

# Повышение быстродействия ВУ

## 1 Преобразование операционной части.

1.1 Увеличение масштабов аппаратной реализации функций.

1.2 Уменьшение глубины логических схем и повышение тактовой частоты.

1.3 Параллельная обработка.

1.4 Другие способы повышения быстродействия.

## 2 Преобразование управляющей части.

2.1 Одновременное считывание нескольких микрокоманд.

2.2 Конвейерное выполнение микрокоманд.

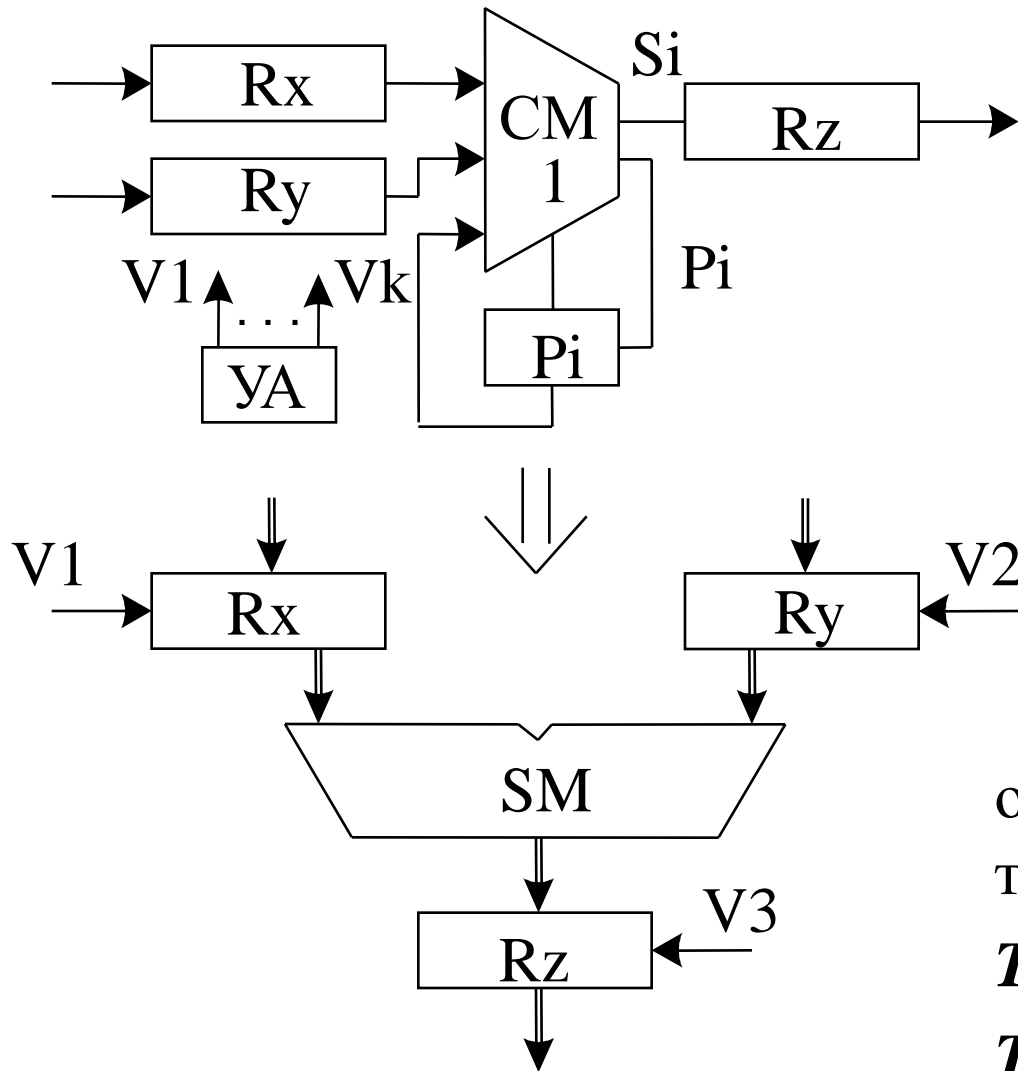
- **Знать:** способы преобразования операционной и управляющей частей ВУ: увеличение масштабов аппаратной реализации функций; уменьшение глубины логических схем и повышение тактовой частоты; многоэлементная и многостадийная обработка; одновременное считывание микрокоманд.
- **Уметь:** преобразовать структуру ВУ с целью повышения быстродействия и оценить достигнутый результат.
- **Помнить:** о сочетании различных способов повышения быстродействия ВУ.
- **Литература:** [1,14].

# **1 Преобразование операционной части**

## **1.1 Увеличение масштабов аппаратной реализации функций**

- Под увеличением масштабов аппаратной реализации функции понимают переход от разворачивания вычислительного процесса во времени (микропрограммная реализация) к разворачиванию его в пространстве (аппаратная реализация).
- Примерами такого перехода могут служить: преобразование последовательного сумматора в параллельный (параллельно-последовательный), преобразование устройства умножения в матричный умножитель и др.

# Переход от последовательного сумматора к параллельному



$$T_1 = t_{\text{Ц}} * n,$$

$t_{\text{Ц}}$  – время цикла суммирования,  $n$  – разрядность операндов;

$$t_{\text{Ц}} = t * k,$$

$t$  – время выполнения МК,  $k$  – число МК ( $k=1$ ).

$$t = (\tau_{\text{CM}} + \tau_T) + \tau_{\text{YA}}$$

$\tau_{\text{CM}}$  – задержка сигнала на одноразрядном сумматоре,  $\tau_T$  – триггере переноса,  $\tau_{\text{YA}}$  – в YA.

$$T_1 = k * (\tau_{\text{CM}} + \tau_T + \tau_{\text{YA}}) * n;$$

$$T_2 = \tau_{\text{CM}} * n + \tau_T + \tau_{\text{YA}}; \quad T_2 \ll T_1.$$

## 1.2 Уменьшение глубины логических схем и повышение тактовой частоты

- Длительность тактового периода:  $t = t_{OY} + t_{YU}$ .
- Тактовая частота работы ВУ:  $F = 1/t$ .
- Время задержки сигнала в ОУ:  $t_{OY} = \max\{t_1, \dots, t_s, \dots, t_S\}$ , где  $t_s$  – время выполнения s-й микрооперации (МО),  $S$  – число МО. Время задержки сигнала в УУ определяется аналогично.
- Время выполнения МО определяется глубиной логической схемы:  $t_{MO} = \tau * (l_{KC} + l_T)$ , где  $\tau$  – время задержки сигнала на логическом элементе;  $l_{KC}$  – глубина логической схемы (число логических элементов, через которые проходит сигнал в комбинационной схеме при выполнении МО);  $l_T$  – число логических элементов, через которые проходит сигнал при фиксации результата в триггерах.

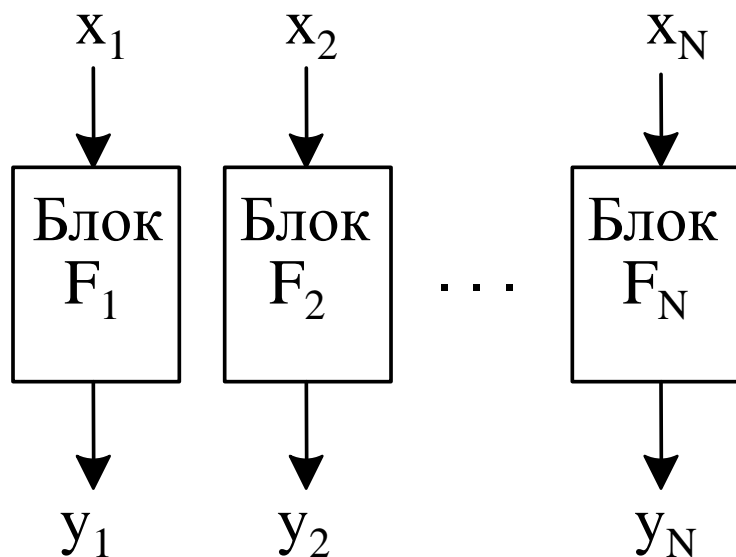
## *Повышение тактовой частоты*

- Уменьшение глубины логических схем позволяет сократить число логических уровней через которые проходит сигнал.
- При представлении логических функций в виде дизъюнктивной формы уровней может быть не более 3: «НЕ», «И», «ИЛИ»).
- Уменьшение числа логических уровней позволяет сократить время выполнения МО, что обеспечивает повышение тактовой частоты работы ВУ.
- Повысить тактовую частоту также можно используя более быстродействующую элементную базу, с меньшим временем задержки сигнала на логическом элементе ( $\tau$ ).

# 1.3 Параллельная обработка

## *Многоэлементная*

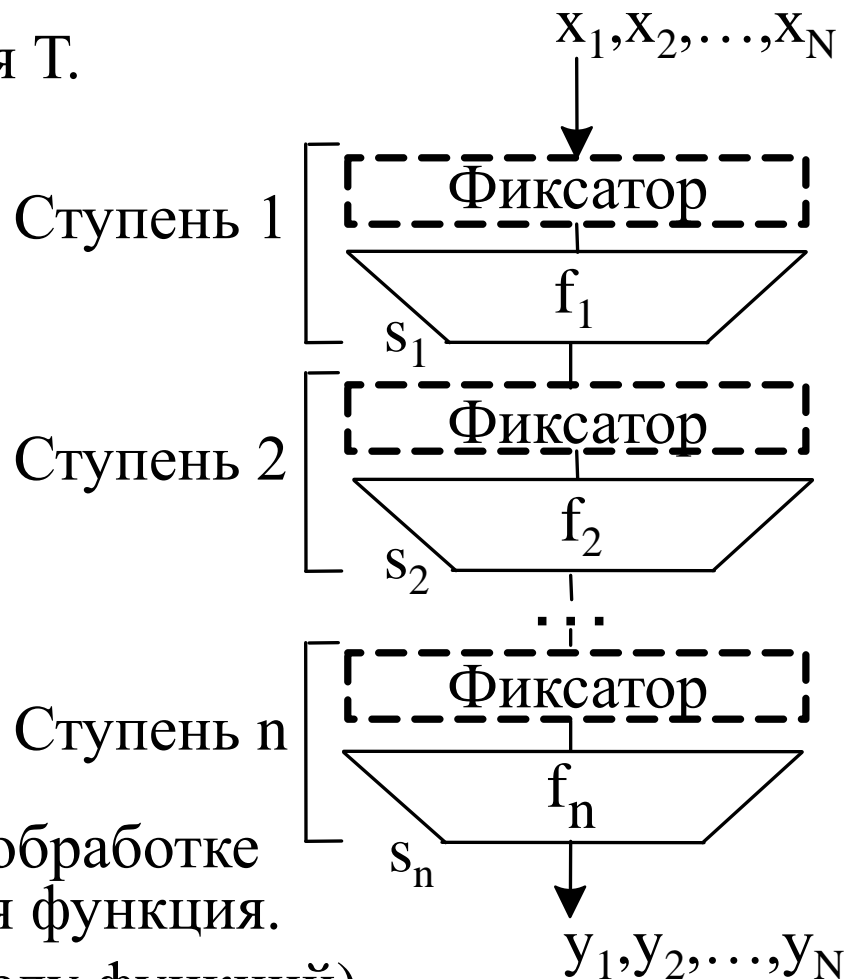
При многоэлементной обработке  $N$  одинаковых функций:  $F_1, F_2, \dots, F_N$  вычисляется одновременно за время  $T$ .



При установившейся конвейерной обработке за время  $T/n$  вычисляется очередная функция.

При  $n=N$  (число ступеней равно числу функций)  $N$  функций в пределе будет вычислено за время  $T$ .

## *Многостадийная (конвейерная)*



# Декомпозиция функции и запуск конвейера

Предполагается представление исходной функции  $F$  в виде суперпозиции функций  $f_1, f_2, f_3, f_4$  одинаковой сложности:  $F = f_4(f_3(f_2(f_1(x_1))))$ .

$$s_i = f_i(s_{i-1}), \quad i = 1, \dots, n; \quad s_0 = x_1; \quad s_n = y_1.$$

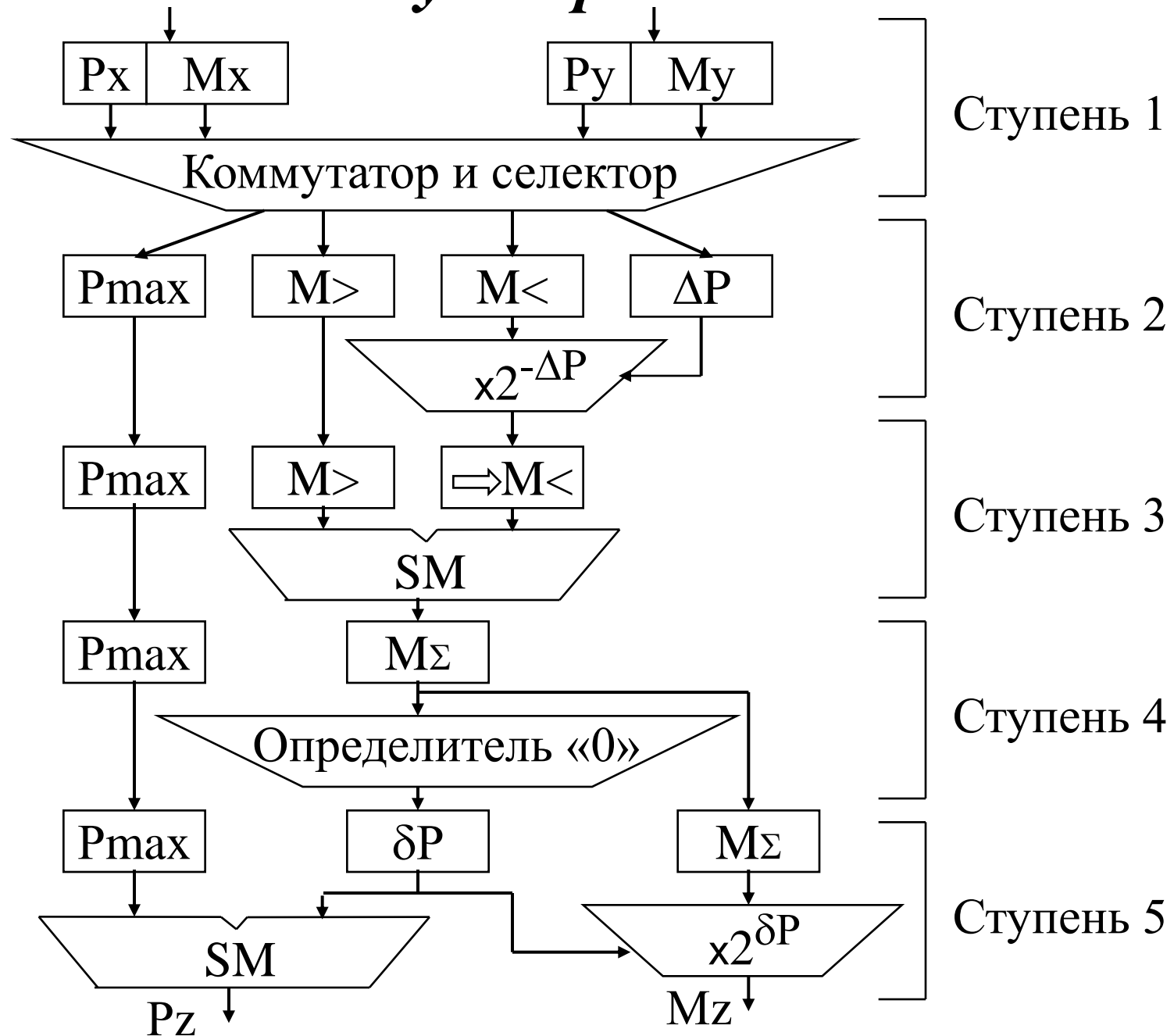
Степень	Такт (вход)			
	$T_1$	$T_2$	$T_3$	$T_4$
	$x_1$	$x_2$	$x_3$	$x_4$
$f_1$	$s_1 = f_1(x_1)$	$s_1 = f_1(x_2)$	$s_1 = f_1(x_3)$	$s_1 = f_1(x_4)$
$f_2$	—	$s_2 = f_2(s_1)$	$s_2 = f_2(s_1)$	$s_2 = f_2(s_1)$
$f_3$	—	—	$s_3 = f_3(s_2)$	$s_3 = f_3(s_2)$
$f_4$	—	—	—	$y_1 = s_4 = f_4(s_3)$



## ***Виды конвейеров***

- Конвейеры делятся на синхронные и асинхронные:
  - в синхронных конвейерах одновременно происходит передача информации во всех ступенях;
  - в асинхронных конвейерах результаты на следующую ступень передаются по мере её готовности.
- В общем случае на каждой из ступеней конвейера может выполняться одна из нескольких функций. При этом ступень настраивается на заданную функцию перед выполнением вычисления.
- Могут быть использованы конвейеры с замыкаемыми и размыкаемыми обратными связями.

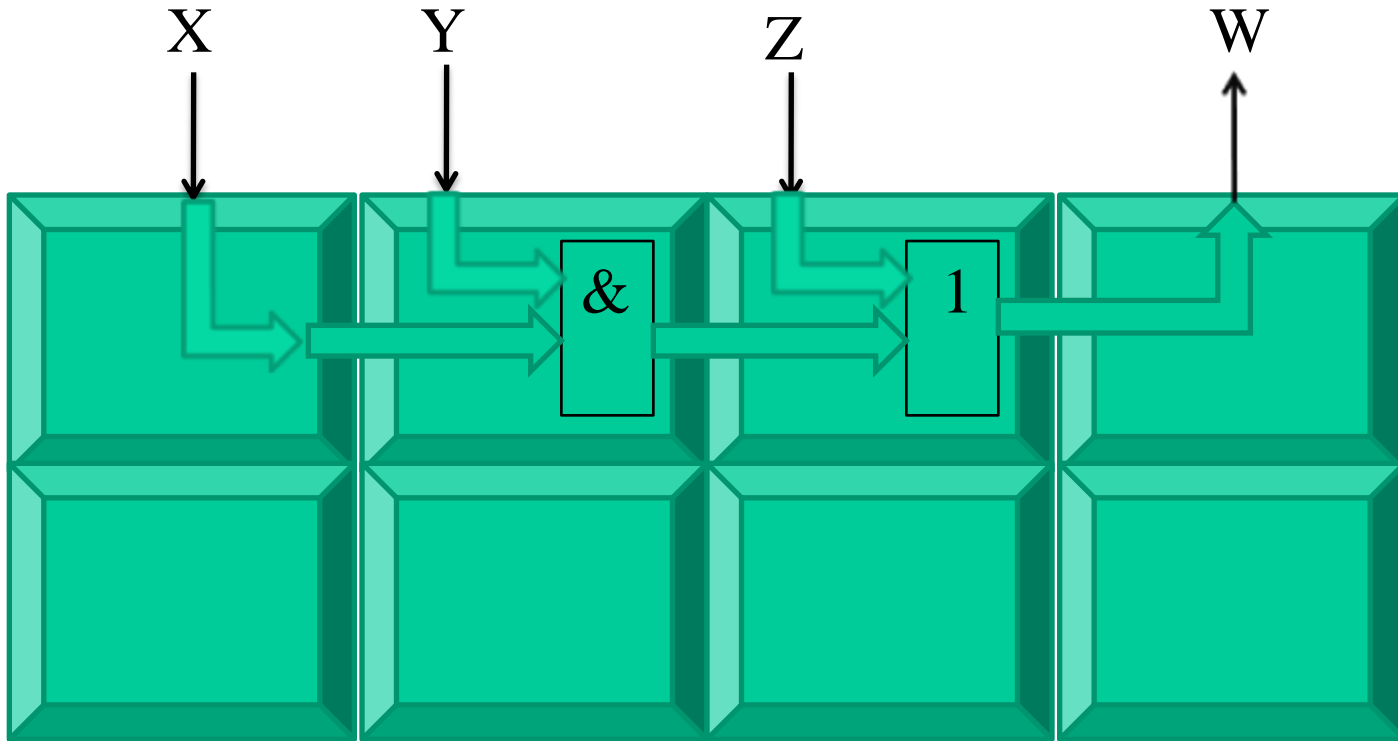
# АЛУ для суммирования чисел ПЗ



## 1.4 Другие способы повышения быстродействия

- Алгоритмические способы повышения быстродействия (ускоренные алгоритмы выполнения операций, например, умножение с просмотром двух разрядов, умножение с запоминанием переносов и т.п.).
- Однородные вычислительные среды (автоматы с настраиваемой структурой).
- Волновые структуры.
- Апериодические схемы (автоматы).

## *Однородные вычислительные среды*



$$W = X \& Y \vee Z$$

## 2 Преобразование управляющей части

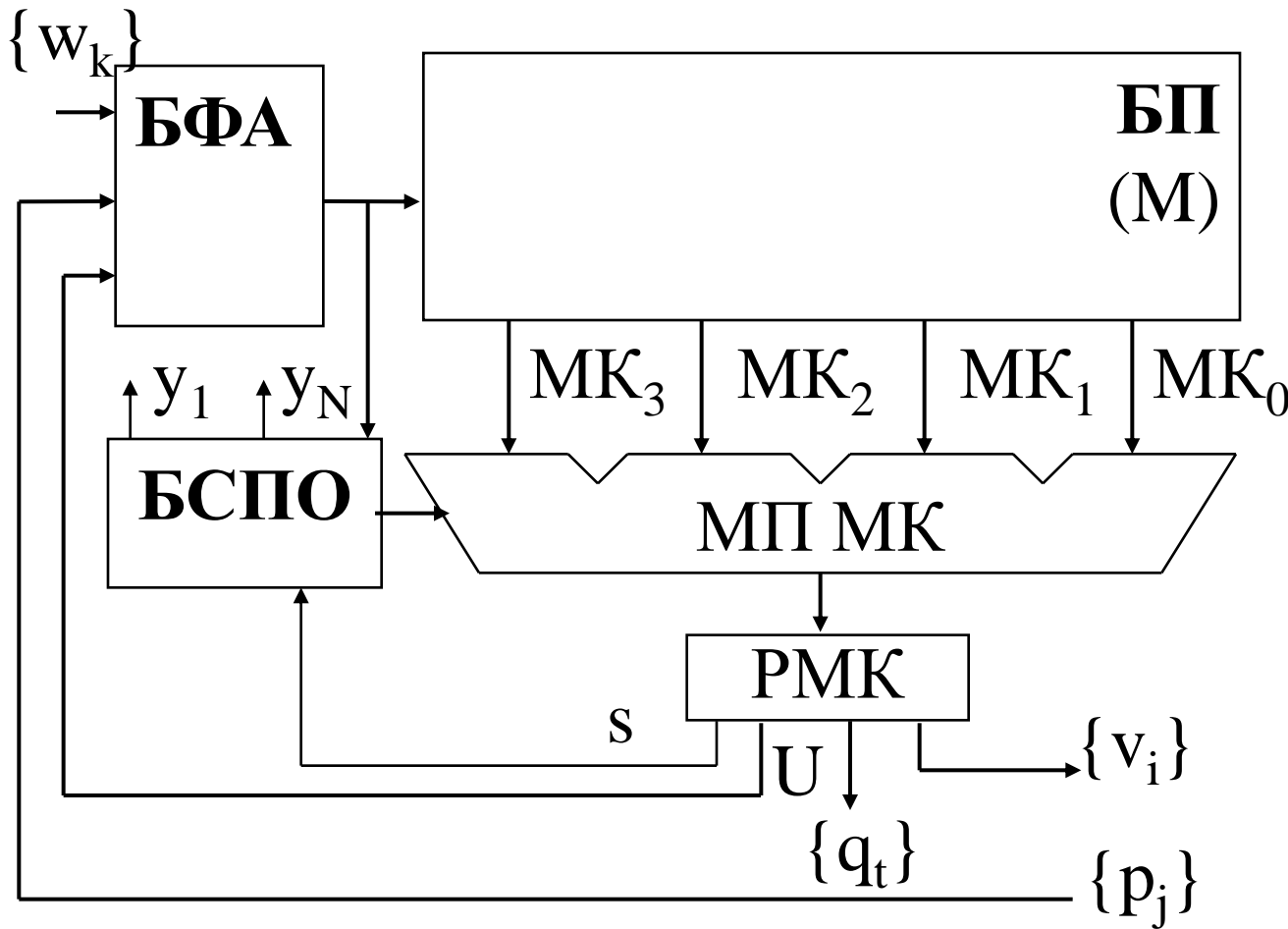
### 2.1 Одновременное считывание микрокоманд

Время выполнения микрокоманды:  $t = \tau_{OY} + \tau_{YU}$ , где  $\tau_{OY}$  — время задержки сигнала в ОУ, а  $\tau_{YU}$  — в УУ.

При использовании в ВУ устройства управления с программируемой логикой задержку в УУ можно оценить по формуле:  $\tau_{YU} = \tau_{БФА} + \tau_{ЧТ}$ , где  $\tau_{БФА}$  — время задержки сигнала в блоке формирования адреса МК, а  $\tau_{ЧТ}$  — время чтения МК из блока памяти микропрограмм. Причем, как правило,  $\tau_{ЧТ} \gg \tau_{БФА}$ .

Поэтому сокращение времени чтения микрокоманд из памяти микропрограмм может обеспечить значительное повышение быстродействия ВУ.

# Пример одновременной выборки МК



Эффективно на линейных участках и циклах, «замыкаемых» на считанных МК.

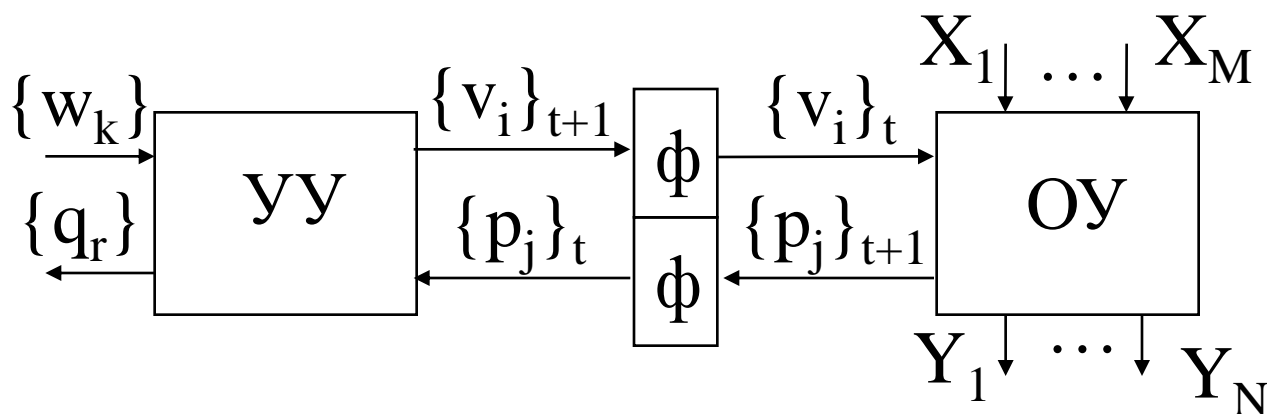
$$t = (1 - P) * t_1 + P * t_2$$

$P$  – вероятность использования уже считанной МК.

$t_1 = \tau_{OY} + \tau_{БФА} + \tau_{ЧТ} + \tau_{МП} + \tau_{РМК}$  – МК нет среди считанных;  
 $t_2 = \tau_{OY} + \tau_{МП} + \tau_{РМК}$  – МК есть среди считанных.

## 2.2 Конвейерное выполнение микрокоманд

- Процесс выполнения микрокоманды обычно делится на два этапа, приблизительно равной продолжительности:
  - Формирования адреса и чтения МК в УУ,
  - Выполнения микрокоманды и формирования значений логических условий в ОУ.
- Ступенями конвейера являются УУ и ОУ.
- Для совмещения во времени процессов в ОУ и УУ между ними устанавливаются фиксаторы ( $\phi$ ) – буферные (конвейерные) регистры (БР).

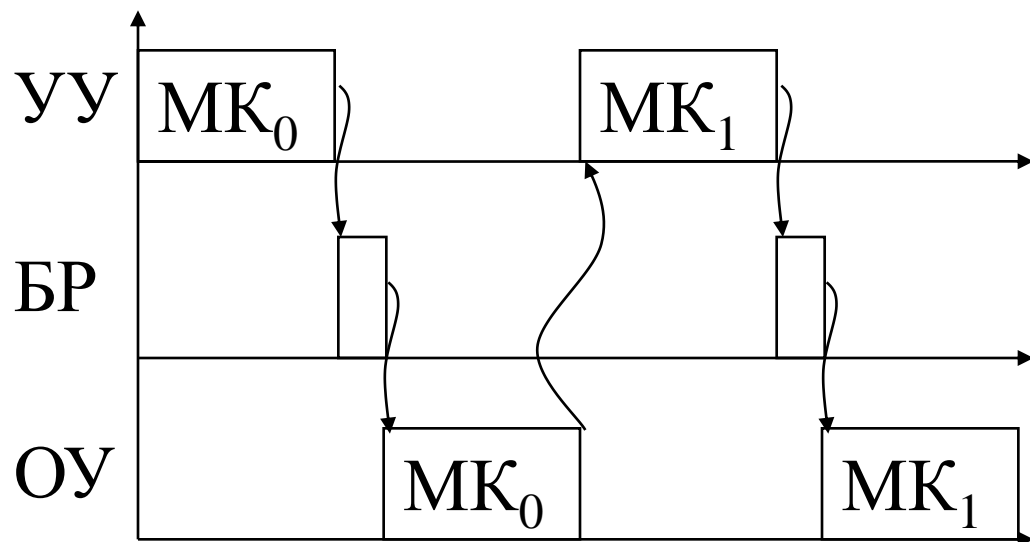


## Обычное выполнение

## Конвейерное выполнение

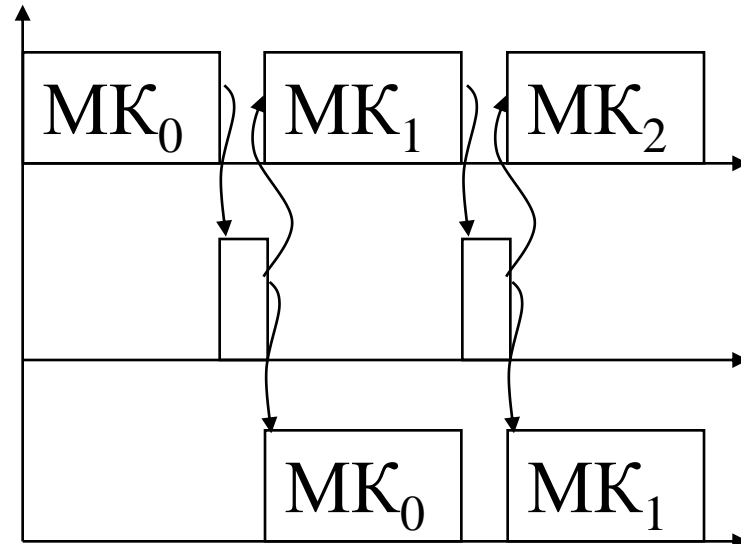
Чтение МК

Чтение МК



Выполнение МК

$$t = \tau_{yy} + (\tau_{БР}) + \tau_{ОУ}$$



Выполнение МК

$$t = \tau_{БР} + \max\{\tau_{yy}, \tau_{ОУ}\}$$



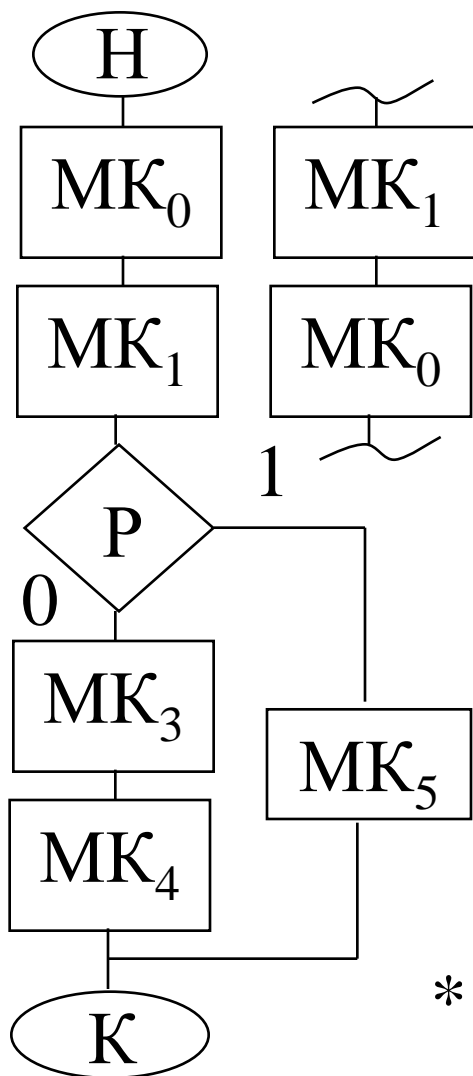
## *Эффективность работы конвейера: предсказание переходов*

- Эффективность работы конвейера максимальна на линейных участках и может снизиться при большом числе ветвлений в микропрограмме (МП).
- Для ослабления влияния ветвлений на эффективность работы конвейера может быть использовано предсказание переходов.
- При предсказании перехода считывается МК, имеющая наибольшую вероятность выполнения. Если значение условия угадать не удалось, то происходит сбой конвейера и повторное считывание нужной МК.

## *Эффективность работы конвейера: преобразование микропрограммы*

- Если алгоритм позволяет, то выполняется преобразование исходной МП: микрокоманда, вычисляющая значение логического условия, перемещается на одну МК к началу МП. При этом вычисление значения логического условия производится заранее и не нарушает работу конвейера.
- Если алгоритм не позволяет преобразовывать МП путем перемещения МК, то в МП вводятся пустые МК. Точнее, МК ветвления заменяется двумя МК. Первая МК выполняет необходимые микрооперации и вычисляет значение логического условия, а вторая («пустая») — выполняет только переход по вычисленному первой значению логического условия.

# Пример преобразования микропрограммы для ускорения работы конвейера



Степень	Такт			
	T <sub>1</sub>	T <sub>2</sub>	T <sub>3</sub>	T <sub>4</sub>
УУ	MK <sub>0</sub>	MK <sub>1</sub>	?	
ОУ	-	MK <sub>0</sub>	MK <sub>1</sub>	
		P=0*		
УУ	MK <sub>1</sub>	MK <sub>0</sub>	MK <sub>3</sub>	MK <sub>4</sub>
ОУ	-	MK <sub>1</sub>	MK <sub>0</sub>	MK <sub>3</sub>

\* - значение условия Р формируется при выполнении МК<sub>1</sub>