

# **Лекция 2**

## **Конвейерные вычисления**

**Ефимов Александр Владимирович**  
**E-mail: [efimov@cpct.sibsutis.ru](mailto:efimov@cpct.sibsutis.ru)**

**Курс «Архитектура вычислительных систем»**  
**СибГУТИ, 2017**

# Конвейерные ВС

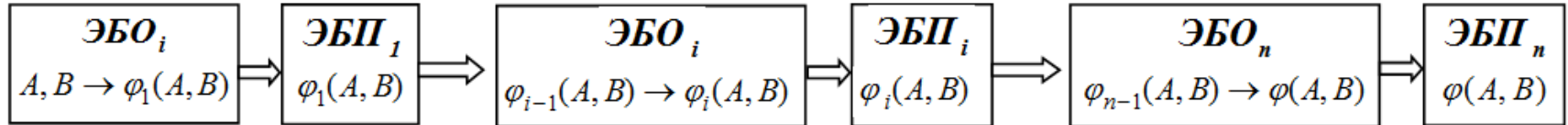
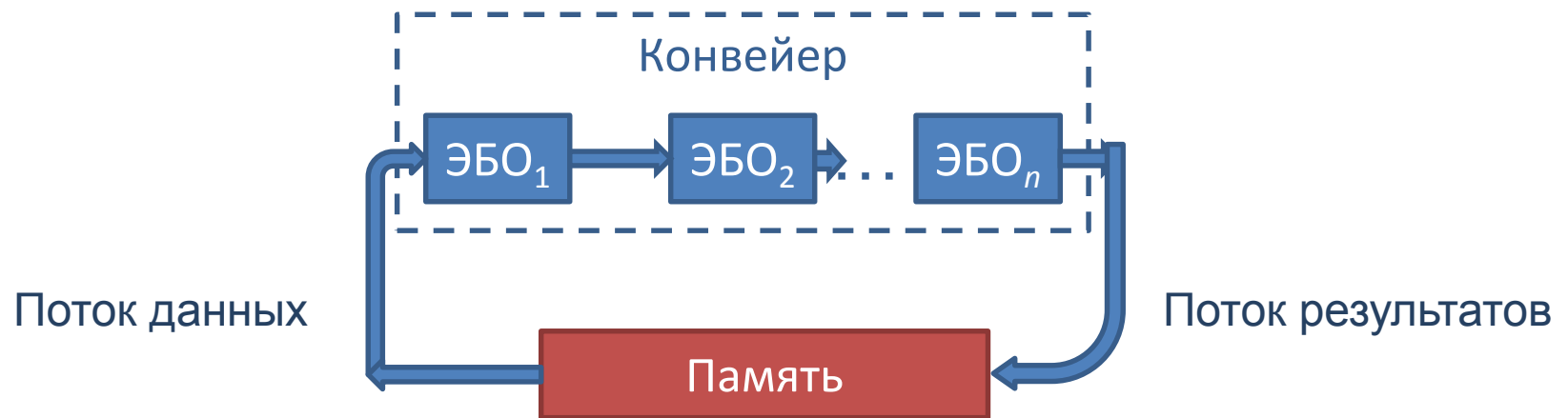
- Конвейерный способ обработки информации;
- аппаратная реализация операции над векторами данных;

*Например:  $A + \alpha B$ ,  $(A + \alpha)B$ , где  $\alpha$  - скаляр*

$$A = (A_1, A_2, \dots, A_i, \dots, A_n), B = (B_1, B_2, \dots, B_i, \dots, B_n)$$

- в 70-х и 80-х годах 20 века обеспечивали быстродействие порядка  $10^8 - 10^9$  опер./с;
- современные быстродействующие микропроцессорные БИС основаны на конвейеризации вычислений.

# Каноническая функциональная структура



$$A, B \rightarrow \varphi_1(A, B) \rightarrow \dots \rightarrow \varphi_i(A, B) \rightarrow \dots \rightarrow \varphi_n(A, B) = \varphi(A, B).$$

$A, B$  — векторы-операнды;  $\varphi_i(A, B)$  — частичное преобразование векторов  $A, B$

# Функционирование конвейера (pipeline)

$A_1, B_1 \rightarrow \varphi_1(A_1, B_1)$				
$A_2, B_2 \rightarrow \varphi_1(A_2, B_2)$	...			
...	...			
$A_i, B_i \rightarrow \varphi_1(A_i, B_i)$	...	$\varphi_{i-1}(A_1, B_1) \rightarrow \varphi_i(A_1, B_1)$		
$A_{i+1}, B_{i+1} \rightarrow \varphi_1(A_{i+1}, B_{i+1})$	...	$\varphi_{i-1}(A_2, B_2) \rightarrow \varphi_i(A_2, B_2)$	...	
	...	...	...	
$A_n, B_n \rightarrow \varphi_1(A_n, B_n)$	...	$\varphi_{i-1}(A_{n-i+1}, B_{n-i+1}) \rightarrow$ $\varphi_i(A_{n-i+1}, B_{n-i+1})$	...	$\varphi_{n-1}(A_1, B_1) \rightarrow \varphi(A_1, B_1)$
$A_{n+1}, B_{n+1} \rightarrow \varphi_1(A_{n+1}, B_{n+1})$	...	$\varphi_{i-1}(A_{n-i+2}, B_{n-i+2}) \rightarrow$ $\varphi_i(A_{n-i+2}, B_{n-i+2})$	...	$\varphi_{n-1}(A_2, B_2) \rightarrow \varphi(A_2, B_2)$
...	...	...	...	...

# Функционирование конвейера (pipeline)



1958 г. ЭВМ М-20 реализовано совмещение IF и EX

1963 г. ЭВМ ATLAS реализовано совмещение IF, ID, RD и EX

1966 г. ЭВМ БЭСМ-6 реализован конвейерный процессор

# Конвейерные систем типа «память-память»

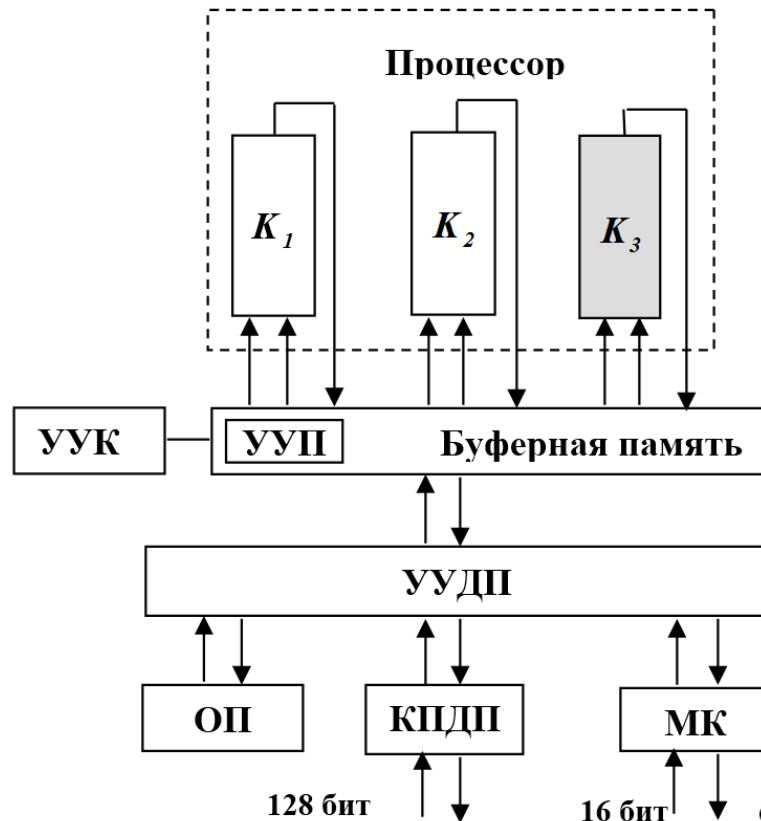
CDC (Control Data Corporation), Сеймур Крей и Уильям Норрис



# Вычислительная система STAR-100

- STAR – STring ARray computer – векторный компьютер;
- разработка с 1965 по 1973;
- быстродействие –  $10^8$  оп./сек.;
- стоимость 15 млн. \$;
- создавалась под непосредственным влиянием языка программирования APL (A Programming Language):  
Язык APL (или АПЛ) – диалоговый язык программирования, характеризуется развитыми средствами работы с регулярными структурами данных (векторами, матрицами, массивами) и богатым набором базовых операций и компактностью записи.

# Функциональная структура STAR-100



$K_1, K_2$  – конвейеры выполнения векторных операций (floating-point pair pipelines);  
 $K_3$  – конвейер реализации операций над калярными операндами (string data pipeline);  
УУК – устройства управления командами;  
УУП – устройства управления потоками;  
УУДП – устройства управления доступом к памяти;  
КПДП – канала прямого доступа в память;  
Сеть из 9 мини-ЭВМ (Распределённая ОС)

МК – мультиплексированный канал;



# Функциональные блоки STAR-100

- Каждый конвейер  $K_1$ ,  $K_2$ ,  $K_3$  мог включать **до 30 блоков** обработки информации;
- Конвейеры  $K_1$ ,  $K_2$ ,  $K_3$  имели **программируемую структуру**, следовательно на одном и том же множестве элементарных блоков обработки можно выполнять различные арифметические операции;
- В конвейерах  $K_1$  и  $K_2$  использовался служебный булевский вектор для обеспечения **избирательная обработка компонентов** векторов-операндов;
- Состав элементарных блоков обработки информации конвейеров выбран с учетом распределения вероятностей использования микроопераций различных типов;
- Каждый конвейер воспринимал **64-разрядный код** либо как один 64-разрядный операнд, либо как два 32-разрядных операнда.
- Время выполнения операции над парой операндов в любом из блоков конвейеров не превышало 40 нс.

# Функциональные блоки STAR-100

**УУК** имело буфер опережающего просмотра команд (емкостью в четыре 512-разрядных суперслова) со стековым механизмом работы;

**УУП** использовалось для управления потоками операндов и команд между УУДП, конвейерами и УУК;

**УУДП** управляло 4-мя параллельными каналами обращения к памяти (2 – чтение операндов в конвейеры  $K_1$  и  $K_2$ , 1 – запись результатов, 1 – работа с устройствами ввода-вывода);

**Буферная память** совокупность регистров с временем цикла 40 нс;

**ОП** (оперативная память) использовалась для хранения программ и данных. Реализована на магнитных сердечниках. Ёмкость 512 – 1024 К 64-разрядных слов, т.е. до 8 Мбайт. Время цикла памяти равно 1,28 мкс.

# Функционирование STAR-100

Набор команд STAR-100 из 230 шт.:

65 команд предназначалось для работы с векторами данных;

130 команд для работы со скалярами;

Операционная система (ОС) STAR-100 относилась к классу распределенных;

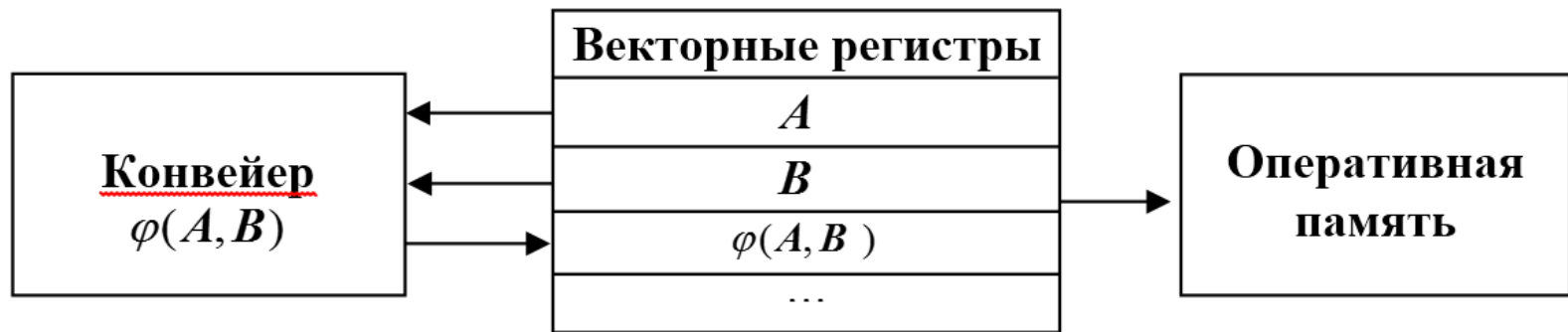
функции реализовывались специальной вычислительной сетью из 9 мини-машин;

Система программирования STAR-100 включала компиляторы с языков APL-STAR, COBOL и FORTRAN.

# Конвейерные систем типа «регистр-регистр»

Cray Research Inc

Сеймур Крей



# Литература

Хорошевский В.Г. Архитектура вычислительных систем. Учебное пособие. – М.: МГТУ им. Н.Э. Баумана, 2005; 2-е издание, 2008.

Хорошевский В.Г. Инженерные анализ функционирования вычислительных машин и систем. – М.: “Радио и связь”, 1987.