

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «РЫБИНСКИЙ
ГОСУДАРСТВЕННЫЙ АВИАЦИОННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
ИМЕНИ П.А. СОЛОВЬЕВА»

ИНСТИТУТ «ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ И СИСТЕМЫ УПРАВЛЕНИЯ»

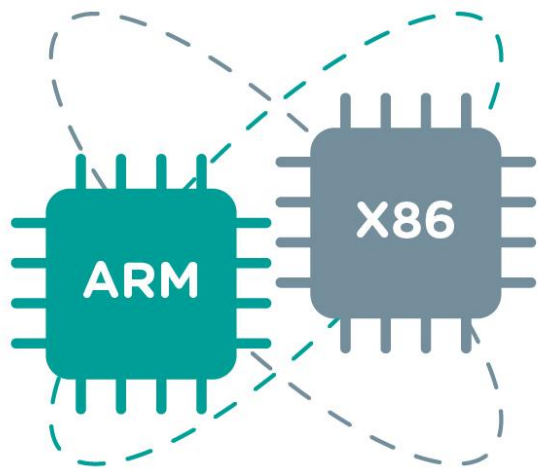
КАФЕДРА ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ

«АРХИТЕКТУРЫ ARM И x86»

Работа выполнена студентом группы ИВМ-24 Морозов А. А.

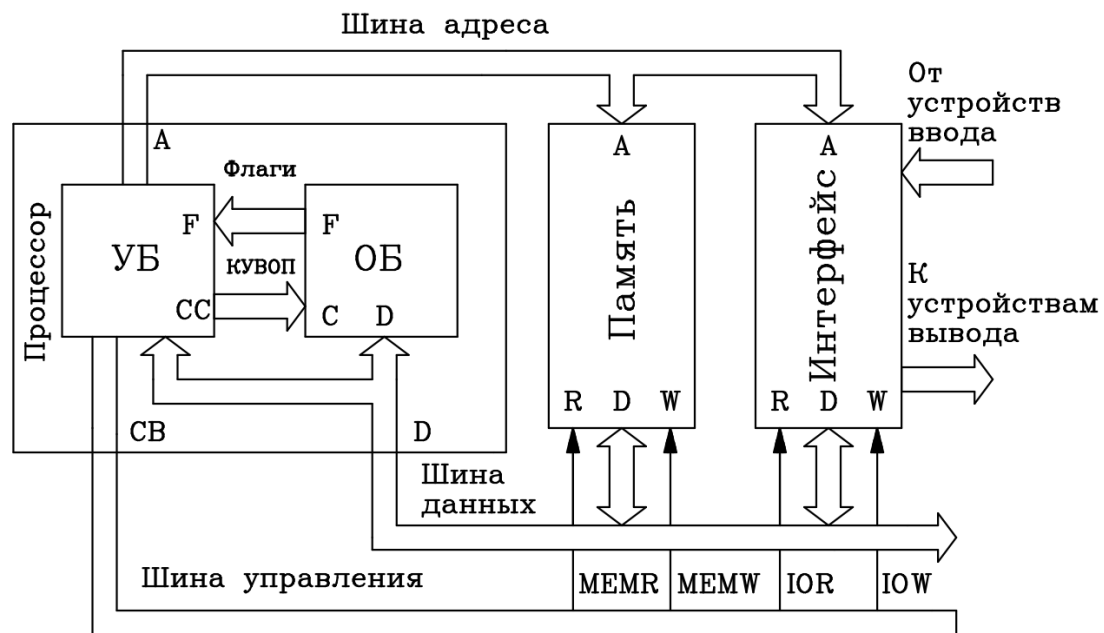
Преподаватель к. т. н., доцент

Павлов Р. В.



ARM И x86

- Современные вычислительные системы базируются на двух основных подходах к архитектуре процессоров – ARM и x86.



ОСНОВЫ РАБОТЫ МИКРОЭВМ

- Любая микропроцессорная система служит для обработки информации с целью получения требуемого результата



РАННИЕ ГОДЫ ЭВМ

- Первая электронная вычислительная машина (ЭВМ) ENIAC была создана в США в 1946 году для расчета баллистических таблиц. В ней использовалось 17468 электронных ламп и 1500 электромагнитных реле

Индиковый код "Минск-22"

Код перевод. эк. мил	Значение кода	Позиция на узкой ленте
	0	0
	1	1
	2	2
	3	3
	4	4
	5	5
	6	6
	7	7
	8	8
	9	9
	+ десятилит.	+
	- десятилит.	-
	+ десятилит.	+
	- десятилит.	-
	Затяжка	1, 6
	Десятилит. пробел	
	Передний адрес	
	Используется как передний адрес	
	Затяжка	
	Используется как затяжка	
	Пробел	
	Используется как пробел	
	Граница блока (вместе с нулем)	

МАШИННЫЙ КОД

- Машинный код — это низкоуровневый код, состоящий из двоичных инструкций, которые могут быть интерпретированы процессором и выполнены без дополнительных преобразований

Машинный код vs Ассемблер

Машинный код (побайтовый)

```
B8 22 11 00 FF
01 CA
31 F6
53
8B 5C 24 04
8D 34 48
39 C3
72 EB
C3
```

Код на Ассемблере

```
foo:
movl $0xFF001122, %eax
addl %ecx, %edx
xorl %esi, %esi
pushl %ebx
movl 4(%esp), %ebx
leal (%eax,%ecx,2), %esi
cmpl %eax, %ebx
jnae foo
retl
```

АССЕМБЛЕР

- Ассемблер был разработан как промежуточный шаг между машинным кодом и более высокоуровневыми языками программирования.

АРХИТЕКТУРА X86

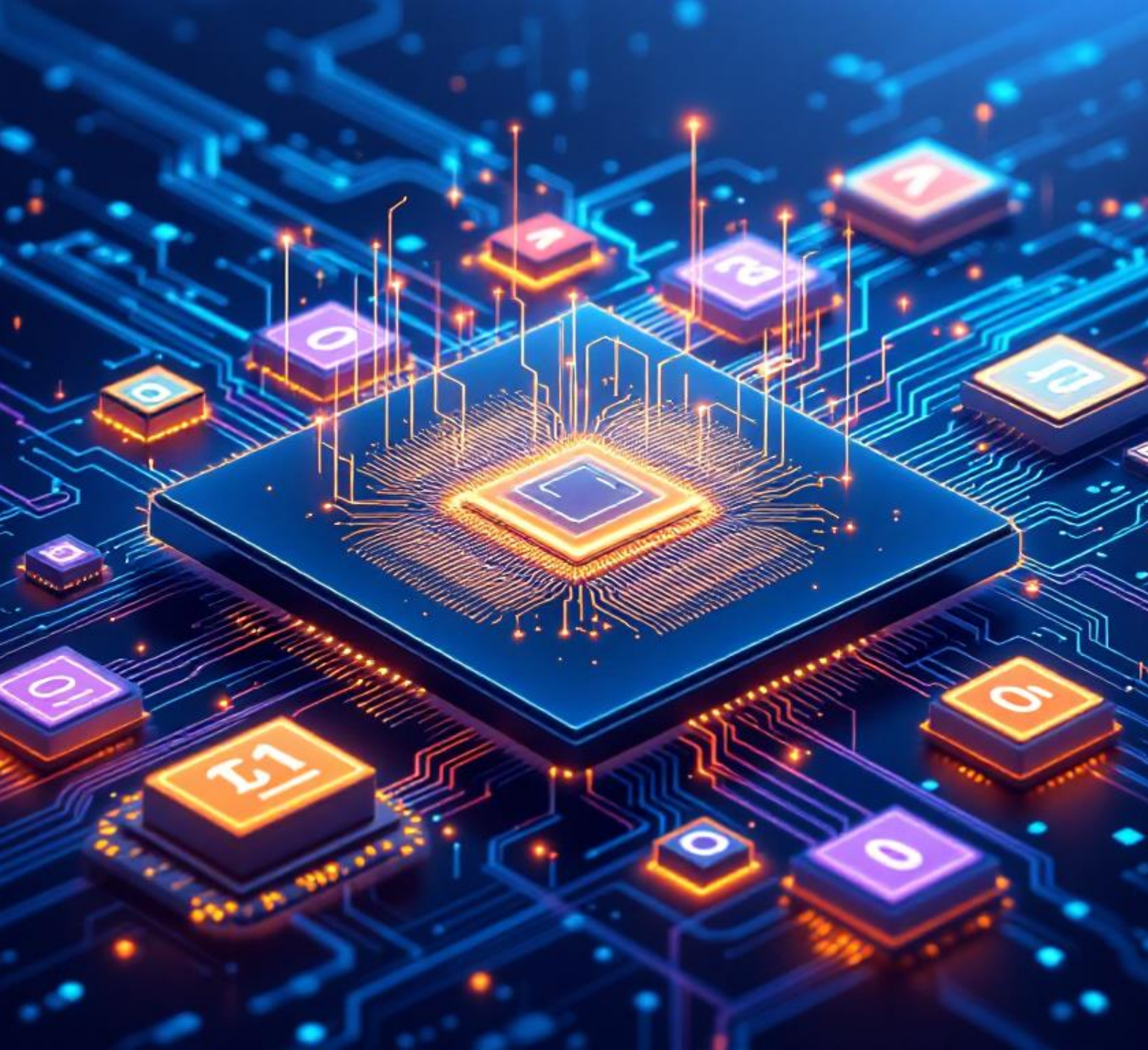


- Архитектура x86 была разработана компанией Intel в начале 1970-х годов. Первый процессор с архитектурой x86, *Intel 8086*, был выпущен в 1978 году, и с тех пор эта архитектура претерпела множество изменений и усовершенствований.

CISC АРХИТЕКТУРА КОМАНД

- Архитектура CISC (*Complex Instruction Set Computing*) относится к архитектуре процессора, в которой используется набор сложных инструкций, каждая из которых может выполнять несколько операций за один цикл.

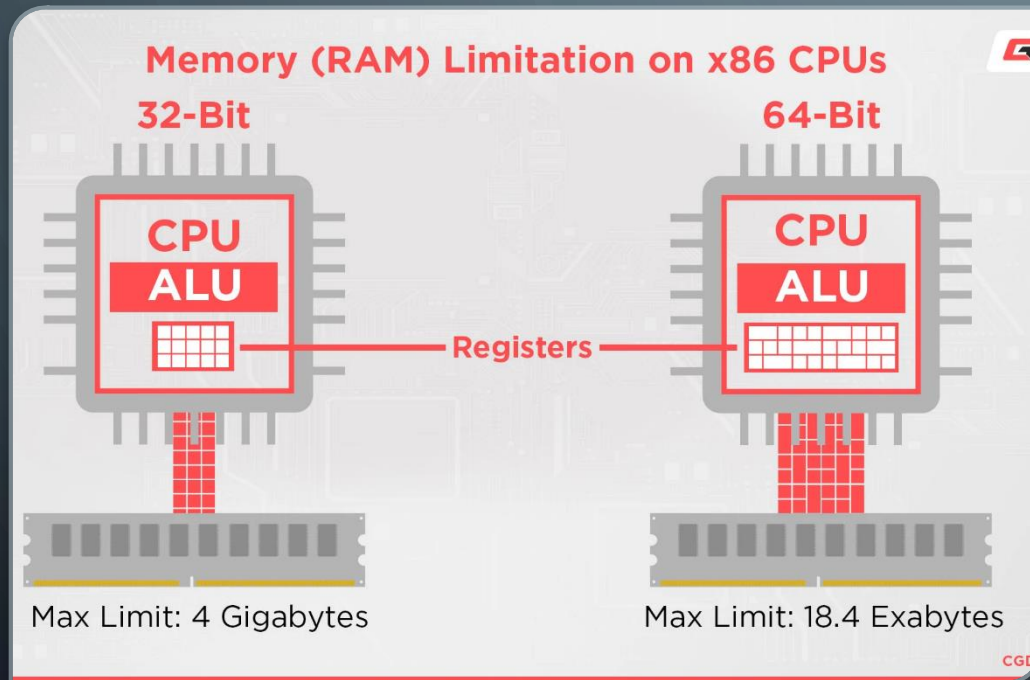




ОСОБЕННОСТИ *CISC*

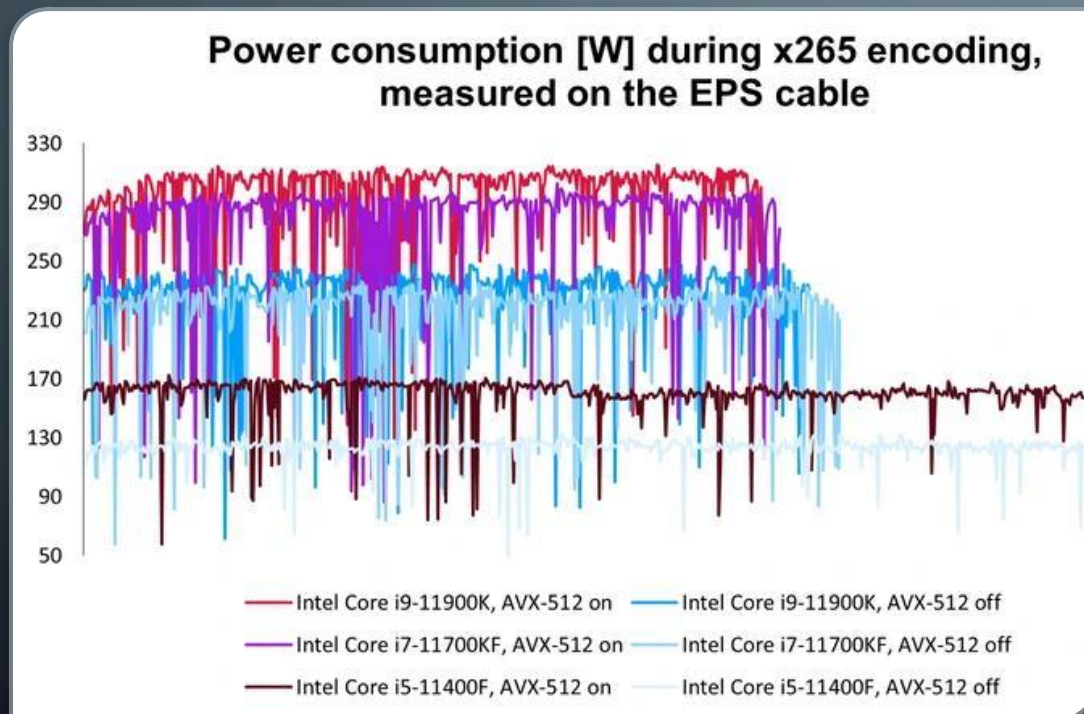
- К особенностям *CISC* относятся:
 - Использование переменной длины команд;
 - Работа напрямую с памятью;
 - Небольшое количество регистров.

СОВМЕСТИМОСТЬ С ПРЕДЫДУЩИМИ ПОКОЛЕНИЯМИ

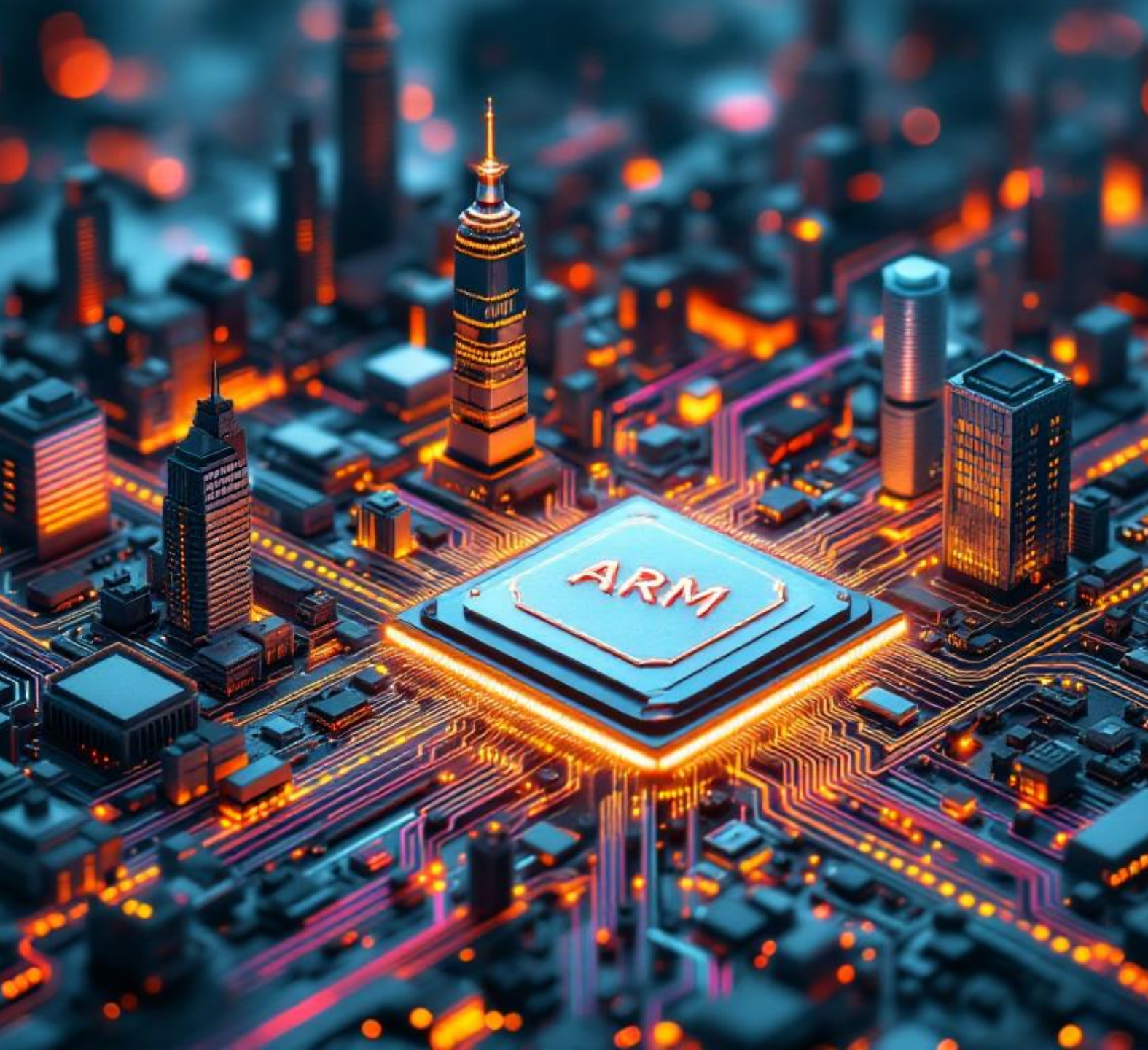


- С развитием архитектуры x86 появились новые расширения, такие как 32-битные и 64-битные режимы работы. Для обратной совместимости используются режимы *Long mode* и *Legacy mode*.

РАСШИРЕНИЯ ДЛЯ X86



- Архитектура x86 прошла через множество улучшений и расширений. Наиболее значимым из них оказались:
 - *MMX (MultiMedia Extensions);*
 - *SSE (Streaming SIMD Extensions);*
 - *AVX (Advanced Vector Extensions).*



АРХИТЕКТУРА ARM

- Разработана в 1980-х годах британской компанией *Acorn Computers* и была ориентирована на создание высокопроизводительных, энергоэффективных процессоров, способных удовлетворять потребности быстро развивающегося рынка.



RISC АРХИТЕКТУРА КОМАНД

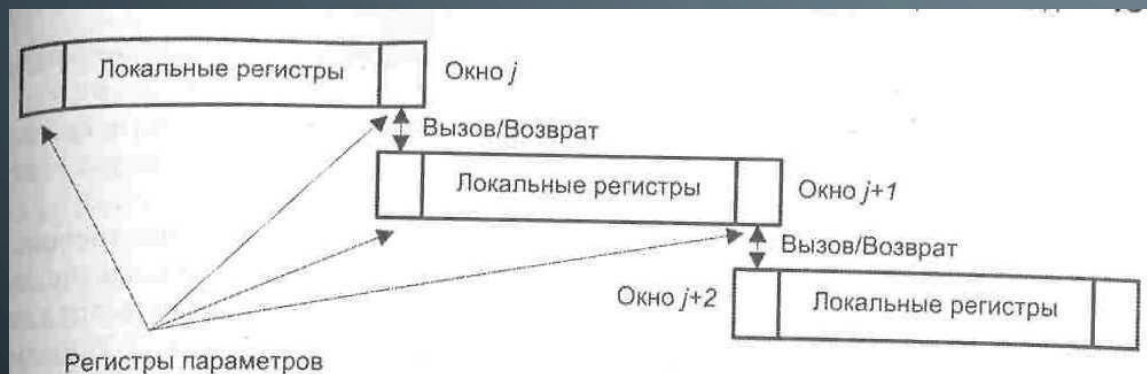
- *RISC (Reduced Instruction Set Computer)* – это архитектурная концепция, основная цель которой заключается в упрощении набора машинных команд для повышения производительности и уменьшения затрат на обработку инструкций.



ОСОБЕННОСТИ *RISC* АРХИТЕКТУРЫ

- К особенностям *RISC* относят:
 - Регистровые окна и файл;
 - Фиксированная длина команд;
 - Конвейер команд.

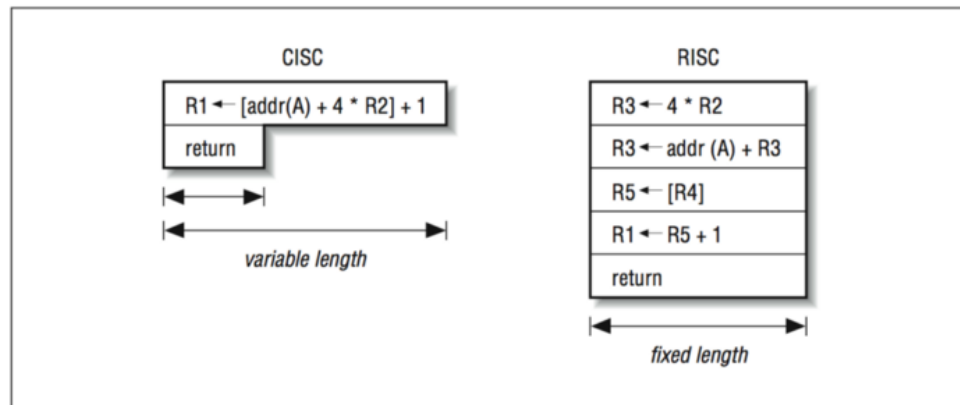
РЕГИСТРОВЫЕ ОКНА



- Регистровые окна состоят из 3 полей:
 - - левое поле для входных значений;
 - - среднее поле для локальных значений;
 - - правое поле для выходных значений.

ФИКСИРОВАННАЯ ДЛИНА КОМАНД

- Все команды в RISC имеют одинаковую длину, обычно 32 бита, что существенно упрощает их обработку процессором (декодирование).

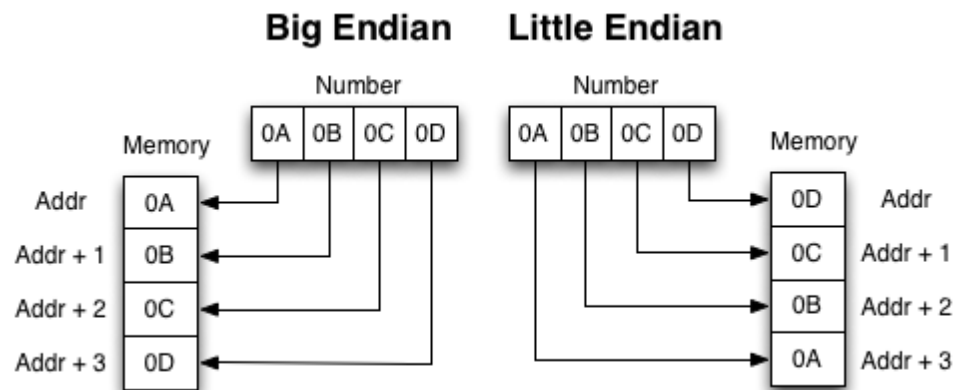


КОНВЕЙЕР КОМАНД

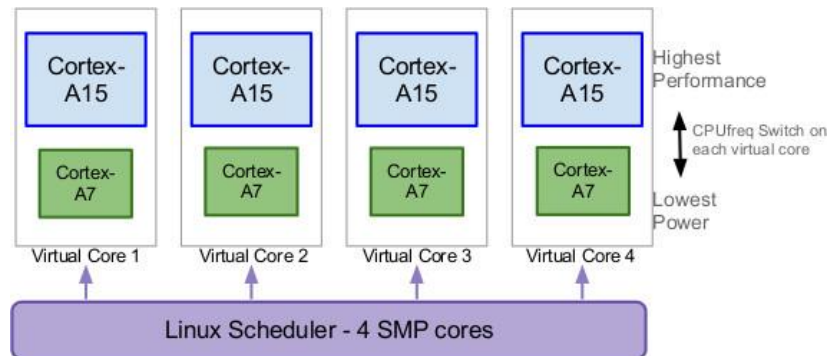
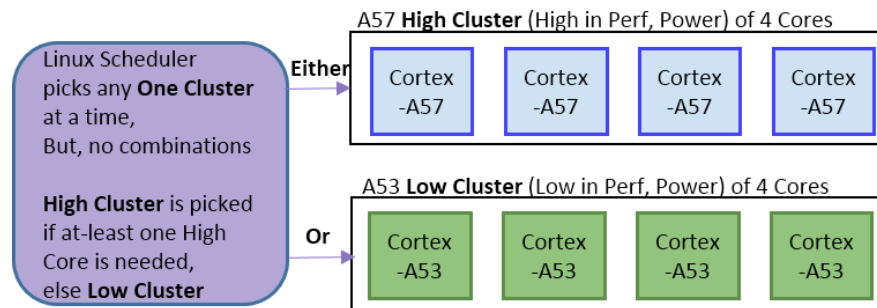
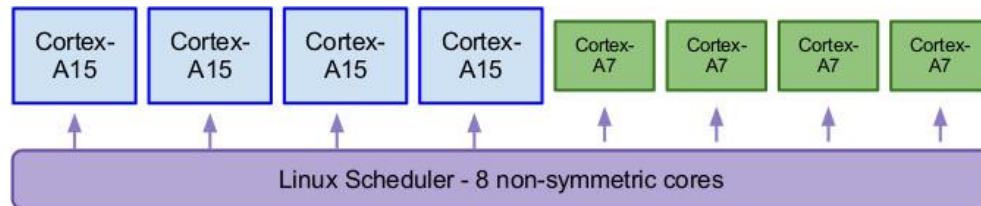


- Конвейер позволяет выполнять процессору сразу несколько команд, значительно увеличивая его производительность. В *RISC* архитектуре конвейер выполняет намного проще и появился раньше чем в *CISC*.

ПОРЯДОК БАЙТОВ



- В ARM реализована поддержка сразу обоих порядков байтов, а в x86 только *little-endian* (младший байт по младшему адресу).



ARM BIG.LITTLE

- 2 типа ядер:
 - Большие ядра;
 - Маленькие ядра.
- 3 способа управления ядрами:
 - *Heterogeneous Multi-Processing;*
 - *Clustered switching;*
 - *In-kernel switcher.*

The background is a dark blue gradient. In the corners, there are decorative white lines resembling circuit traces or a stylized city skyline. These lines connect to small white circles, some of which are arranged in a grid-like pattern.

СПАСИБО ЗА ВНИМАНИЕ!

Презентацию подготовил студент группы ИВМ-24 Морозов А. А.