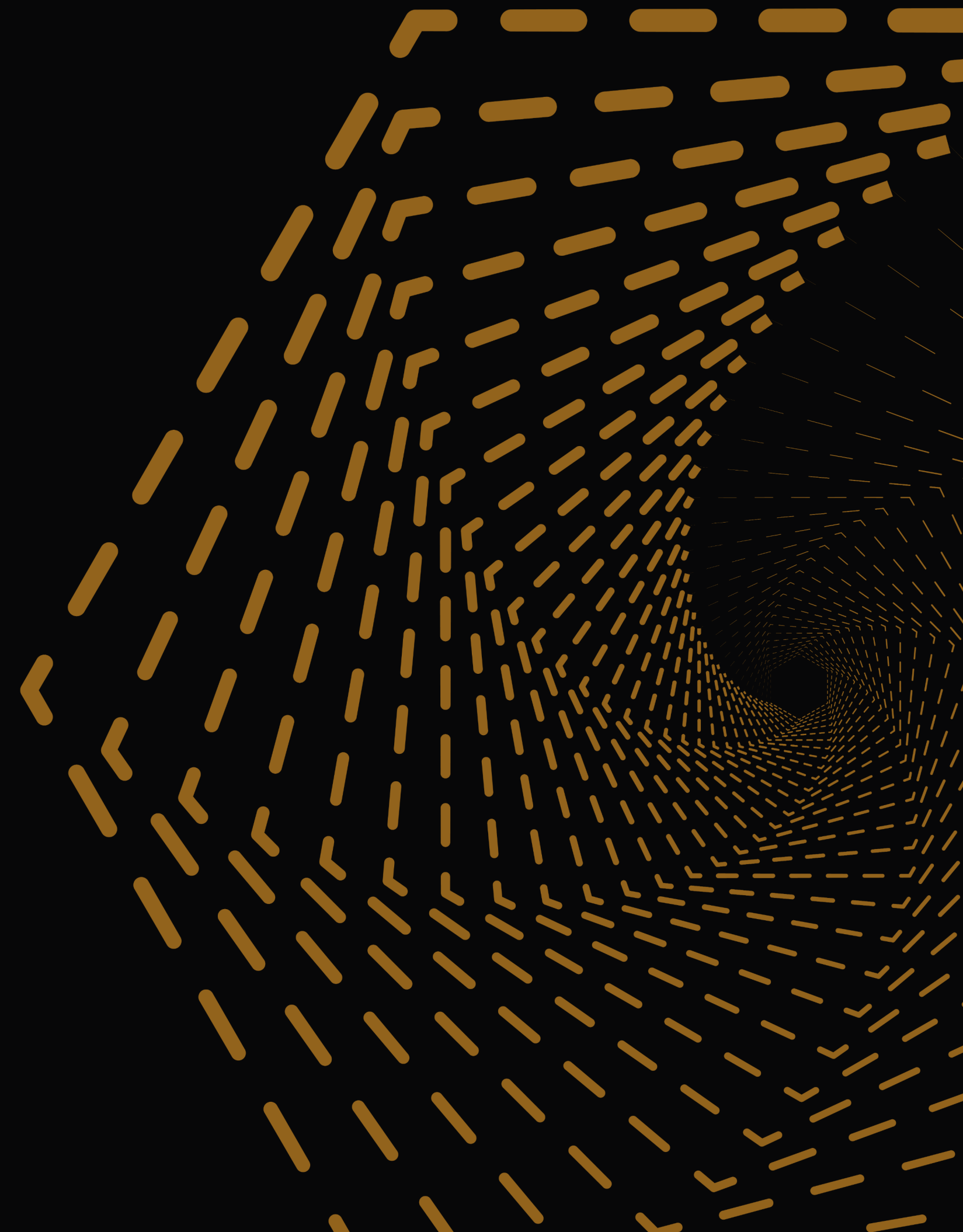
Computer architecture

ПРОВЕРЬ ЗАПИСЬ

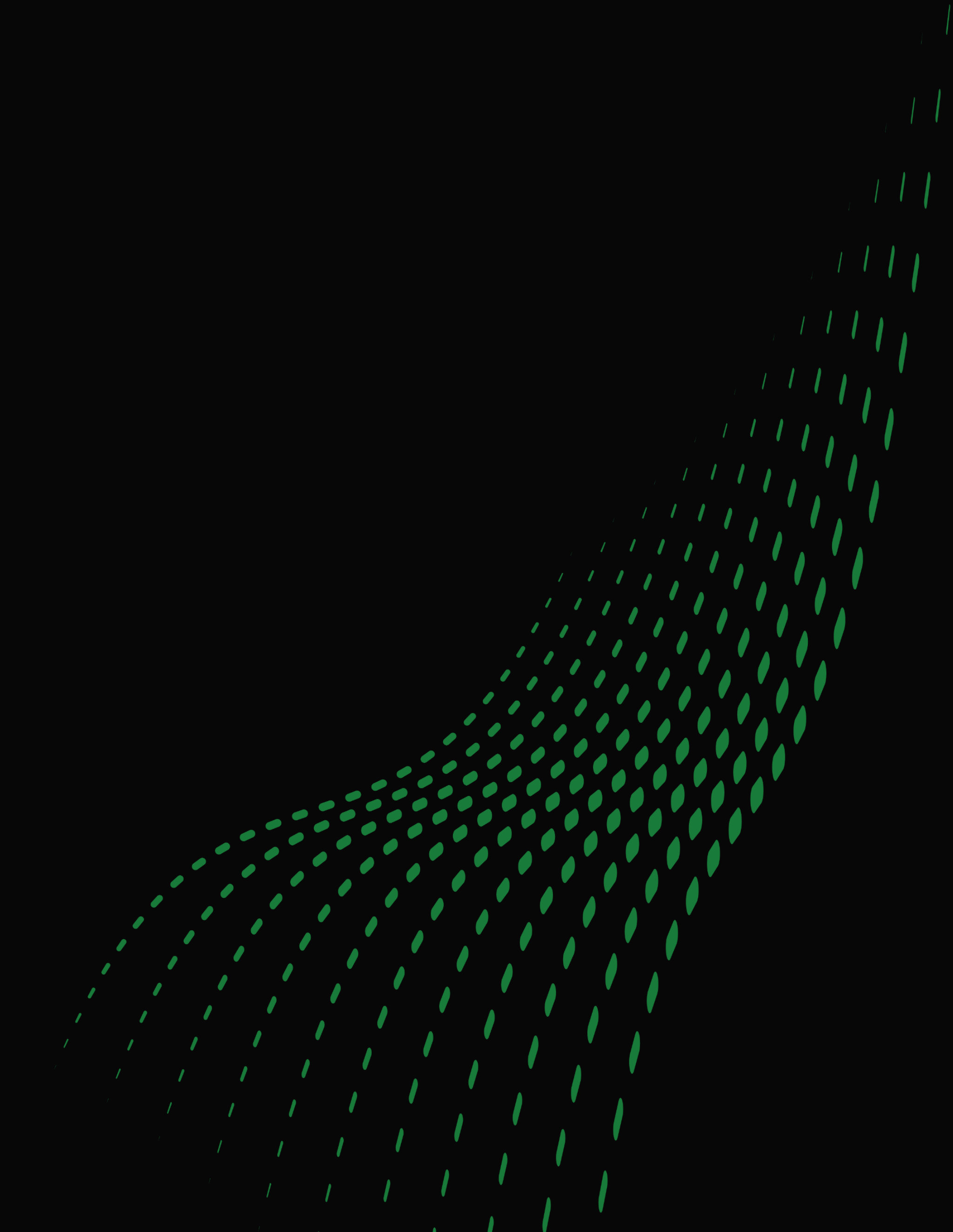


ПРАВИЛА ЗАНЯТИЯ

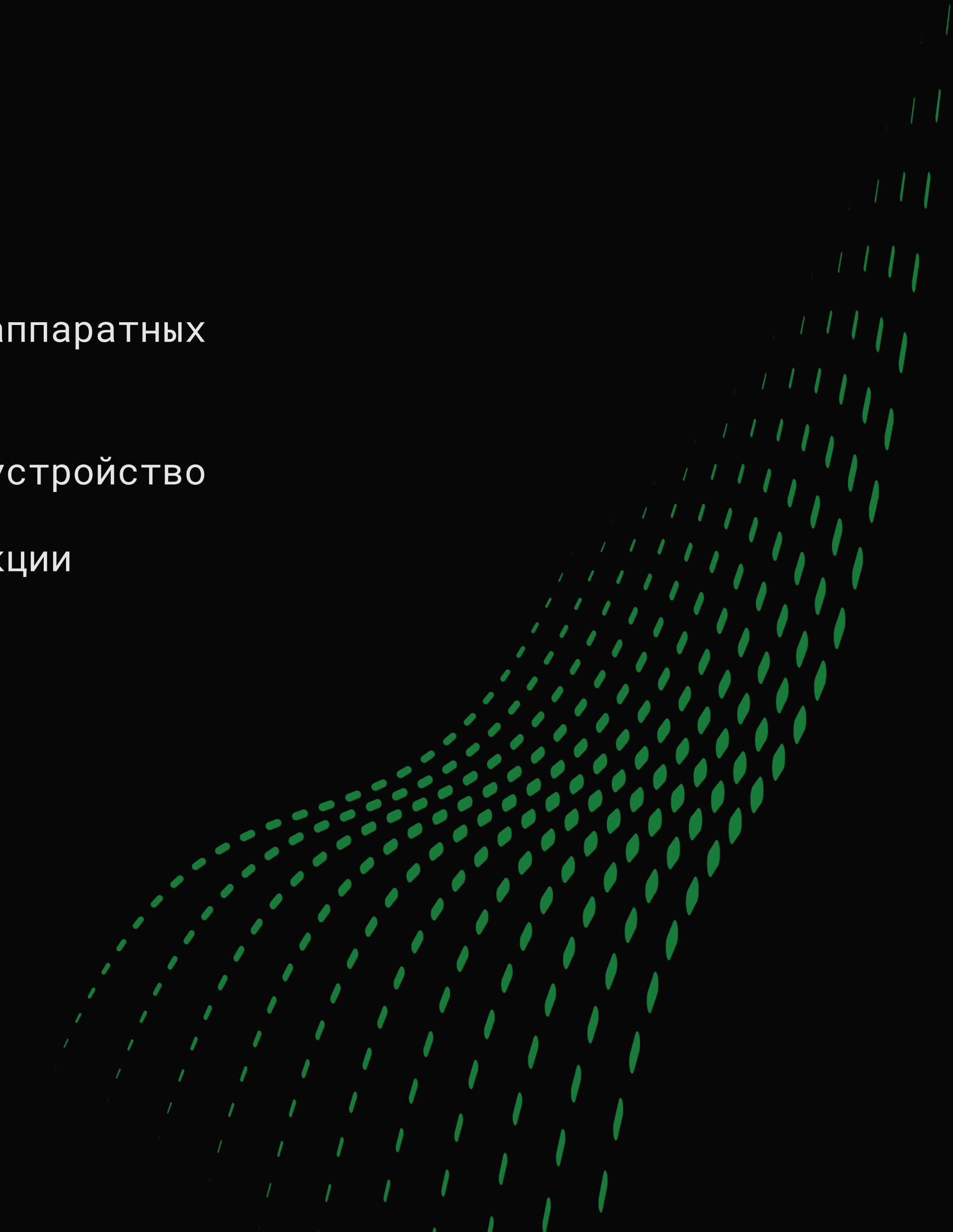
1. вопросы в чате можно задавать
в любое время
2. вопросы голосом задаем
по поднятой руке в Zoom
3. ответы на вопросы будут
в запланированных местах



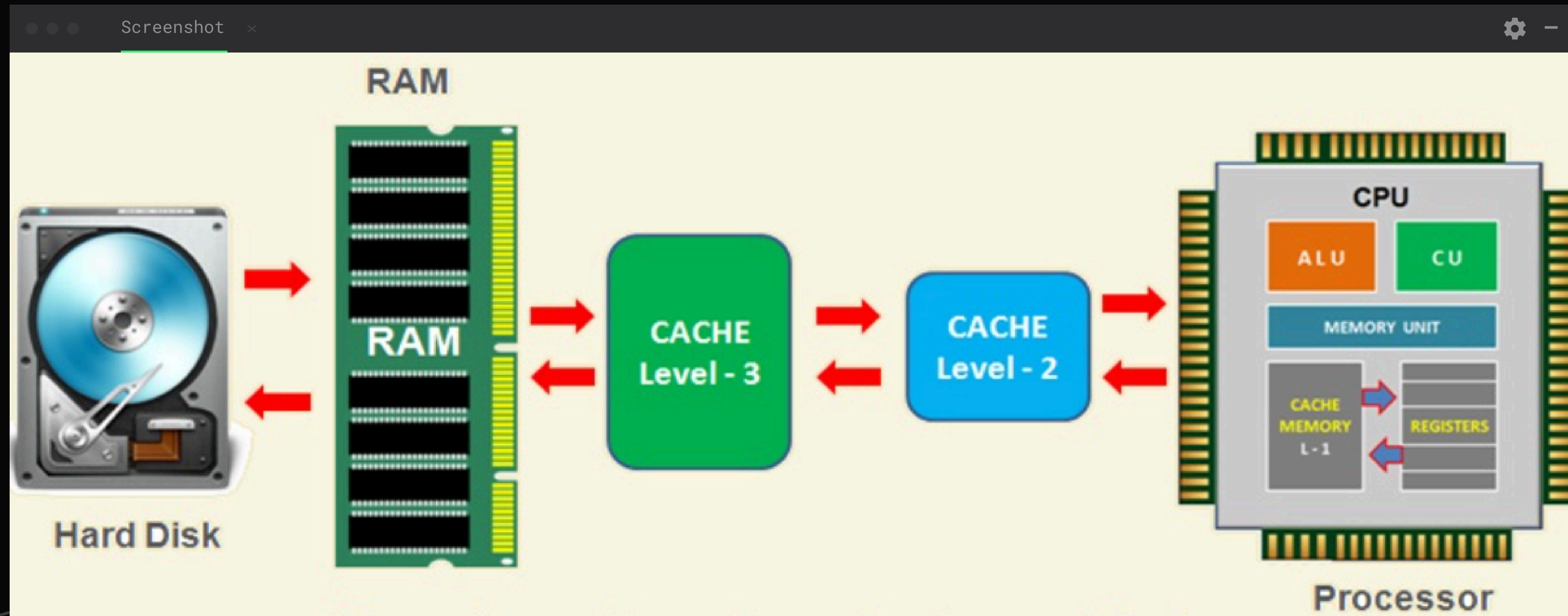
ПЛАН ЛЕКЦИИ

1. Мотивация изучения
 2. Bird's-eye view plan
 3. Организация чипсетов
 4. Central processing unit
 5. Типы команд
 6. Instruction Set Architecture
 7. Демонстрация процессора на Verilog
 8. Конвейерная архитектура, Superscalar, VLIW
 9. Устройство памяти, регистры
 10. Таксономия Флинна, SMT, HyperThreading
- 

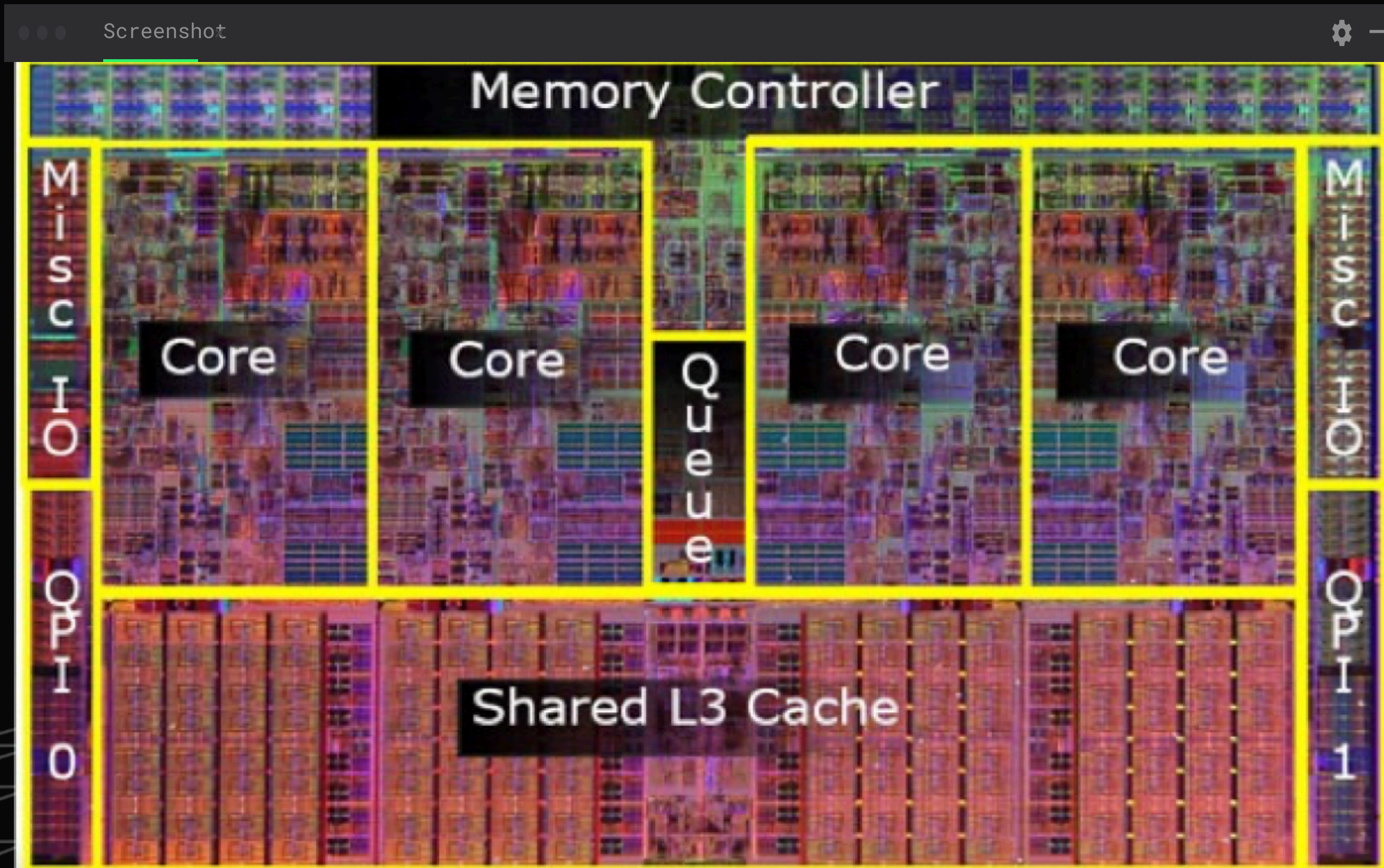
МОТИВАЦИЯ

1. Возможность оптимизаций
 2. Многие типовые конструкции языков сделаны на основании аппаратных интерфейсов
 3. Возможность видеть внешние проявления, зная внутреннее устройство
 4. Способность отлаживать поведение на любом уровне абстракции
 5. Полезность понимания того, что используется
- 

BIRD'S-EYE VIEW PLAN

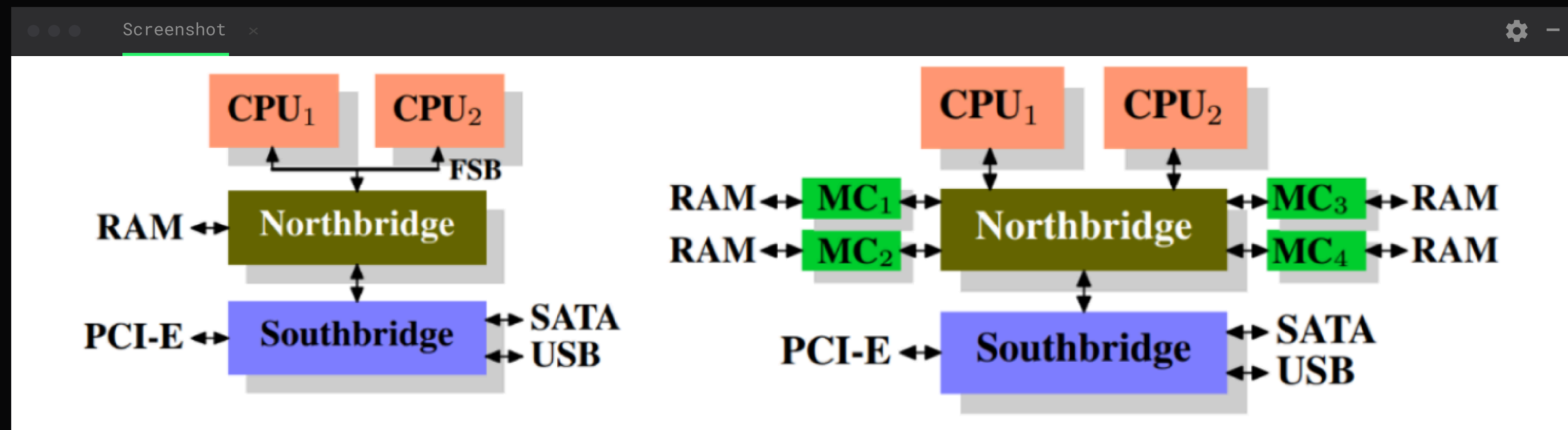


BIRD'S-EYE VIEW PLAN



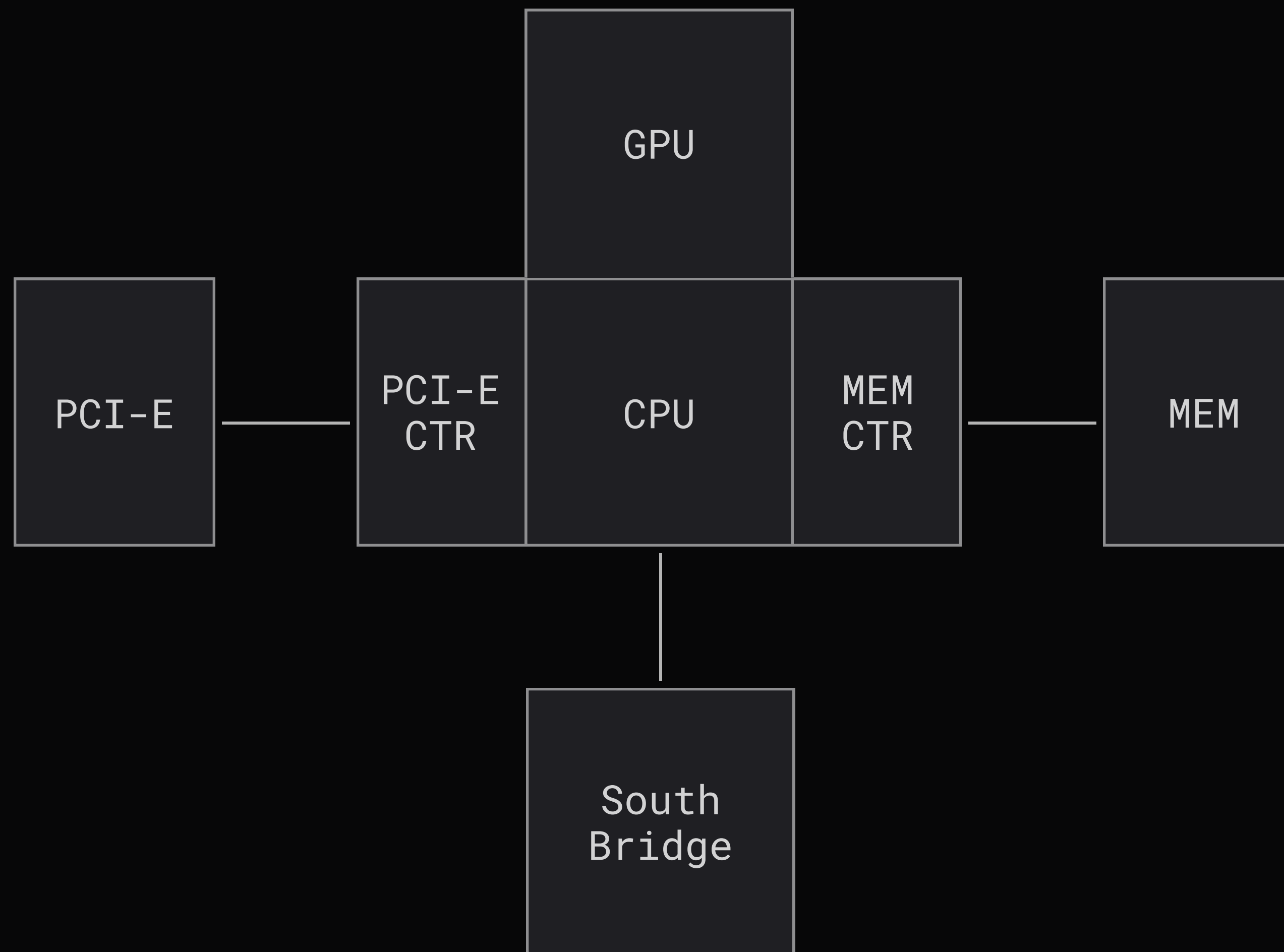
ОРГАНИЗАЦИЯ ЧИПСЕТОВ

Какие недостатки?

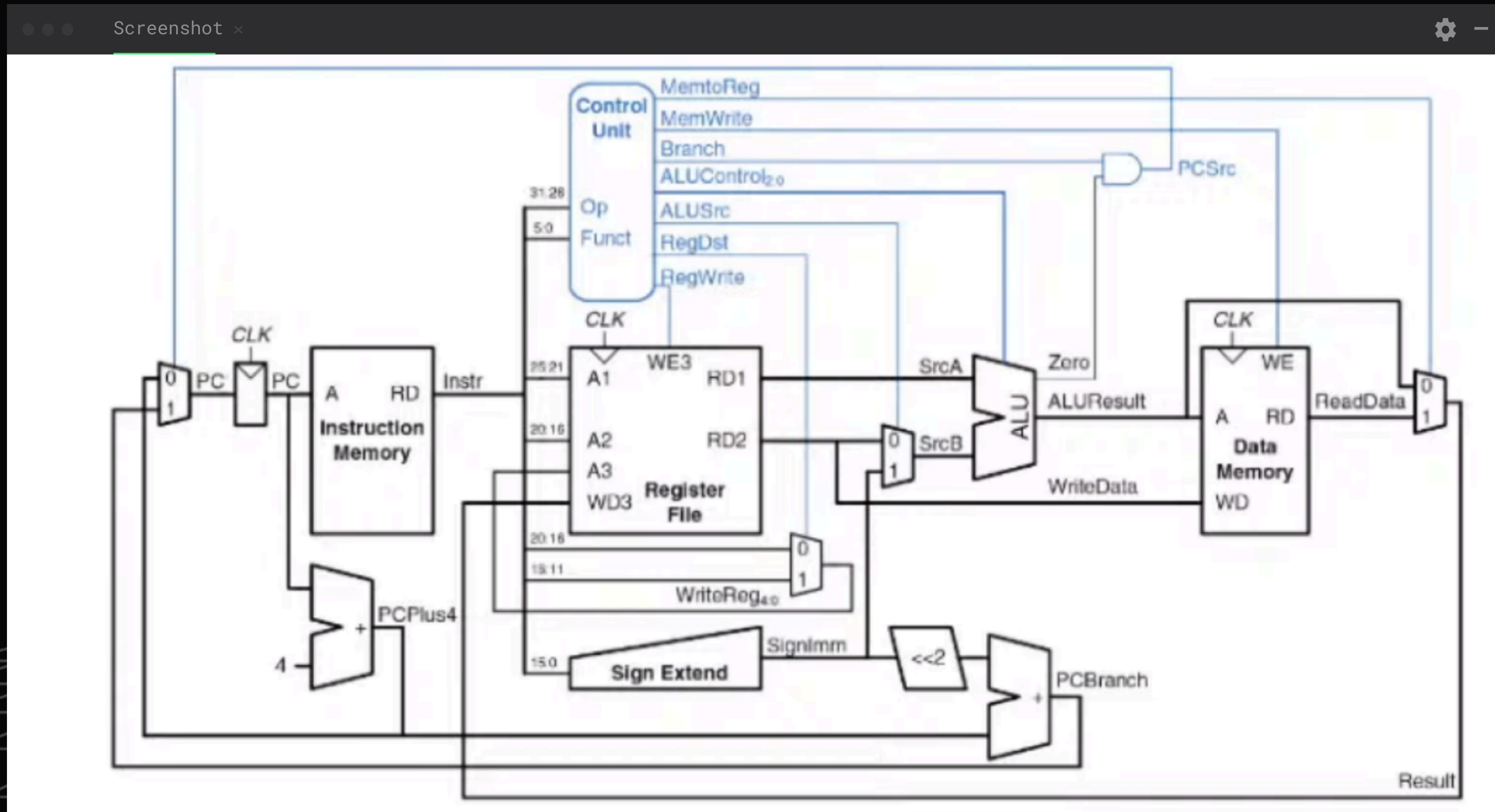


ОРГАНИЗАЦИЯ ЧИПСЕТОВ

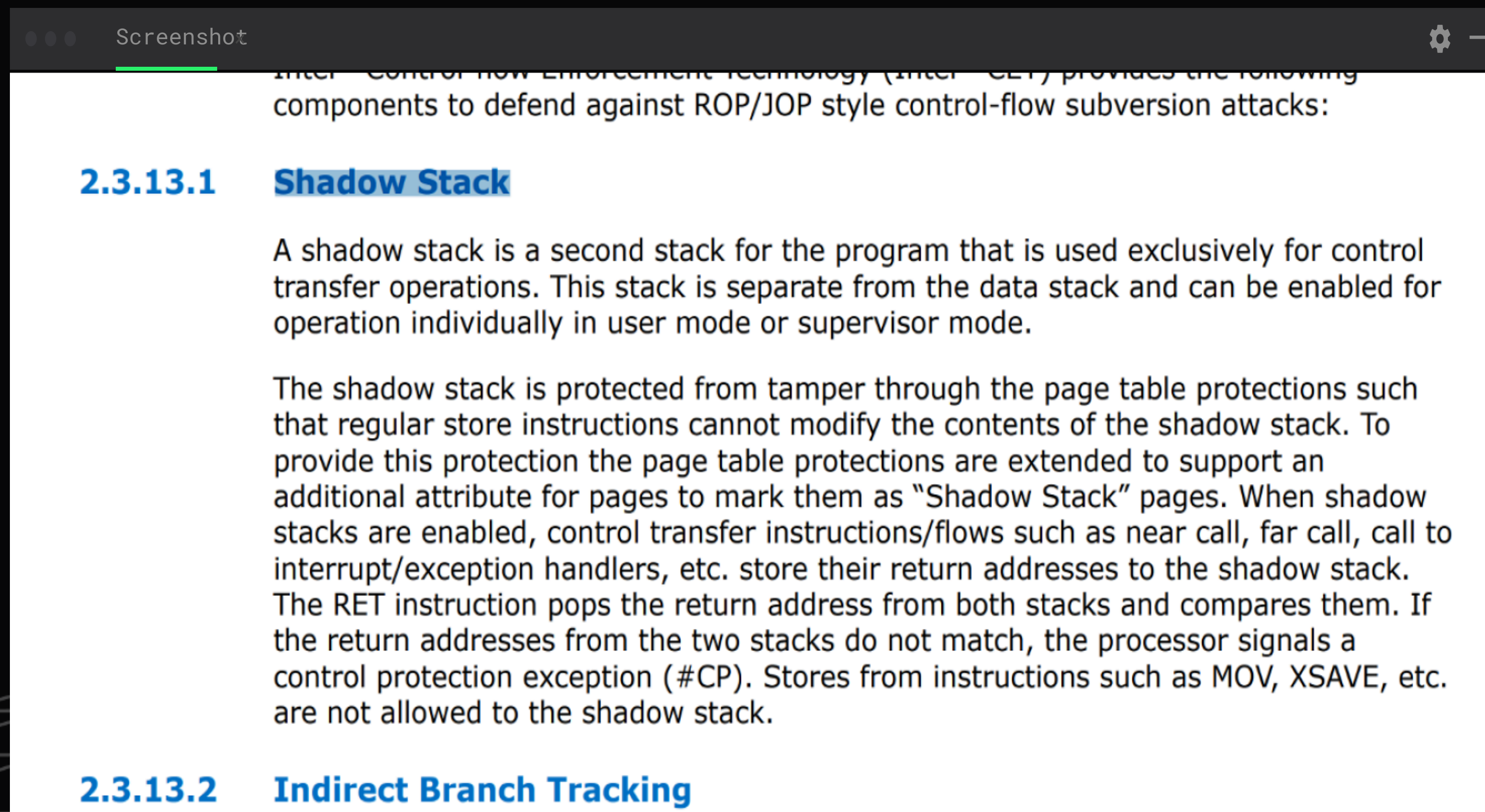
System on chip - SoC



CENTRAL PROCESSING UNIT

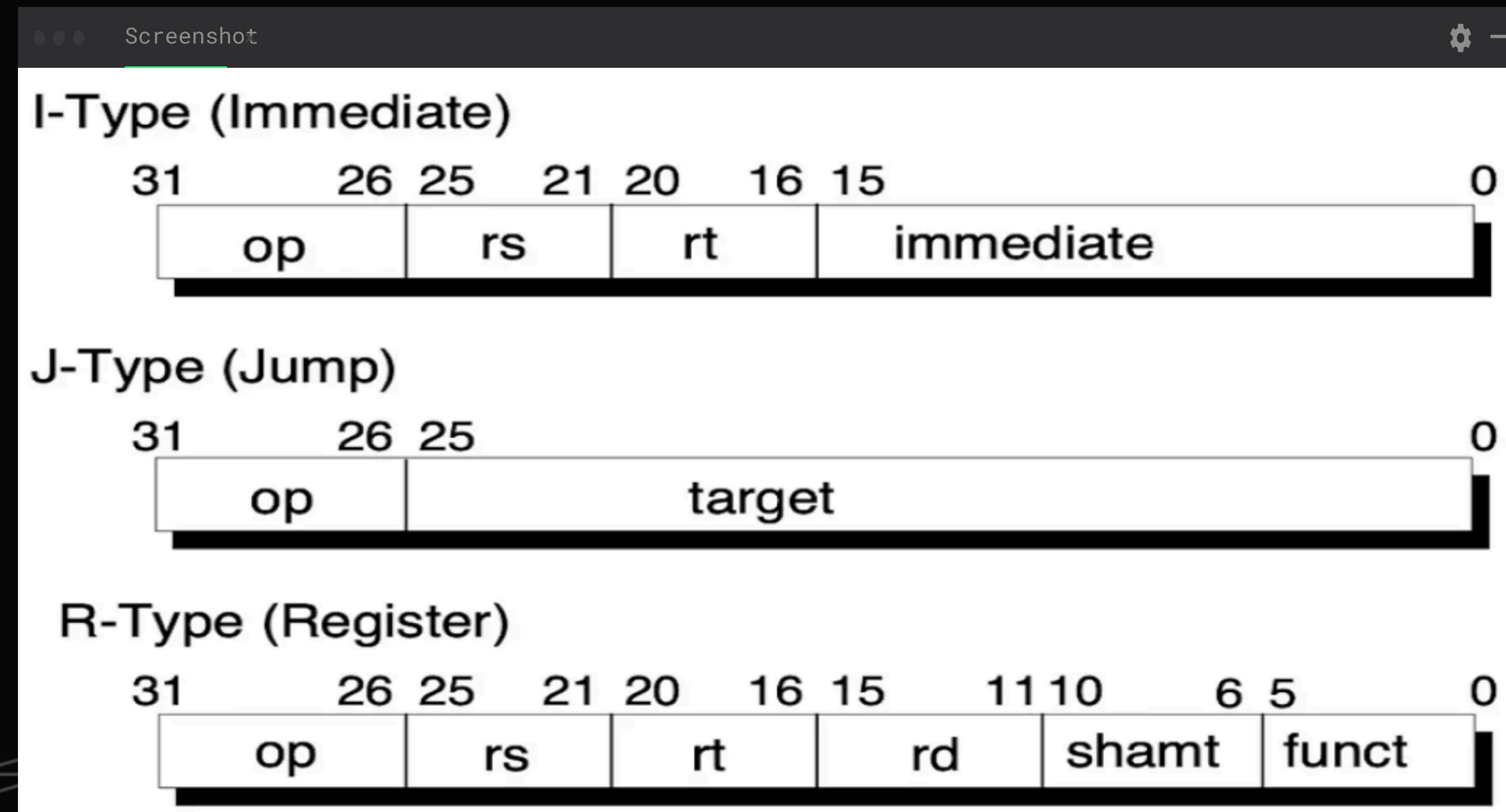


РЕАЛЬНЫЕ ПРОЦЕССОРЫ ГОРАЗДО СЛОЖНЕЕ



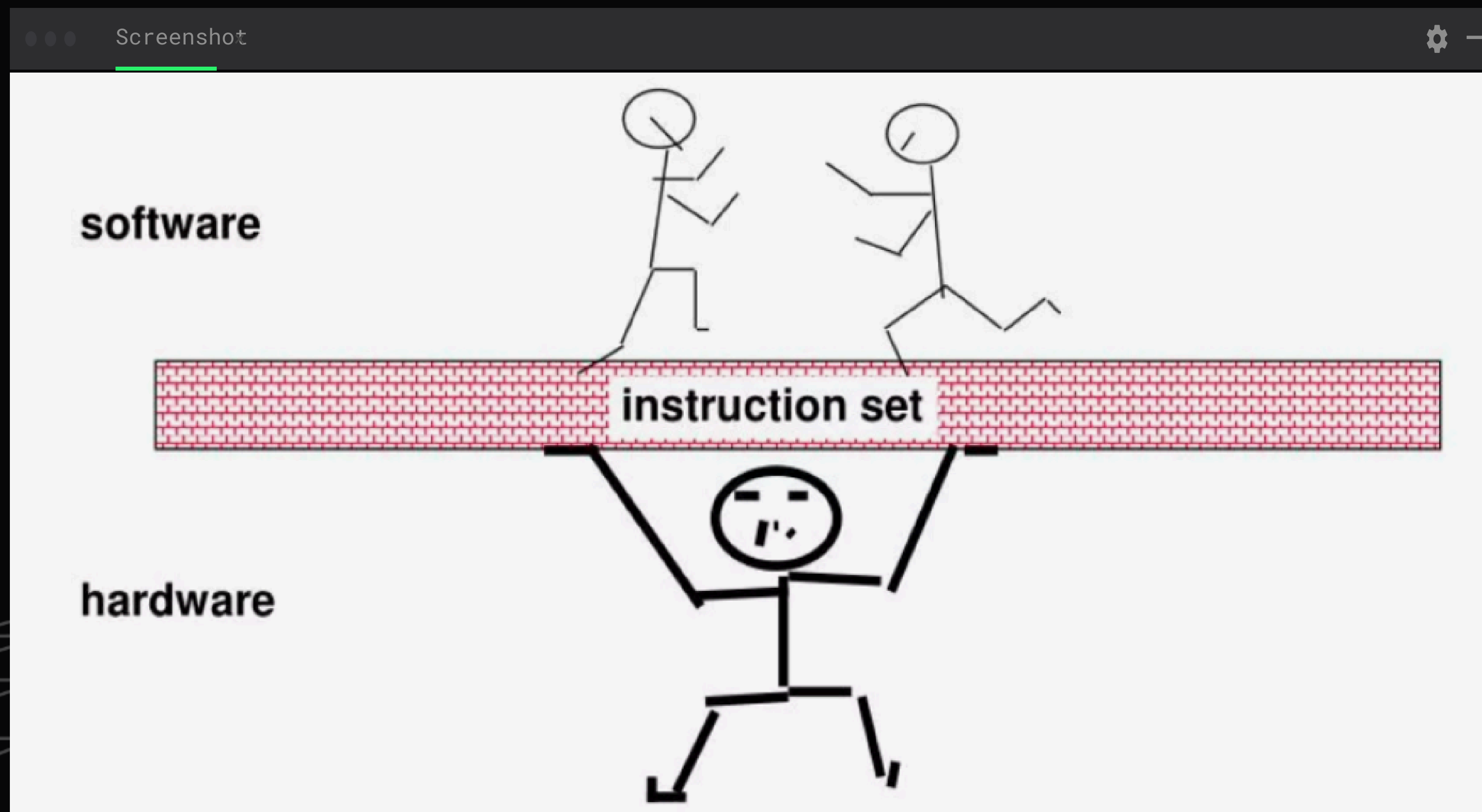
ТИПЫ КОМАНД

MIPS example



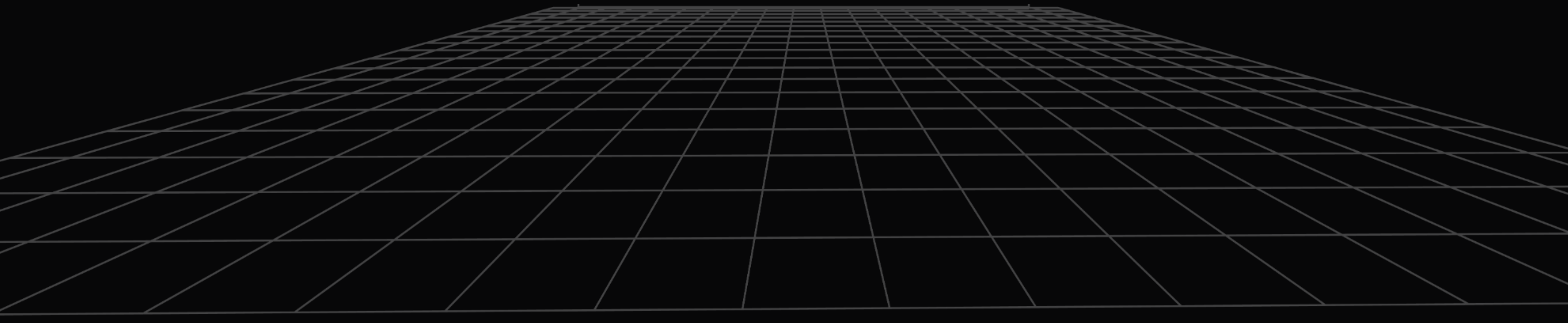
РЕАЛЬНЫЕ ПРОЦЕССОРЫ ГОРАЗДО СЛОЖНЕЕ

MIPS example



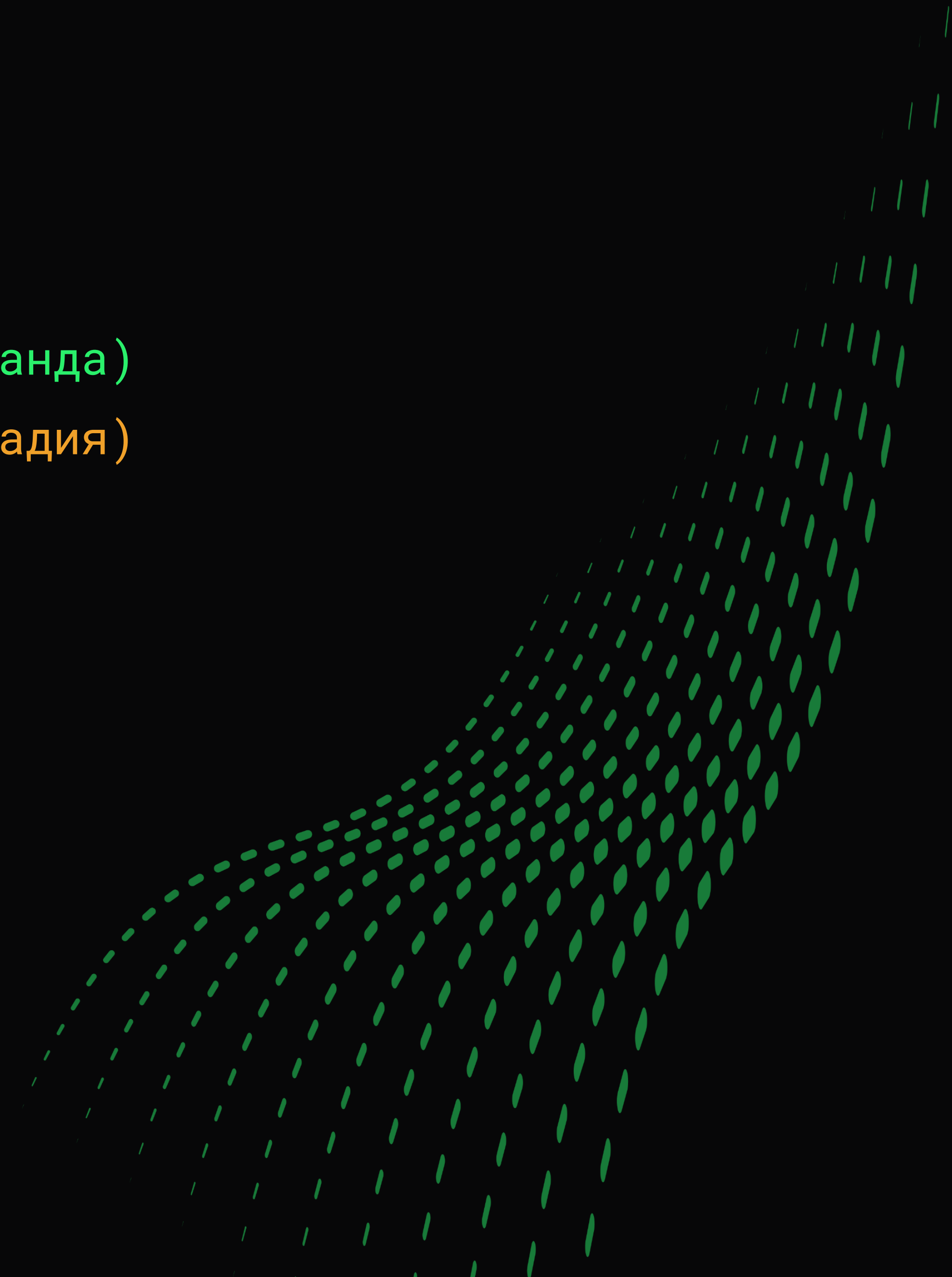
DEMO

Процессор на Verilog

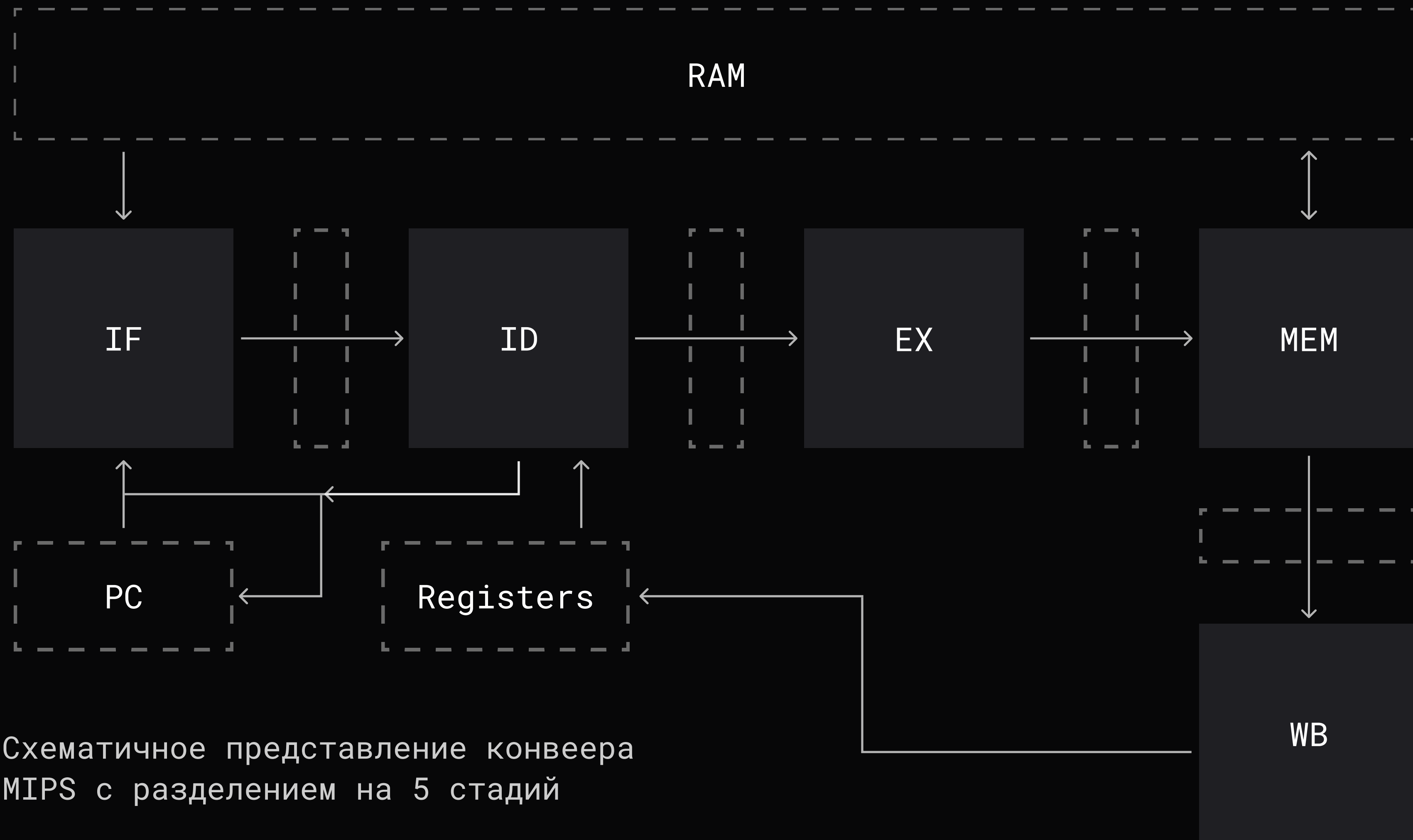


КОНВЕЙЕРНАЯ МИКРОАРХИТЕКТУРА

1. Однотактная микроархитектура (1 такт – одна команда)
2. Многотактная микроархитектура (1 такт – одна стадия)



КОНВЕЙЕРНАЯ МИКРОАРХИТЕКТУРА



КОНВЕЙЕРНАЯ МИКРОАРХИТЕКТУРА

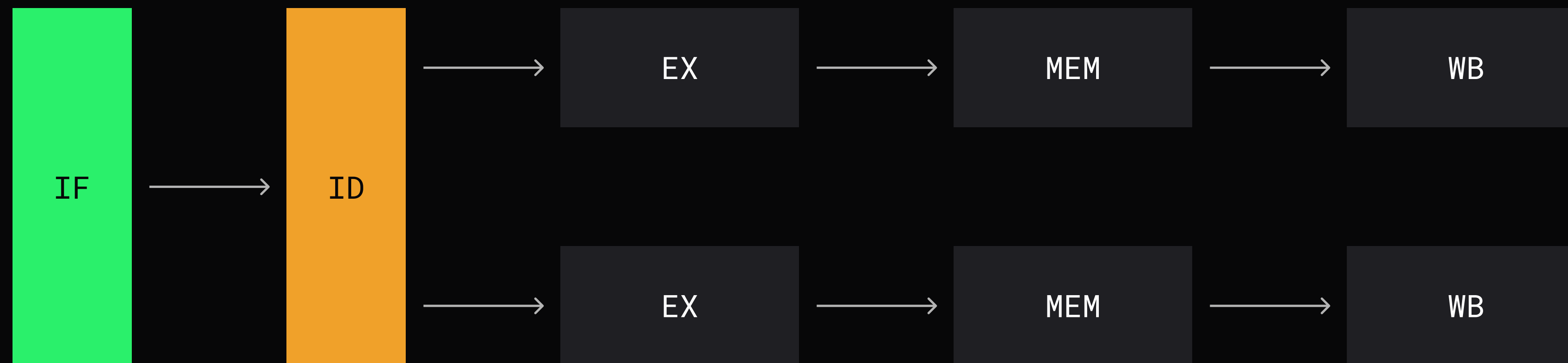


- Data hazards
 - Structural hazards
 - Control hazards
- (+ Branch predictor)

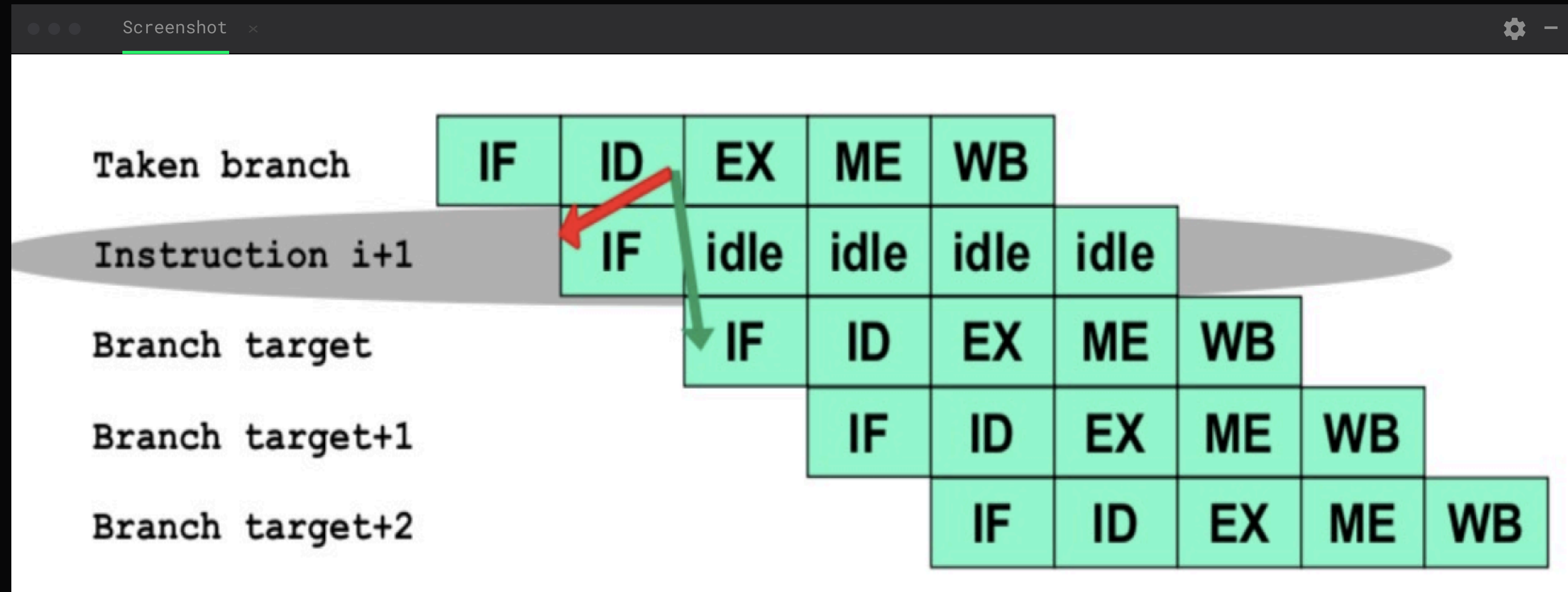
КОНВЕЙЕРНАЯ МИКРОАРХИТЕКТУРА

Very long instruction word (VLIW)

Pros and cons?

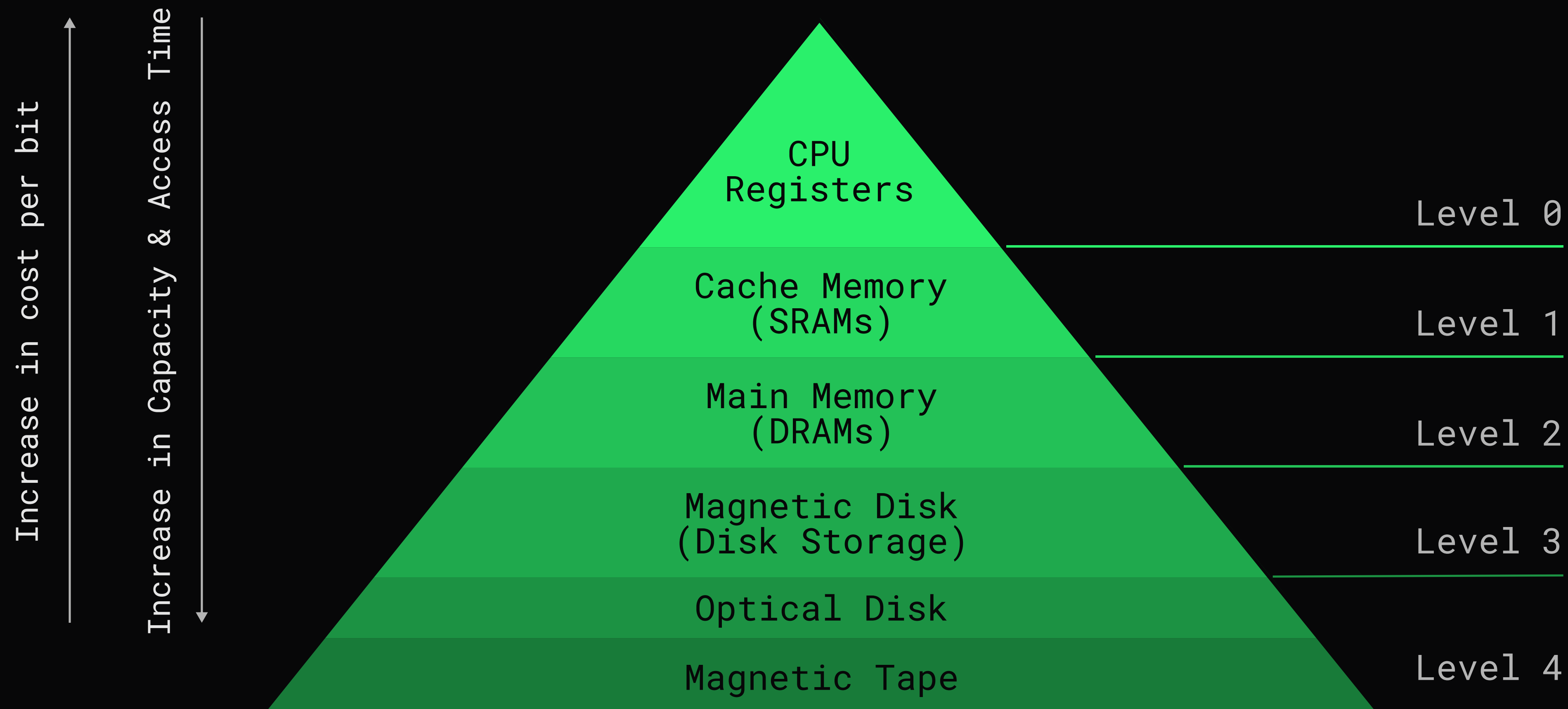


BRANCH PREDICTION



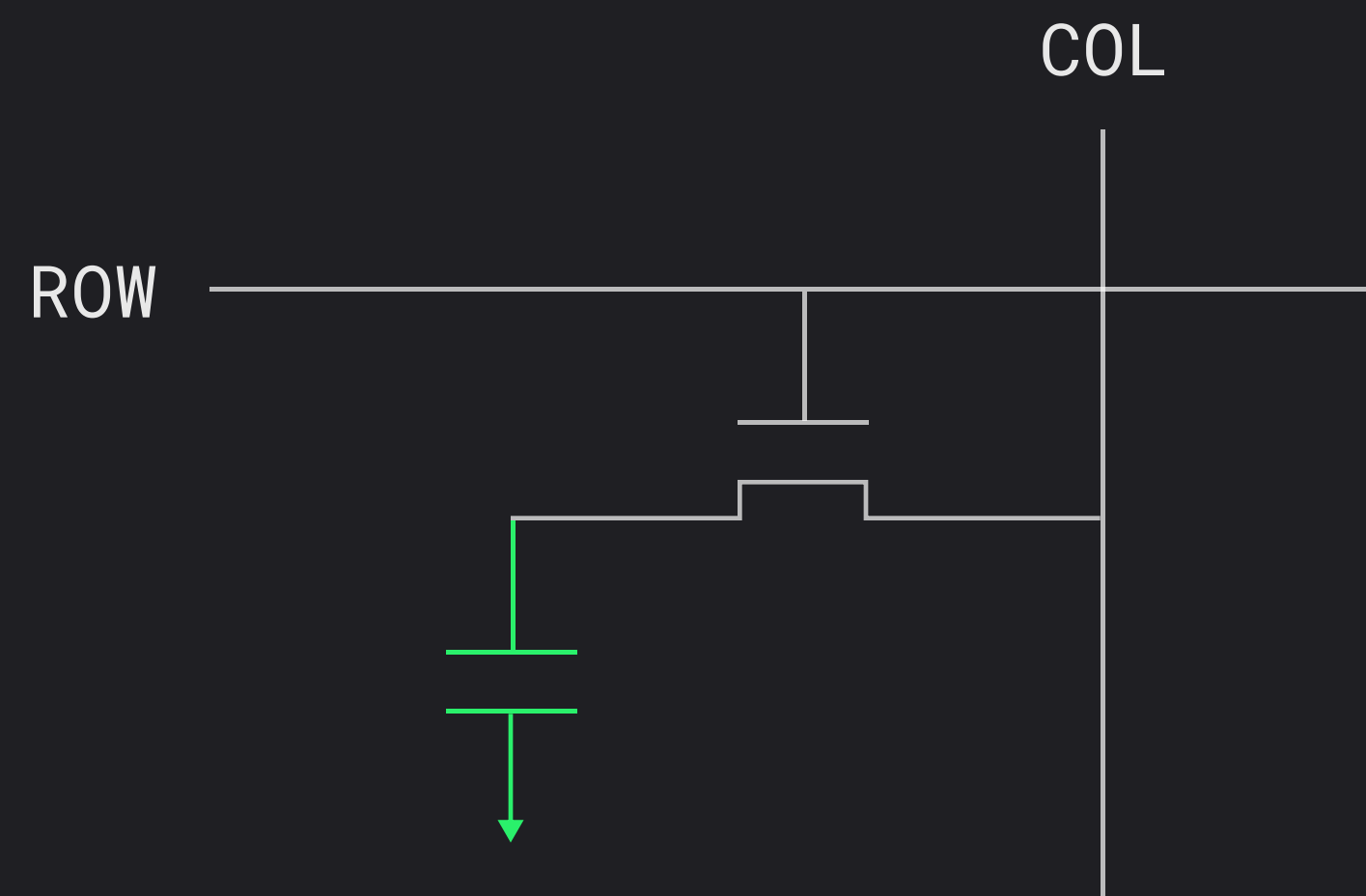
УСТРОЙСТВО ПАМЯТИ

Memory Hierarchy Design

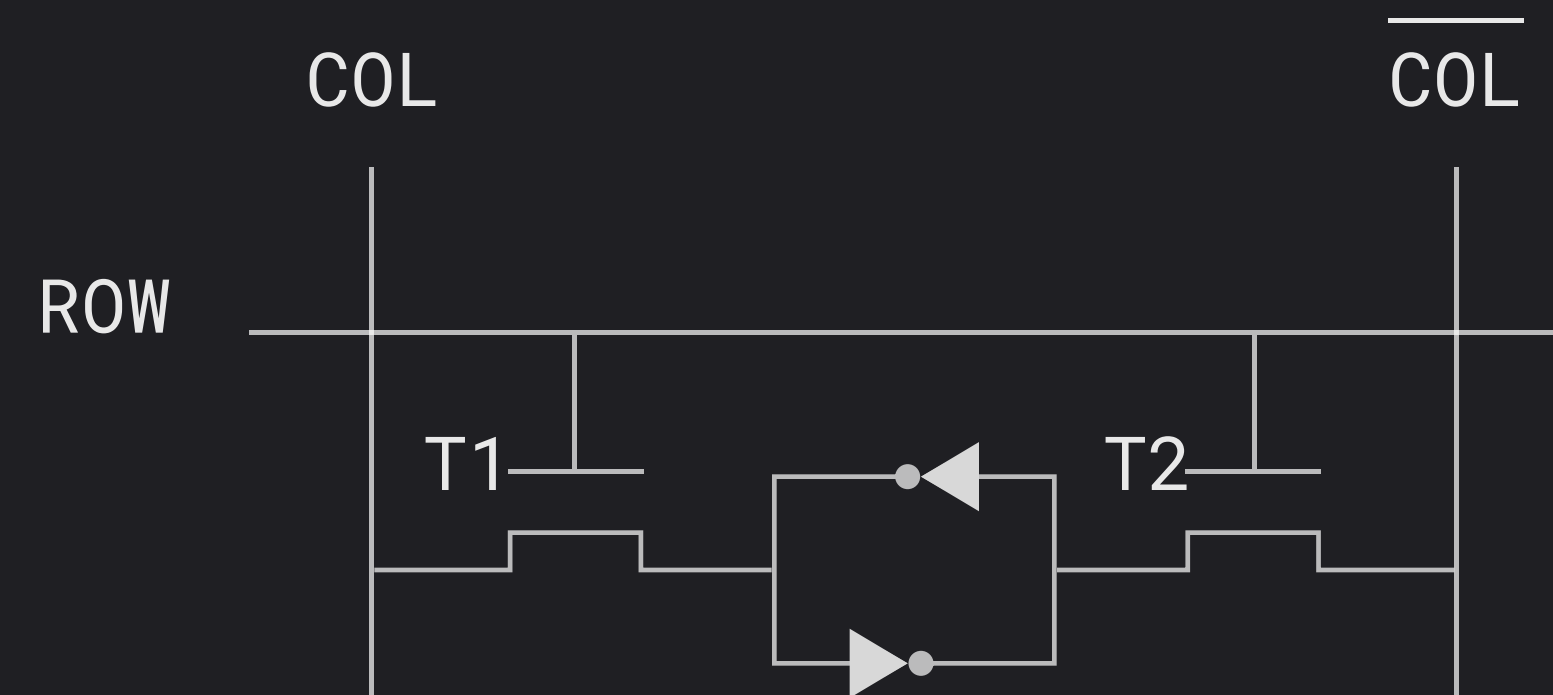


ЯЧЕЙКИ ПАМЯТИ

Где что используется и почему?

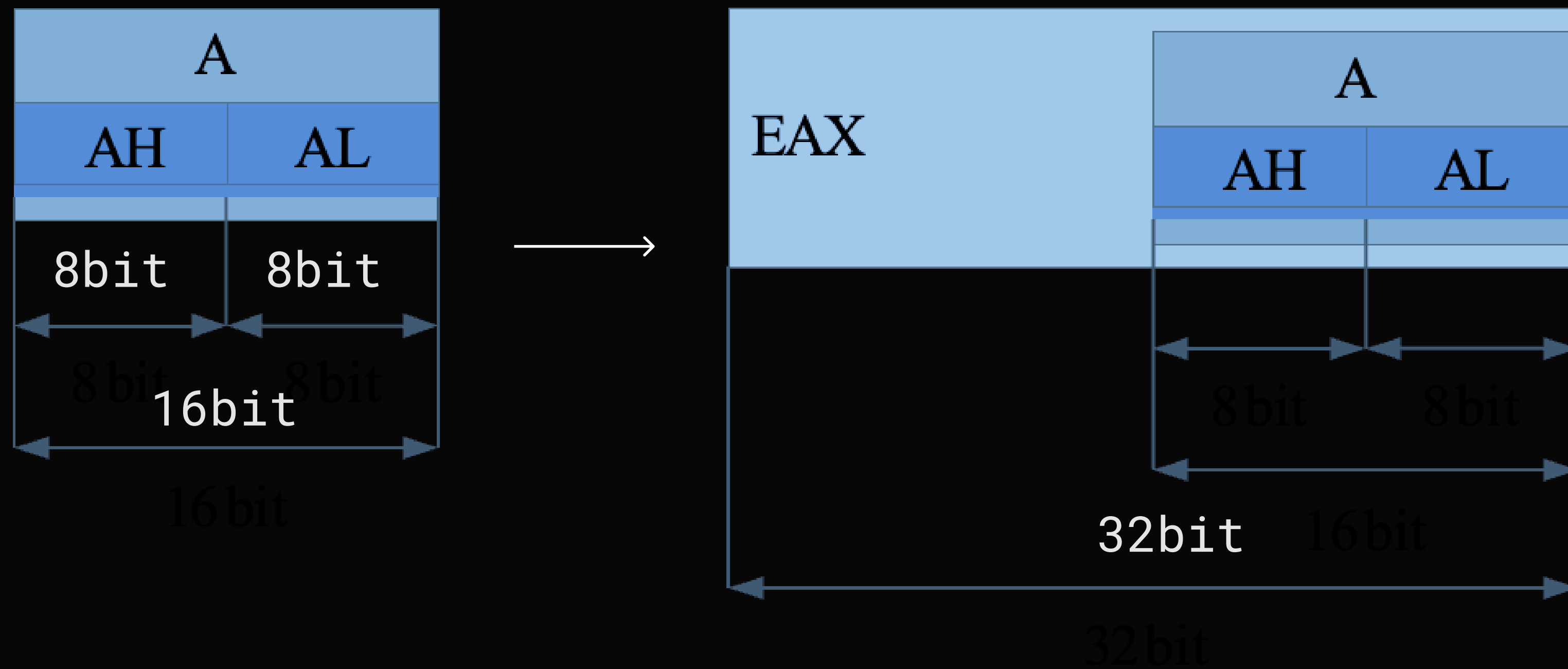


DRAM



SRAM

УСТРОЙСТВО РЕГИСТРОВ X86



УСТРОЙСТВО РЕГИСТРОВ X86



32bit

УСТРОЙСТВО РЕГИСТРОВ X86

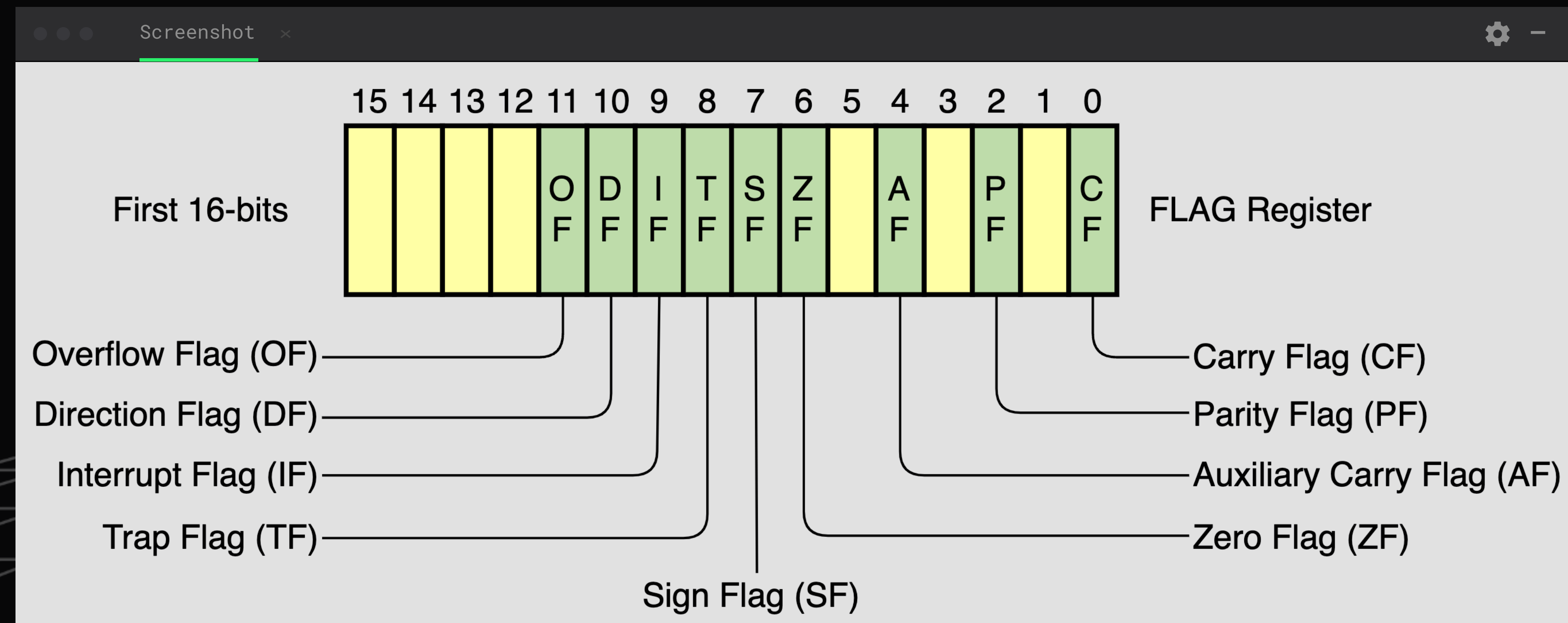
RAX	EAX	AX AH AL	R8	R8D	R8W R8B
RBX	EBX	BX BH BL	R9	R9D	R9W R9B
RCX	ECX	CX CH CL	R10	R10D	R10W R10B
RDX	EDX	DX DH DL	R11	R11D	R11W R11B
RSP	ESP	SP SPL	R12	R12D	R12W R12B
RBP	EBP	BP BPL	R13	R13D	R13W R13B
RSI	ESI	SI SIL	R14	R14D	R14W R14B
RDI	EDI	DI DIL	R15	R15D	R15W R15B

64bit

УСТРОЙСТВО РЕГИСТРОВ X86

Как реализовывать операции сравнения
и инструкции управления?

Регистр флагов



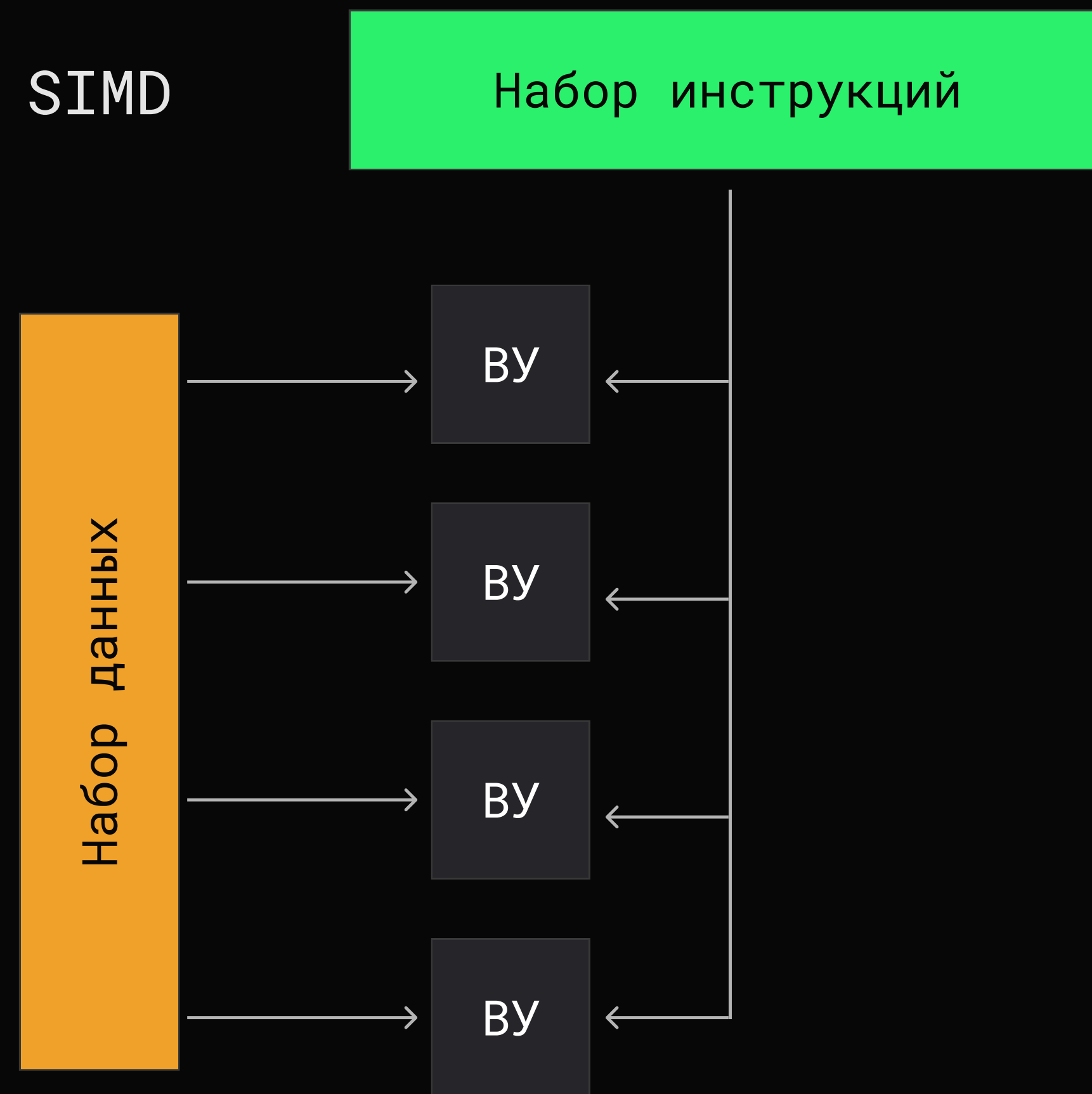
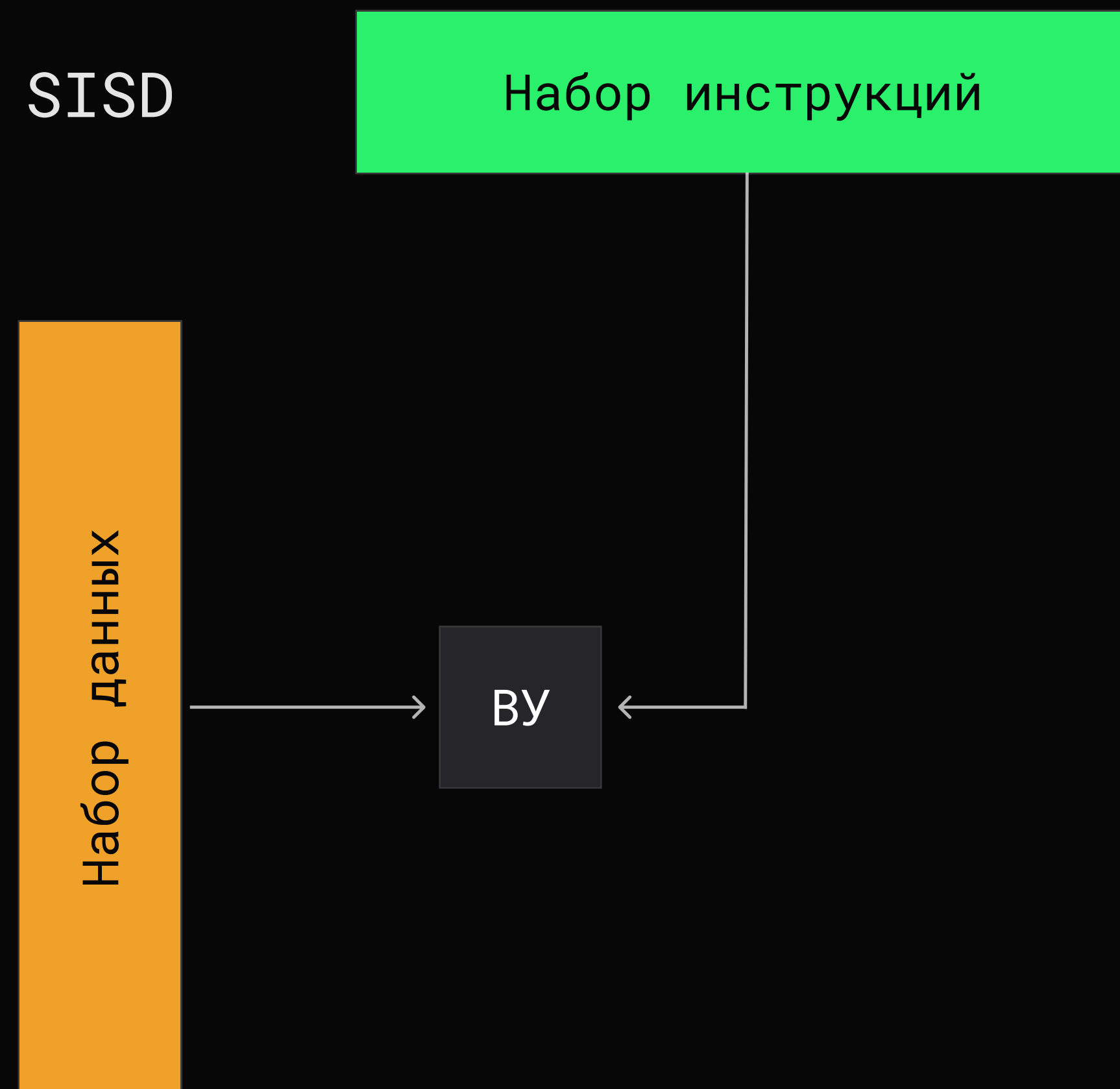
УСТРОЙСТВО РЕГИСТРОВ X86

Как понимать, какую инструкцию
исполнять в данный момент?

```
func main() {  
    a := 1      // 0x6e20dde2  
    b := 2      // 0x6e20bbe2  
    c := a + b  // 0x5a20ddd3  
}
```

PC
program counter
(instruction pointer)

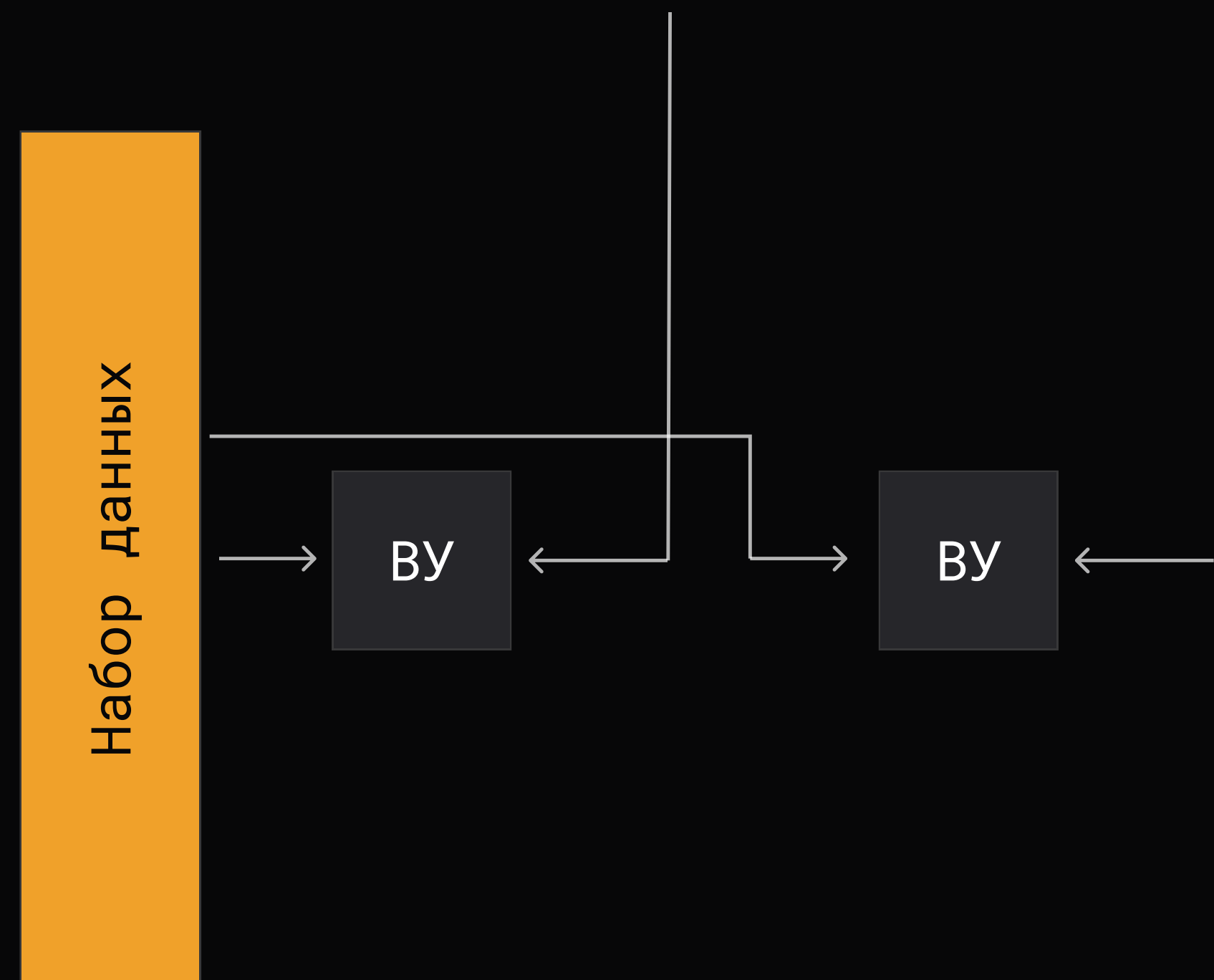
ТАКСОНОМИЯ ФЛИННА



ТАКСОНОМИЯ ФЛИННА

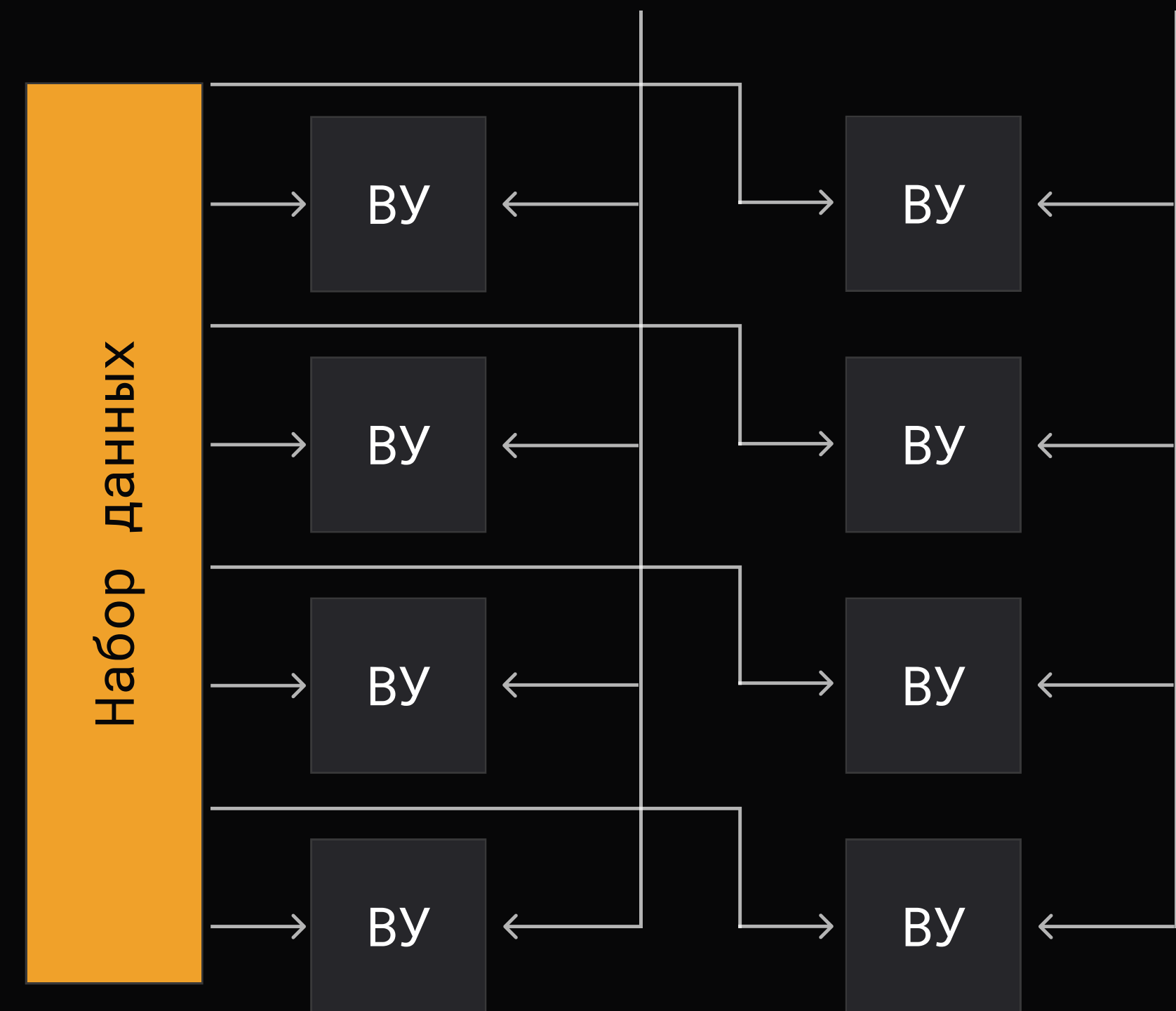
MISD

Набор инструкций



MIMD

Набор инструкций

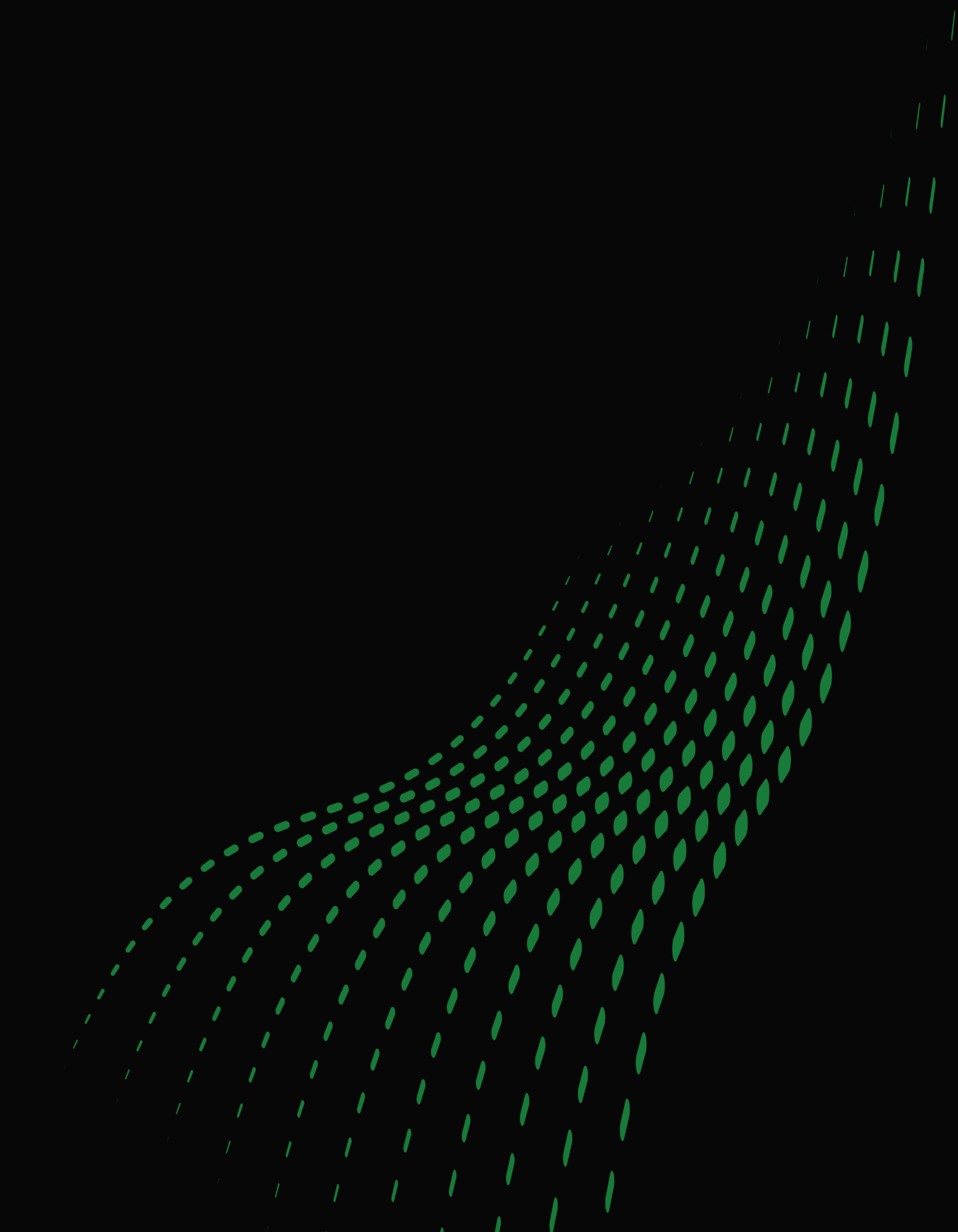


SIMD

Advanced Vector Extensions (Intel) до 256 бит

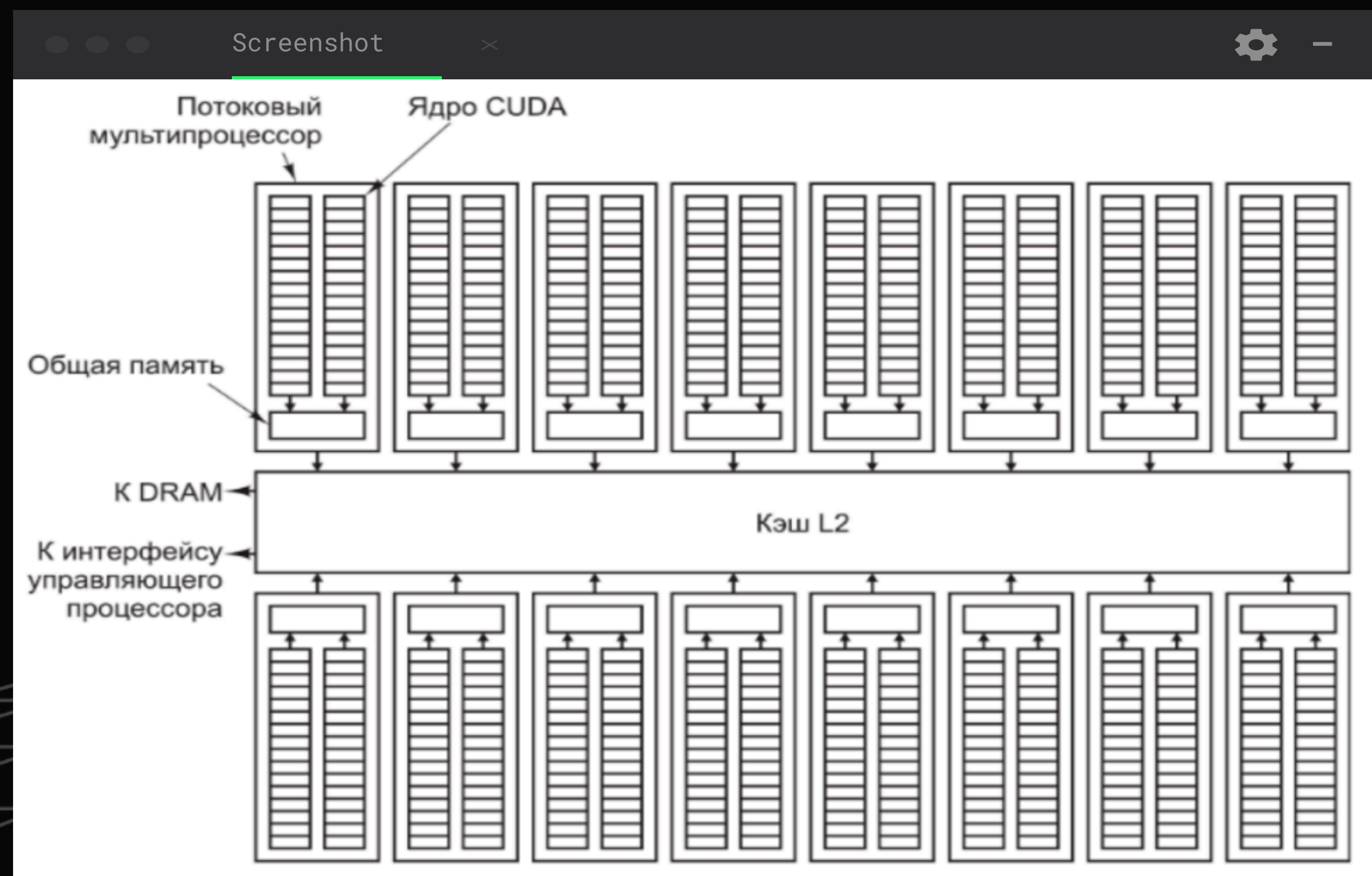
Intel MIC до 512 бит

NEON (ARM) до 128 бит

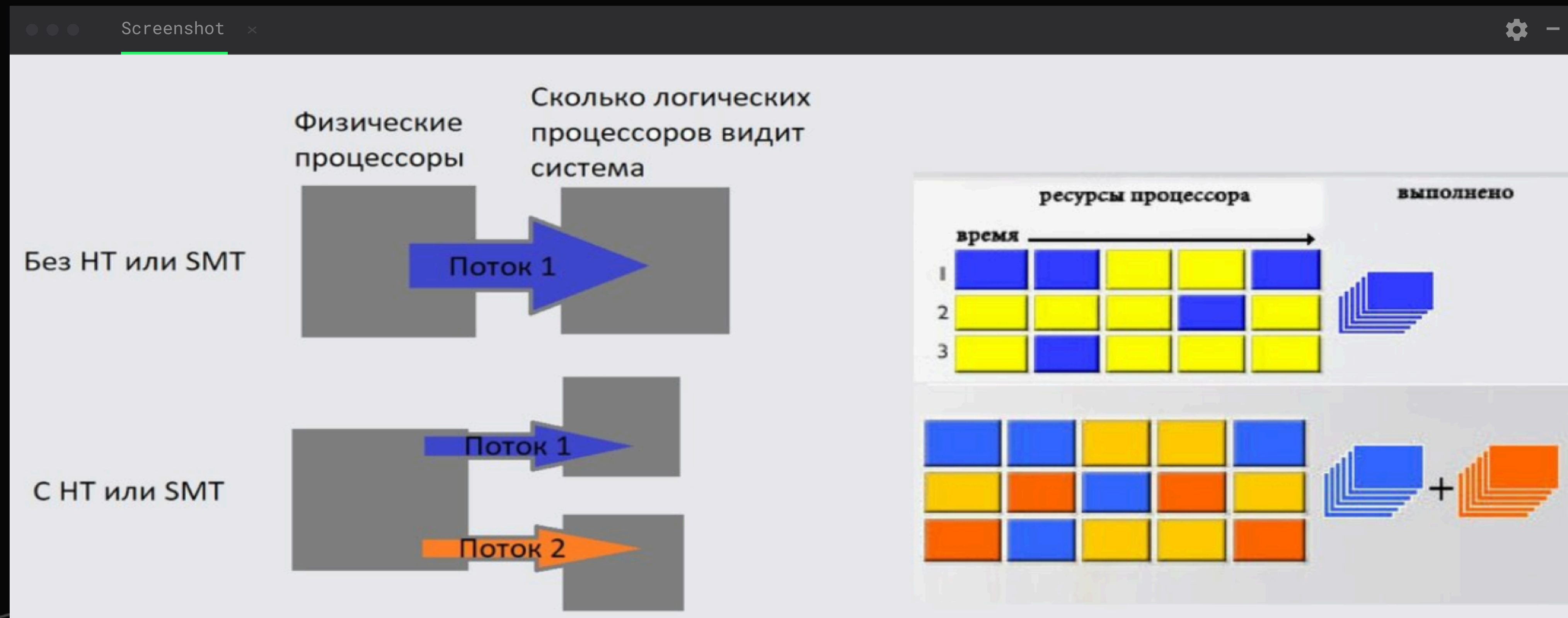


SIMD, GPGPU

GPU обычно основаны на SIMD-архитектуре с поддержкой векторов длины 128, 256 или более бит



SMT И HYPER-THREADING



ИТОГИ И ВЫВОДЫ

ЧТО БУДЕТ ДАЛЬШЕ