



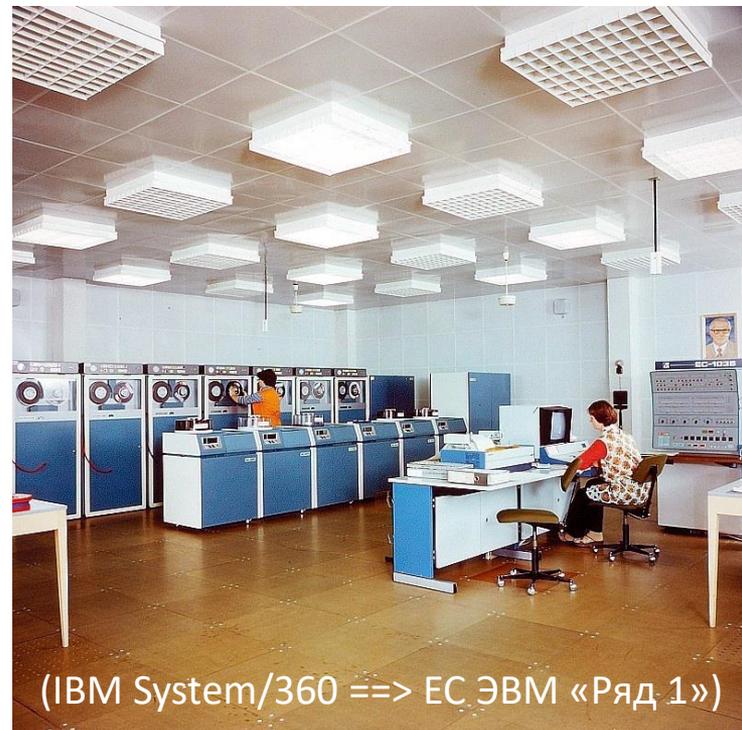
Е.А. Еремин (Пермь)

Возвращение раздела «Основы ВТ» в общеобразовательный курс информатики ради развития собственного производства в этой области

15-16 мая 2025 года, г. Омск

Чуть-чуть истории

(или *отстаем, когда начинаем копировать*)



(IBM System/360 ==> ЕС ЭВМ «Ряд 1»)



1985 г. – появление предмета ОИВТ



ОИВТ = **о**сновы **и**нформатики и **в**ычислительной **т**ехники



Изучение основ устройства ЭВМ – давняя традиция курса информатики. Предполагалось, что для грамотного использования ЭВМ надо в общих чертах понимать, как она работает.





Изучение велось на примере:

- либо реальной ЭВМ
- либо ее учебной модели («Кроха», «Малютка», «Нейман» и др.)

18.3. ПРИМЕРЫ КОМАНД ПРОЦЕССОРА

Поясним принцип работы процессора на примере нескольких команд школьной ЭВМ "Электроника УКНЦ". Каждая команда занимает в памяти 1, 2 или 3 слова (словом называется пара байт с адресами 0—1, 2—3, 4—5 и т. д.). Целое число занимает в памяти ЭВМ также 1 слово.

Команда	Длина в словах	Что происходит при выполнении команды
стоп	1	ЭВМ кончает работу
очистить x	2	$x := 0$
увеличить x на единицу	2	$x := x + 1$
уменьшить x на единицу	2	$x := x - 1$
переслать из x в y	3	$y := x$
прибавить x к y	3	$y := y + x$
вычесть x из y	3	$y := y - x$
сравнить x с y	3	$S := \text{sign}(x - y)$
сравнить x с нулем	2	$S := \text{sign}(x)$
если меньше переход на d	1	проверяется знак S в ССП. Если указанное в команде условие выполнено, то $CK := CK + 2 \cdot d$ иначе CK не меняется
если равно переход на d	1	
если больше переход на d	1	
если не меньше переход на d	1	
если не равно переход на d	1	
если не больше переход на d	1	

ПАМЯТЬ										ЭКРАН					
Адрес	Код операц.	A1	A2	A3	Двоичн.	Десят.									
000	0 0 0 0 0 0 0 0 0 0 0 0 0 0														
001	0 0 0 0 0 0 0 0 0 0 0 0 0 0														
010	0 0 0 0 0 0 0 0 0 0 0 0 0 0														
011	0 0 0 0 0 0 0 0 0 0 0 0 0 0														
100	0 0 0 0 0 0 0 0 0 0 0 0 0 0														
101	0 0 0 0 0 0 0 0 0 0 0 0 0 0														
110	0 0 0 0 0 0 0 0 0 0 0 0 0 0														
111	0 0 0 0 0 0 0 0 0 0 0 0 0 0														
Счетчик команд : 000															
Регистр команд : 000 000 000 000										0	- десятичное значение				
Регистр сумматора : 000000000000										0	- десятичное значение				



КУВТ отечественного производства



КУВТ = комплект **у**чебной **в**ычислительной **т**ехники

Изначально планировалось создать для школы свою учебную ВТ.

Вот как выглядели наиболее распространенные КУВТ:

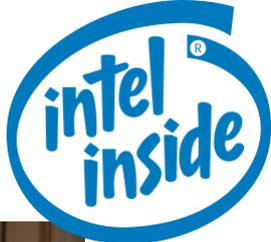
1. «УКНЦ»
2. «Корвет»
3. «Yamaha MSX2» ("образец")

(фото из личного архива автора)

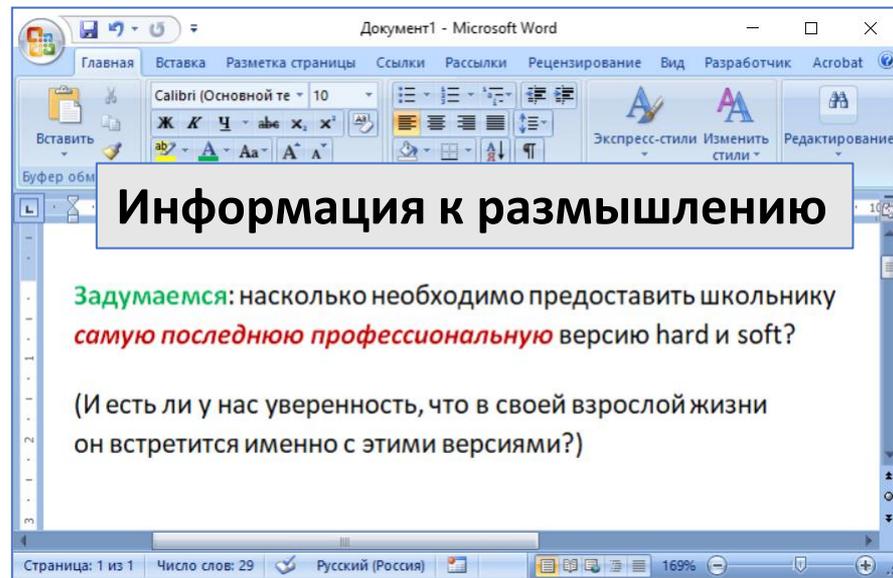




Далее пришла эпоха



А с ней назойливый вопрос:
зачем нам знать, как устроен процессор?
Пусть Intel® об этом думает!





(Без комментариев)



intel

 ENGLISH

 Search Intel.com

Коммуникации

Мы немедленно приостановили все деловые операции в России. Это следует за нашим предыдущим решением приостановить все поставки клиентам в России и Беларуси.

[Company Overview](#) | [Contact Intel](#) | [Newsroom](#) | [Investors](#) | [Careers](#) | [Corporate Responsibility](#) | [Inclusion](#) | [Public Policy](#)



intel

[© Intel Corporation](#) | [Terms of Use](#) | [*Trademarks](#) | [Cookies](#) | [Privacy](#) | [Supply Chain Transparency](#) | [Site Map](#) | [Recycling](#)

Intel technologies may require enabled hardware, software or service activation. // No product or component can be absolutely secure. // Your costs and results may vary. // Performance varies by use, configuration, and other factors. Learn more at [intel.com/performance/index](https://www.intel.com/performance/index). // See our complete legal [Notices and Disclaimers](#). // Intel is committed to respecting human rights and avoiding causing or contributing to adverse impacts on human rights. See Intel's [Global Human Rights Principles](#). Intel's products and software are intended only to be used in applications that do not cause or contribute to adverse impacts on human rights.



Требуются отечественные процессоры!



Важные выводы

Производство невозможно без специалистов.

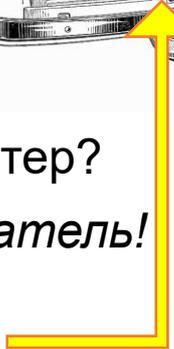
В школе надо рассказать об этих специальностях (профориентация).

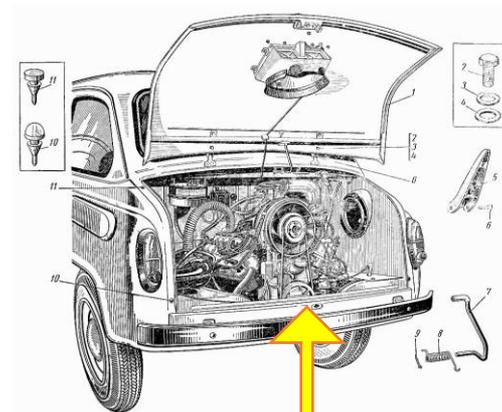


(Т.к. времени мало, детали не обсуждаем.)

Зачем изучать ОВТ?

И действительно, зачем школьнику знать, как работает компьютер?
Можем же мы водить автомобиль, не зная, как устроен двигатель!

(И все же полезно знать,  с *какой стороны* подходить к технологиям!)





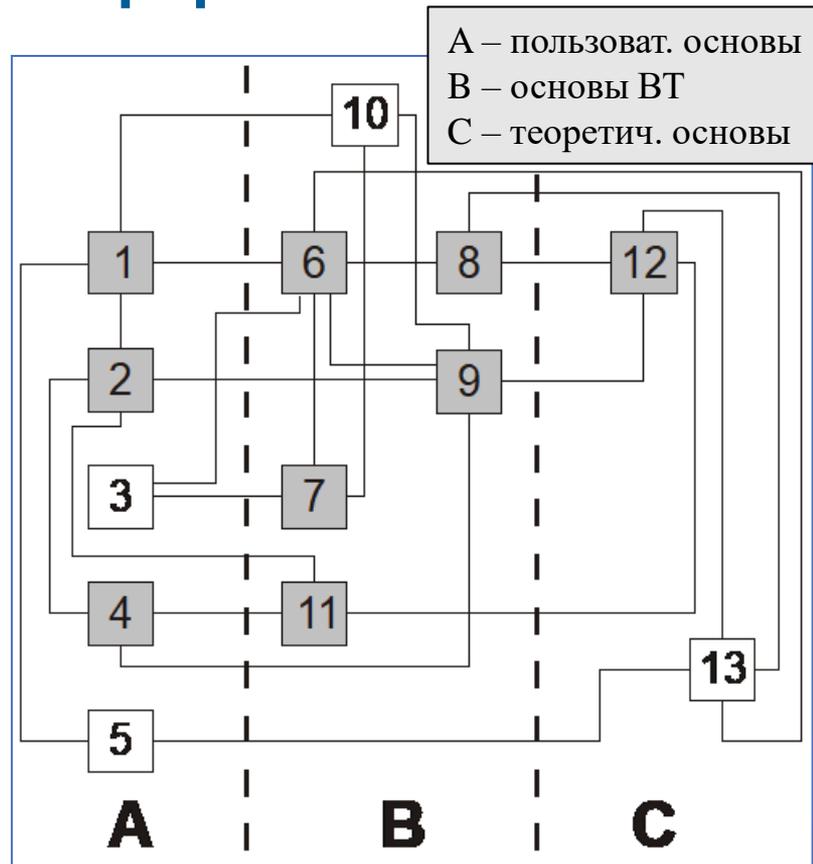
Роль изучения ОВТ в курсе информатики



1	виды ПО [в т.ч. сетевые], пользовательский интерфейс
2	понятие о данных и носителях
3	конструкция компьютера, соединение устройств и характеристики
4	разрядность, слово, бит, байт, единицы измерения
5	история ЭВМ, поколения (в т.ч. элементная база)
6	базовые принципы, исполнение программы, функциональные блоки
7	взаимодействие блоков, шины и интерфейсы, протоколы [в т.ч. сети]
8	процессор, его характеристики, система команд и регистры
9	устройство внешней и внутренней памяти
10	управление ОС аппаратной частью, файловая система, загрузка
11	представление данных
12	теория: системы счисления, дискретность, кодирование, алгебра логики, двоичная арифметика
13	логические элементы, схемы (напр., триггер) и узлы (напр., регистр)

Еремин Е.А. Анализ содержательной линии "Компьютер" курса информатики с применением компьютерных средств представления знаний.

Информатика ("Первое сентября"), 2008, N 9, с.8-18.





Поясняющий пример

Без рассказа о сумматоре и АЛУ польза от изучения теории сложения и прочих операций двоичной арифметики (напр. умножение = сложение + сдвиг) неочевидна.

Информация к размышлению

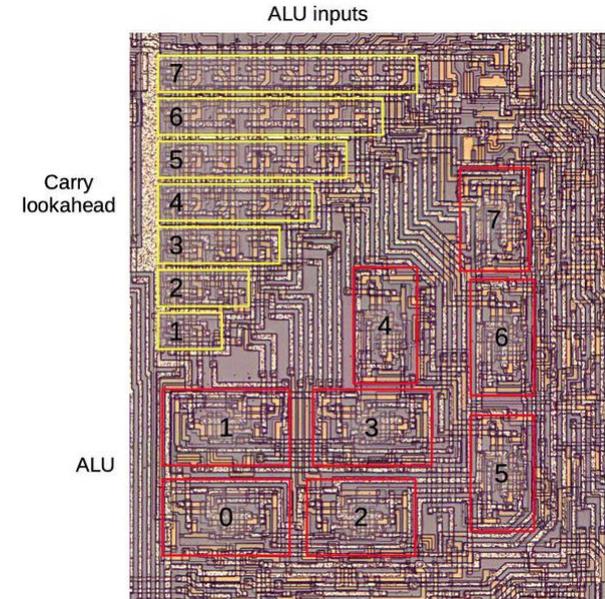
Сложение в математике из-за наличия переноса выполняется **поразрядно**.

В литературе по ВТ утверждается, что разряды чисел суммируются **одновременно**.

???

Ключевое слово для поиска: **ускоренный перенос**.

Увеличенный сегмент кристалла Intel 8008 со схемой ускоренного переноса и АЛУ
<https://habr.com/ru/companies/rvuds/articles/528026/>
(перевод с <https://www.righto.com/2020/11/reverse-engineering-carry-lookahead.html>)



Примеры



(Где в курсе информатики уместно поговорить об основах ВТ?)



Акцент 1. Байт – особая единица



Байт – это единица, **непосредственно связанная с аппаратным устройством**.

Каждый байт памяти имеет адрес и является **минимальной** порцией данных, которую можно прочитать; **читать менее 1 байта невозможно**. Зато **можно читать сразу несколько байт**: 1, 2, 4 или 8.

Специфические (но важные!) вопросы, связанные с многобайтовыми данными:

1. выравнивание (alignment);
2. порядок байтов в ОЗУ (big/little endian);
3. преобразование данных разной длины, в т.ч. расширение знака (sign extension).

Где это полезно:

- (1) True color RGB – 3 байта, а из ОЗУ читается по 4;
- (1) оптимизация структур из данных разной длины;
- (2) сохранение изображения в Photoshop с порядком байтов Windows и Mac OS;
- (3) расширяющие (widening) и сужающие (narrowing) преобразования типов данных в языках;
- (3) без расширения знака трудно понять арифметику в дополнительном коде, а также signed и unsigned типы; **и много еще где!**



4

Компьютерная арифметика



Рис. 4.2

Наличие **переполнения** —
главное отличие
арифметики в ВТ
от математики. !

Вывести на такой индикатор число 10 000 невозможно: не хватает технического устройства для пятого разряда. Такая «аварийная» ситуация называется переполнением разрядной сетки или просто переполнением (англ. *overflow* — переполнение «сверху»).

Переполнение разрядной сетки — это ситуация, когда число, которое требуется сохранить, не умещается в имеющемся количестве разрядов вычислительного устройства.

В нашем примере переполнение возникает при значениях, больших $9999 = 10^4 - 1$, где 4 — это количество разрядов. В общем случае, если в системе счисления с основанием B для записи числа используется K разрядов, максимальное допустимое число C_{\max} вычисляется по аналогичной формуле¹

$$C_{\max} = B^K - 1.$$

Именно эта формула для $B=2$ неоднократно применялась в главе 2.

Подчеркнём, что переполнение никак не связано с системой счисления: оно вызвано *ограниченным количеством разрядов* устройства и не зависит от количества возможных значений в каждом из этих разрядов.

Что важно?

- Причина — ограниченное число разрядов.
- Знать, как реагирует машина (а часто — *никак!*)
- В идеале: надо **анализировать** возможность переполнения в задаче.
- Существует **программная** реализация арифметики (она снимает ограничения на разрядность, но медленная).



Акцент 2. Переполнение



4

Компьютерная арифметика



Рис. 4.2

Наличие **переполнения** —
главное отличие
арифметики в ВТ
от математики. 

Вывести на такой индикатор число 10 000 невозможно: не хватает технического устройства для пятого разряда. Такая «аварийная» ситуация называется переполнением разрядной сетки или просто переполнением (англ. *overflow* — переполнение «сверху»).

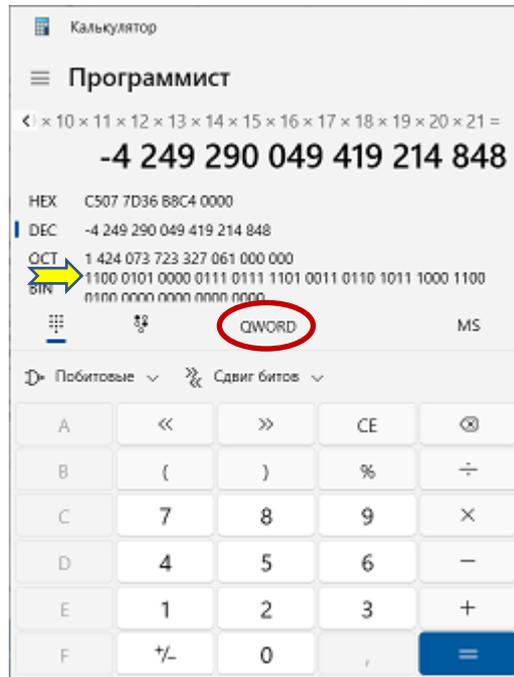
Переполнение разрядной сетки — это ситуация, когда число, которое требуется сохранить, не умещается в имеющемся количестве разрядов вычислительного устройства.

В нашем примере переполнение возникает при значениях, больших $9999 = 10^4 - 1$, где 4 — это количество разрядов. В общем случае, если в системе счисления с основанием B для записи числа используется K разрядов, максимальное допустимое число C_{\max} вычисляется по аналогичной формуле¹

$$C_{\max} = B^K - 1.$$

Именно эта формула для $B=2$ неоднократно применялась в главе 2.

Подчеркнём, что переполнение никак не связано с системой счисления: оно вызвано *ограниченным количеством разрядов* устройства и не зависит от количества возможных значений в каждом из этих разрядов.



Где можно увидеть?
64 бита (QWORD): $21! < 0$



Доп. 1. Языки описания аппаратуры (HDL)

NB

Описание на языке *SystemVerilog*

Пример: полный 1-битный сумматор

```
module fulladder(input logic a, b, cin,  
                output logic s, cout);
```

```
    logic p, g;
```

```
    assign p = a ^ b;
```

```
    assign g = a & b;
```

```
    assign s = p ^ cin;
```

```
    assign cout = g |(p & cin);
```

```
endmodule
```

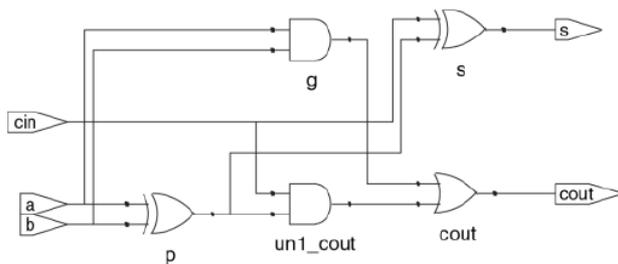


Рис. 4.8 Синтезированная схема модуля fulladder

К вопросу об **ускорении переноса** при сложении:

p – сквозной перенос

g – генерация переноса

Харрис Д.М., Харрис С.Л.

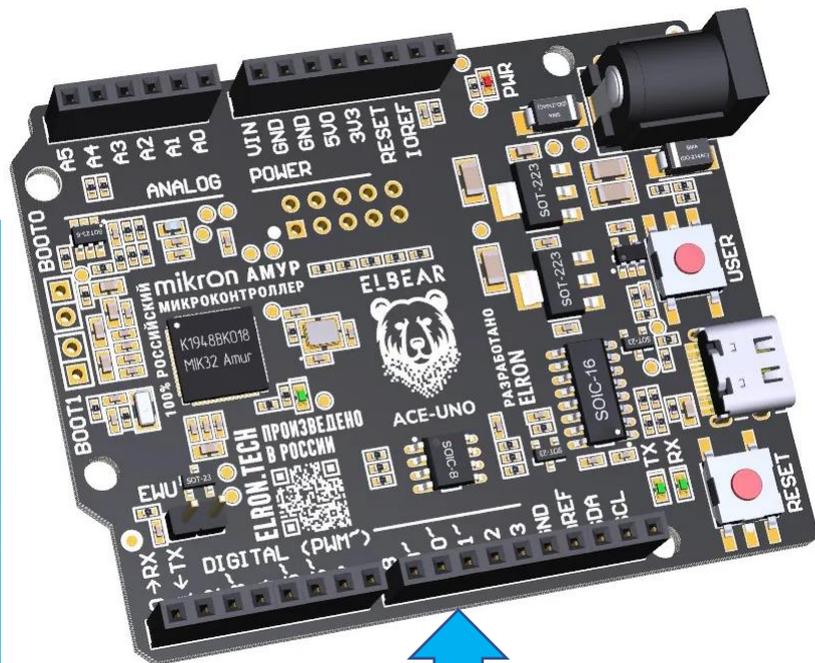
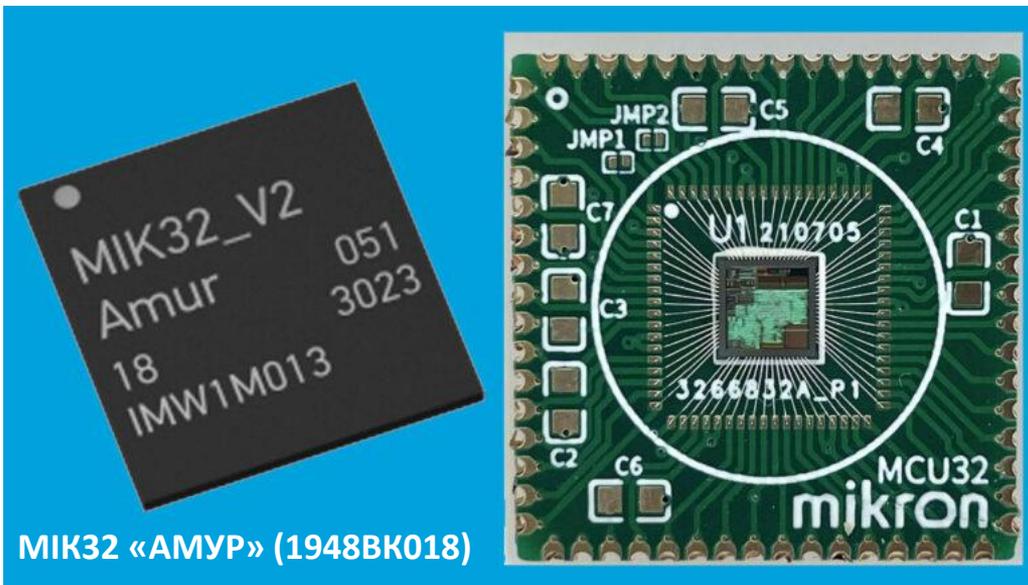
Цифровая схемотехника и архитектура компьютера: RISC-V.

М.: ДМК Пресс, 2021. 810 с. (см. стр. 234)

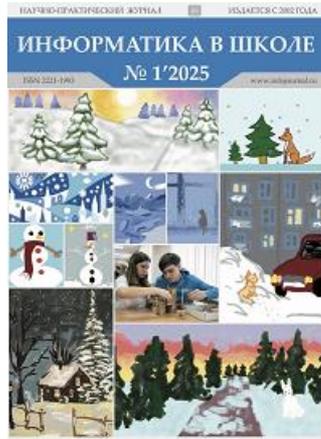


Доп. 2. Архитектура RISC-V

Большое распространение получает архитектура **RISC-V** – простая и очень удобная для обучения.



Arduino-совместимая плата
ELBEAR ACE-UNO



Информатика в школе
2025, N 1, с.53-63.

МЕТОДИЧЕСКАЯ КОПИЛКА TEACHING PRACTICES



DOI: 10.32517/2221-1993-2025-24-1-53-63



Е. А. Еремин

Пермский государственный гуманитарно-педагогический университет, г. Пермь, Россия

МНОГОСЛОЙНАЯ МОДЕЛЬ КОМПЬЮТЕРА КАК ОРИЕНТИР ПРИ ОТБОРЕ МАТЕРИАЛА ДЛЯ КУРСА ИНФОРМАТИКИ

Аннотация

Многослойная модель компьютера — это представление его в виде последовательности слоев (уровней): от физического кодирования значений 0 и 1 до прикладного программного обеспечения. Для формирования целостной картины знаний каждый слой имеет определенное значение. В статье традиционно выделяемые слои модели сопоставляются с содержанием сложившегося в школе курса информатики. Показано, что в программе углубленного курса X—XI классов можно найти материал, в той или иной степени связанный с каждым из слоев. Тем не менее в свете возникшей острой необходимости в вычислительной технике отечественного производства и ее элементной базе, удельный вес изучения некоторых слоев следует увеличить. На основе проведенного анализа федеральных рабочих программ школьного курса информатики приводятся конкретные рекомендации по возможному совершенствованию содержания предмета. Статья будет полезна учителям информатики, а также школьникам, интересующимся компьютерами и желающим связать с ними будущую специальность.



Сайт «**Учебные модели компьютера**» (emc.orgfree.com – кажется, это китайский хостинг).

Как автор методических материалов при их размещении я ощущаю необходимость в сетевой ***стабильно существующей некоммерческой образовательной*** площадке. Без рефератов и ГДЗ, без всякой посторонней рекламы, где главный акцент – на содержании материала, а не на обсуждении вокруг него (как это часто бывает в соцсетях).

Нужна ли российскому образовательному сообществу такая площадка или вдруг она уже есть?



Выводы

1. Обстановка такова, что стране необходимо выпускать **собственные микропроцессоры**. Для этого требуются **специалисты**.
2. И здесь **курс школьной информатики может помочь**. Не столько в подготовке будущих специалистов, сколько **в профориентации** школьников в эту область.
3. Технологии это замечательно, но кто-то должен их обеспечить **аппаратной основой!** Важно сказать ученикам, что это **не менее важно**, чем собственно технологии.
4. Еще важно показать, что разработка новых микросхем **не менее интересна**, чем написание новых программ. Тем более, что процессы похожи (вспомним HDL).
5. Кроме того, не стоит отрицать **пользу от понимания принципов работы** компьютера для его грамотного использования, и уж тем более для программирования.
6. Предлагается **два пути** усовершенствования курса информатики.
 - Больше **акцентировать внимание на аппаратных аспектах** реализации технологий.
 - Добавить **привлекательный материал о разработке hardware**.

