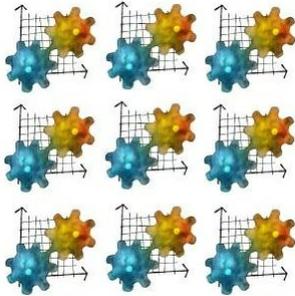


Е.А. Еремин

Об изучении основ параллельной обработки данных в школьном курсе информатики



Предлагается новый исполнитель –
параллельный Удвоитель



Москва, 2018

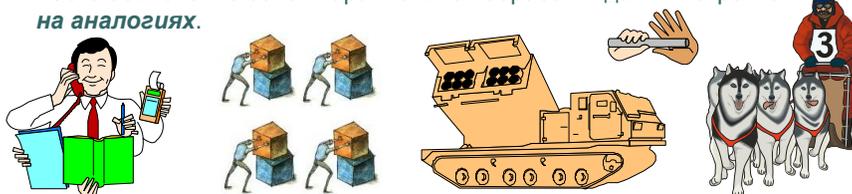
Новая реальность: многоядерные процессоры



- Сейчас они внутри **всех** компьютеров и бытовых электронных устройств.
- Каждое ядро работает **по отдельной программе**; между программами необходим **согласованный обмен данными** (требуется **понимание** происходящих процессов).
- Чтобы планировать параллельную работу нужны **особые** навыки мышления.

● ● ● | Нужен вычислитель...

Часто объяснение основ параллельной обработки данных строится *на аналогиях*.



При всей полезности аналогии, абсолютизировать ее все же не стоит.

«Между сходными в каком-либо отношении объектами может даже вовсе не оказаться никакого дальнейшего сходства, а может быть оно и есть, да идет не в том направлении, в каком мы его ждем.»

Майергойз Д.М. Аналогия в педагогическом процессе.
Математика в школе, 1947, N 1, с. 60-65

Предложение состоит в рассмотрении именно параллельной *вычислительной системы*.

● ● ● | Предлагается параллельный исполнитель



Для объяснения новой темы в школе предлагается *параллельный* исполнитель, построенный путем модификации известного вычислительного исполнителя *Удвоитель*.



Устройство исполнителя

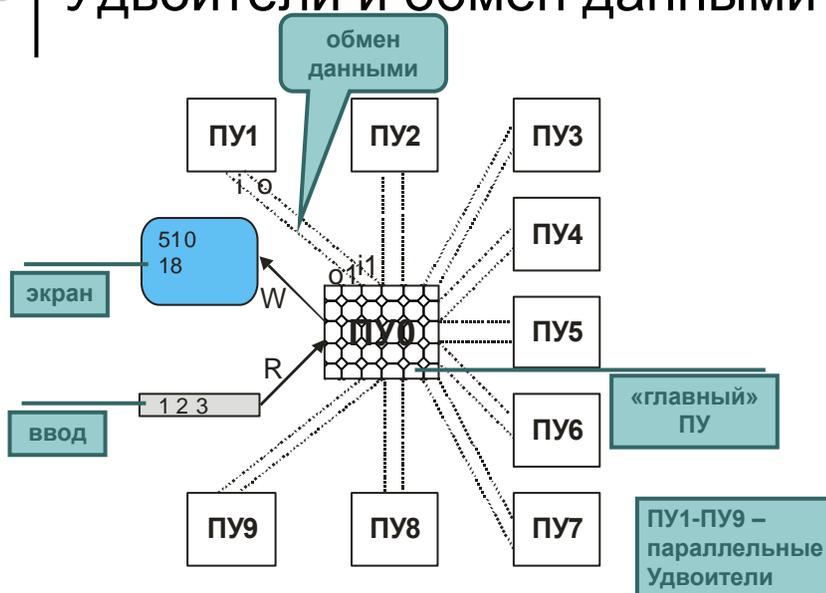
Возьмем несколько Удвоителей, способных работать параллельно. Будем называть их *параллельными Удвоителями* (ПУ). Присвоим каждому такому устройству номер: **от ПУ0 до ПУ9**.



Чтобы система работала согласованно (синхронно), необходимо ей управлять. Примем, что одно из устройств, скажем, **ПУ0**, является **главным**.

По сути ПУ0 делает то, что в реальных компьютерах выполняет многопроцессорная (многоядерная) операционная система.

Удвоители и обмен данными





СКИ: ПУ1-9 и ПУ0

Время выполнения пропорционально количеству тактов.

команда	действие	такты
1	$см = см + 1$	1
2	$см = см * 2$	1
0	$см = 0$	1
операции обмена данными		
i	из ПУ0 в см	2+ожид.
o	из см в ПУ0	3+ожид.
A	$см = см + \text{ввод}$	2+ожид.
см – это сокращенно сумматор		
операции с памятью		
MO	$M = 0$	1
MW	$M = см$	1
M+	$M = M + см$	1
MR	$см = M$	1

команда	действие	такты
1	$см = см + 1$	1
2	$см = см * 2$	1
0	$см = 0$	1
операции обмена данными		
iN	из ПУN в см	2+ожид.
oN	из см в ПУN	3+ожид.
AN	$см = см + \text{ввод}$	2+ожид.
R	чтение в см	2
R+	$см = см + \text{ввод}$	2
W	вывод см	2
операции с памятью		
MO	$M = 0$	1
MW	$M = см$	1
M+	$M = M + см$	1
MR	$см = M$	1



Примеры



Какие особенности параллельной обработки можно показать с помощью параллельного Удвоителя?

1. Простейший тест

Тест передает из ПУ0 в ПУ1 и ПУ2 число, соответствующее номеру ПУ, а те возвращают это значение, увеличенное на 1.

Выполнение:

	ПУ0	ПУ1	ПУ2
0			
1			
o1 === 1 ===>	i		
1	1		
o2 === 2 ===>	==== 2 =====>	i	
i1 <=== 2 ===	o	1	
W			
i2 <=== 3 ===	<===== 3 =====	o	
W			

Программа:
 ПУ1 и ПУ2:
 i 1 o
 ПУ0:
 0 1 o1 1 o2 i1 W i2 W

Важно
увеличивать
объем
вычислений

2. Много задач

параллелизм по задачам

Есть несколько программ для Удвоителя и мы хотим запустить их параллельно, каждую в отдельном ПУ

(как в ОС на многопроцессорном компьютере).

Пусть запускается несколько экземпляров *одинаковой* программы (гарантируется равный объем вычислительной работы).

Программа – это **умножение** некоторого входного параметра на 10^m , где m есть постоянное для конкретной серии расчетов значение.

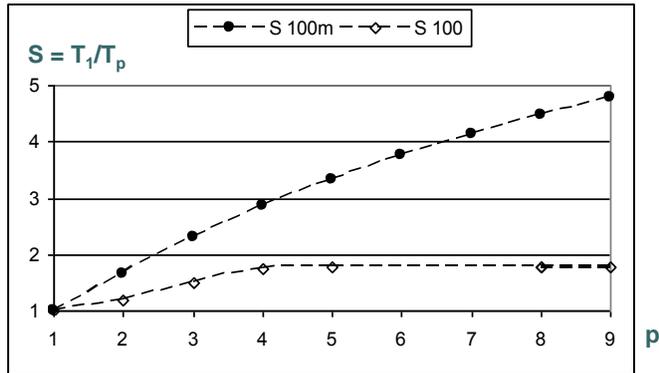
Итак, у нас есть два варианта:

- запускать задачи друг за другом по очереди в обычной однопроцессорной системе;
- выполнить их все сразу каждую в своем процессоре параллельной системы.

Будет ли выигрыш времени?



2. Результаты



(100m – это
100 000 000
= 10^8)

(S – ускорение вычислений, p – количество ПУ)



3. Сумма чисел

параллелизм по данным

Найти сумму последовательных чисел

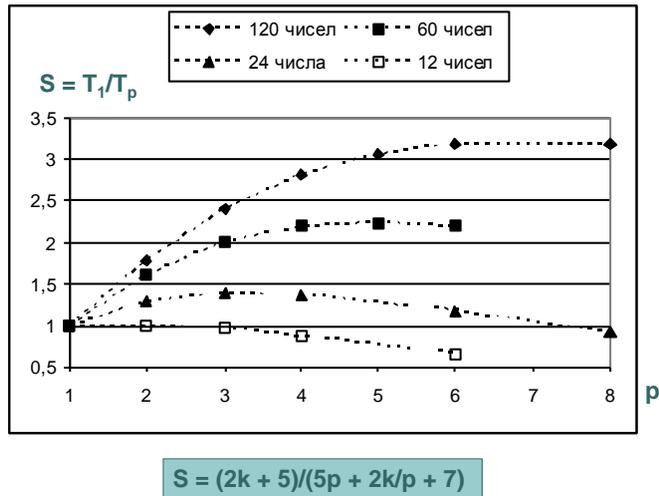
$$Y = 1 + 2 + 3 + \dots + k$$

Для определенности сейчас будем рассматривать случай, когда $k = 24$, а сумму вычисляют ПУ1-ПУ4 ($p = 4$). Тогда:

- ПУ1 просуммирует числа 1-6,
- ПУ2 – 7-12,
- ПУ3 – 13-18 и
- ПУ4 – 19-24.



3. Результаты



4. Двоичное число

Перевести из двоичной системы в десятичную заданное 8-битное число

$$Y = b_7 \cdot 2^7 + b_6 \cdot 2^6 + b_5 \cdot 2^5 + b_4 \cdot 2^4 + b_3 \cdot 2^3 + b_2 \cdot 2^2 + b_1 \cdot 2 + b_0$$

где b_m – значения битов числа.

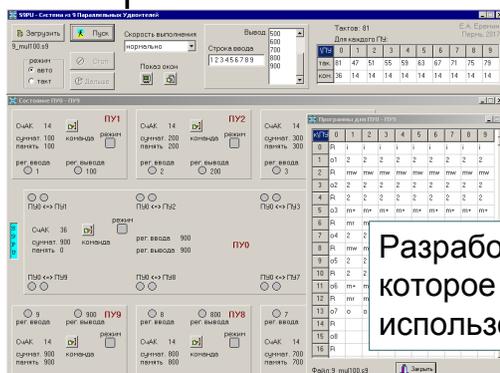
Испробуем несколько алгоритмов:

- для ПУ0
- для ПУ0 с ПУ1-ПУ7
- для ПУ0 с ПУ1-ПУ3
- для ПУ0 с ПУ1-ПУ2
- схема Горнера для ПУ0

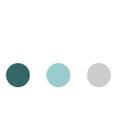
4. Результаты

алгоритм	тактов	ускорение
ПУ0	55	
ПУ0 с ПУ1-ПУ7	53	$55/53=1,04$
ПУ0 с ПУ1-ПУ3	43	1,28
ПУ0 с ПУ1-ПУ2	41	1,34
ПУ0, схема Горнера	25	2,2

Программная реализация



Разработано ПО для Windows, которое бесплатно могут использовать все желающие.



Выводы (1)

Формальное построение параллельного алгоритма далеко не всегда приводит к успеху. Отсюда следует, что:

1. возможности автоматизации ограничены (а значит, не стоит в нее слепо верить!);
2. необходимо учиться строить эффективные алгоритмы.



Выводы (2)

Работая с параллельным Удвоителем, можно показать следующие важные идеи.

- Параллельные части должны быть максимально независимы, а обмен данными – минимальным.
- Эффективность параллельной вычислительной системы существенно зависит от объема вычислений (по отношению ко времени обмена). Т.е. плохо для небольших задач.
- Эффективность растет с ростом количества однотипных данных. Это хорошо для практики.
- Время вычислений существенно зависит от обмена данными, следовательно, важно уметь грамотно планировать ход вычислений.
- Часто применение математических знаний дает больший эффект, чем мощная параллельная система.

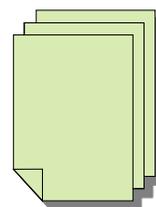
Выводы (3)

При работе с параллельным Удвоителем полезно совершенствовать некоторые имеющиеся у учеников умения:

- обработка и анализ результатов с помощью электронных таблиц;
- вывод и анализ формул (ускорение расчета на параллельной системе);
- применение программирования (формирование текстовых файлов программ с заданным числом повторений команд).

Загрузить программу можно с сайта журнала «Информатика в школе».

Публикации



1. Еремин Е.А. **Разработка исполнителя для демонстрации процесса параллельных вычислений.** Вестник Пермского государственного гуманитарно-педагогического университета. Выпуск 13, 2017, с.5-18.
<http://elibrary.ru/item.asp?id=32765186>
2. Еремин Е.А. **Исполнитель для изучения в курсе информатики основ параллельных вычислений.** Информатика в школе, 2018, N 2, с.30-40.

Спасибо за внимание!