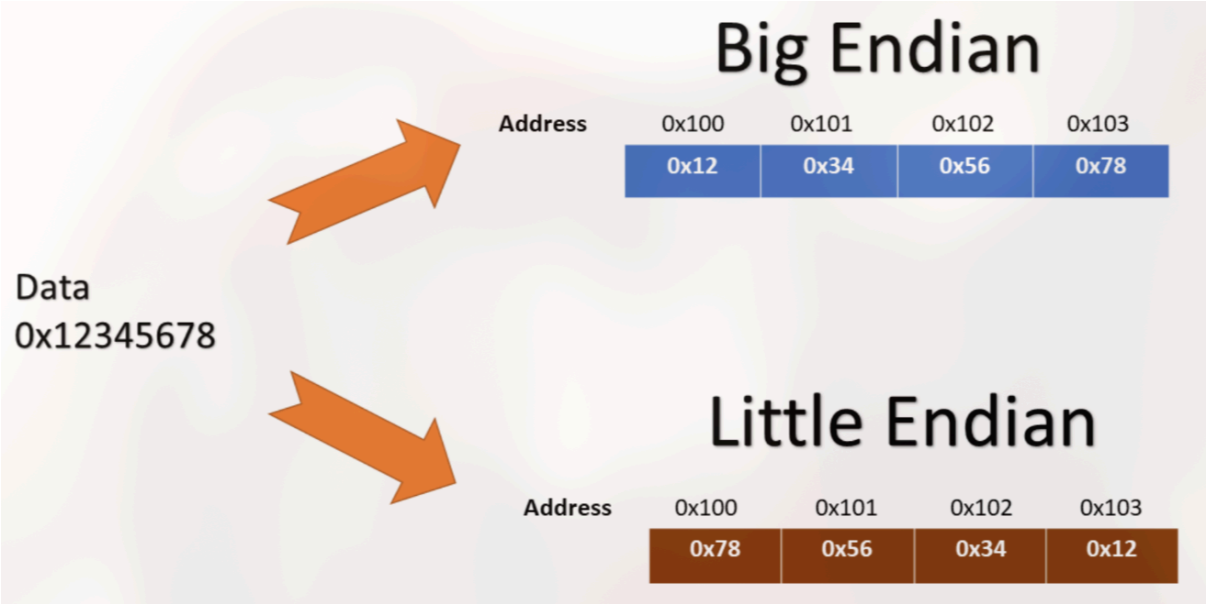
ОПД ответы на вопросы с экзамена

1. IP - сетевой уровень. Нужен для передачи информации между разными устройствами в сети. UDP - транспортный уровень. Нужен для передачи данных между процессами хоста, добавляет к ip порт и контрольную сумму.
   Оба протокола не гарантируют соединения, но UDP добавляет проверку целостности через контрольную сумму.
2. Big Endian: Самый значимый байт хранится по наименьшему адресу памяти (Сетевой порядок байтов (*Network Byte Order*, именно его используют протоколы TCP/IP), применяется в старых RISC-архитектурах).

   Little Endian: Самый младший байт хранится по наименьшему адресу памяти (x86/x86-64 (Intel, AMD), ARM)



3. CISC (*Complex Instruction Set Computer*) - сложные инструкции переменной длины (1-15 байт) - их еще называют микропрограммы (набор микроинструкций), и хранятся они в ROM-памяти. ROM-память - это либо «впаянная» память на самом кристалле, либо загрузка с материнской платы в SRAM процессора при запуске системы (здесь SRAM - не кэши, а отдельная область). Из-за множества режимов адресации и длины команд мало (8-16) регистров. В CISC сделали аналог конвейеризации - разбиение инструкций на микрооперации, а также гипертрединг: процессор берет несколько потоков инструкций, каждый поток разделяется на части и конвертируется в микрооперации.

   RISC (*Reduced Instruction Set Computer*) - простые инструкции одинаковой длины (4 байта). Инструкции в RISC имею немного режимов адресации, а потому для 32-битной команды есть больше бит для указания регистра -> получаем много (16-32) регистров. Архитектура основана на принципе Load/Store - четкого разделения работы с памятью и арифметических операций,

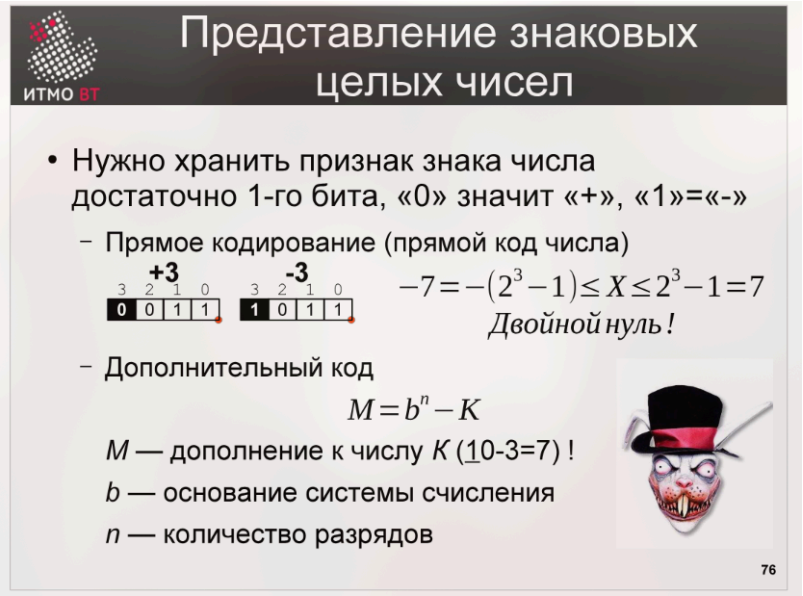
то есть арифметика выполняется только с регистрами. Благодаря вышеперечисленному RISC допускает конвейеризацию

.

RISC также допускает сжатые инструкции, 16-битные команды вместо стандартных 32-битных для повышения плотности кода и уменьшения объема памяти. В RISC-V Работают по принципу декодирования в соответствующие 32-битные, проверяются через 2 младших бита (= 11).
RISC имеет большие кэши.

Еще из интересного, CISC лицензировано, а RISC - открытая архитектура.

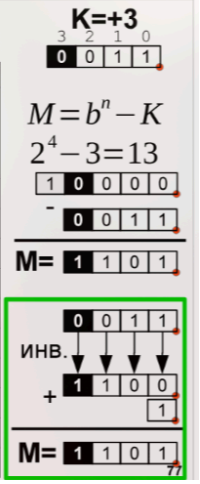
4. Представление беззнаковых чисел в ЭВМ
    1) прямой код: знак в самом старшем бите. Есть двойной нуль (-0, +0). Для вычислений обычно не используется, он делает арифметические операции неэффективными и сложными для реализации.
    2) обратный код: представляет отрицательные числа, дополняя до максимально возможного положительного числа в разрядной сетке + 1.
на слайде: b - CC, n - размер разрядной сетки, K - модуль отриц. числа



Представление знаковых чисел: дополнительный код

$$M = b^n - K = ((b^n - 1) - K) + 1$$

Прямой код 5-ти разр. дес. чисел	Дополнительный код		
	5-ти разр. дес. чисел	4-х разр. шестн. чисел	16-ти разрядных двоичных чисел
-50000	50000		
-49999	50001		
-32768	**67232**	**8000**	**1000 0000 0000 0000**
-32767	67233	8001	1000 0000 0000 0001
-2	99998	FFFE	1111 1111 1111 1110
-1	99999	FFFF	1111 1111 1111 1111
0	**00000**	**0000**	**0000 0000 0000 0000**
1	00001	0001	0000 0000 0000 0001
32767	32767	7FFF	0111 1111 1111 1111
49999	49999		



77

Получение дополнительного кода БЭВМ

Адрес	Содержимое		Комментарии
	Код	Мнемоника	
010	0200	**CLA**	
011	4016	**ADD 16**	X в аккумуляторе (2)
012	0280	**NOT**	Вычисление дополнения (инверсия битов - FFFD)
013	0700	**INC**	Инкремент (FFFE)
014	E017	**ST 17**	Сохранение результата
015	0100	**HLT**	
016	0002	X	X
017	FFFE	R	-X

Да, я знаю, все это можно было сделать проще!

-32768	**8000**	**1000 0000 0000 0000**
-32767	8001	1000 0000 0000 0001
-2	FFFE	1111 1111 1111 1110
-1	FFFF	1111 1111 1111 1111
0	**0000**	**0000 0000 0000 0000**
1	0001	0000 0000 0000 0001
32767	7FFF	0111 1111 1111 1111

78