

УНИВЕРСИТЕТ ИТМО

Курс «Системы на кристалле»

## Лекция 3

# Технология высокоуровневого синтеза (часть 2)

БЫКОВСКИЙ С.В

e-mail: [sergei\\_bykovskii@itmo.ru](mailto:sergei_bykovskii@itmo.ru)

Санкт-Петербург, 2019



## Операции, выполняемые в процессе синтеза

- ✓ **Binding.** Определение ресурсов, с помощью которых можно выполнить операции.
- ✓ **Scheduling.** Планирование вычислительного процесса. Распределение операций по тактам.
- ✓ **Control logic extraction.** Извлечение логики управления. Создание управляющего автомата

## Трансляция ресурсов

- ✓ Функциональные блоки описываются в виде функций
- ✓ Аргументы транслируются в порты ввода/вывода
- ✓ Переменные и массивы транслируются в элементы памяти (регистровую или блочную)
- ✓ Циклы транслируются в одноктактовые, многотактовые или конвейерные вычислительные блоки

## Конструкции C/C++, которые нельзя синтезировать в современных САПР

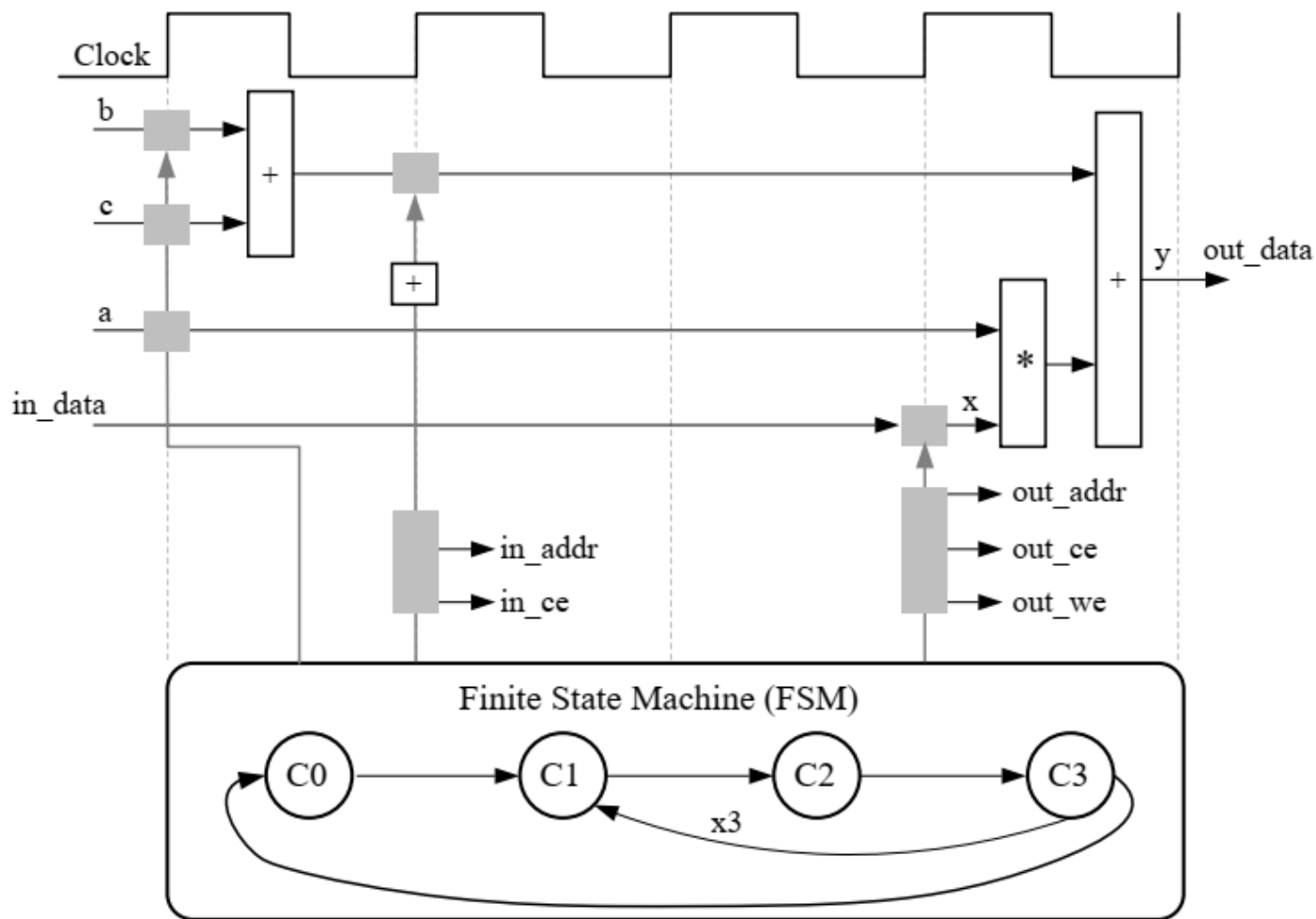
- ✓ Системные вызовы
- ✓ Динамическое выделение памяти
- ✓ Приведение указателей к типу void \*
- ✓ Рекурсивные функции



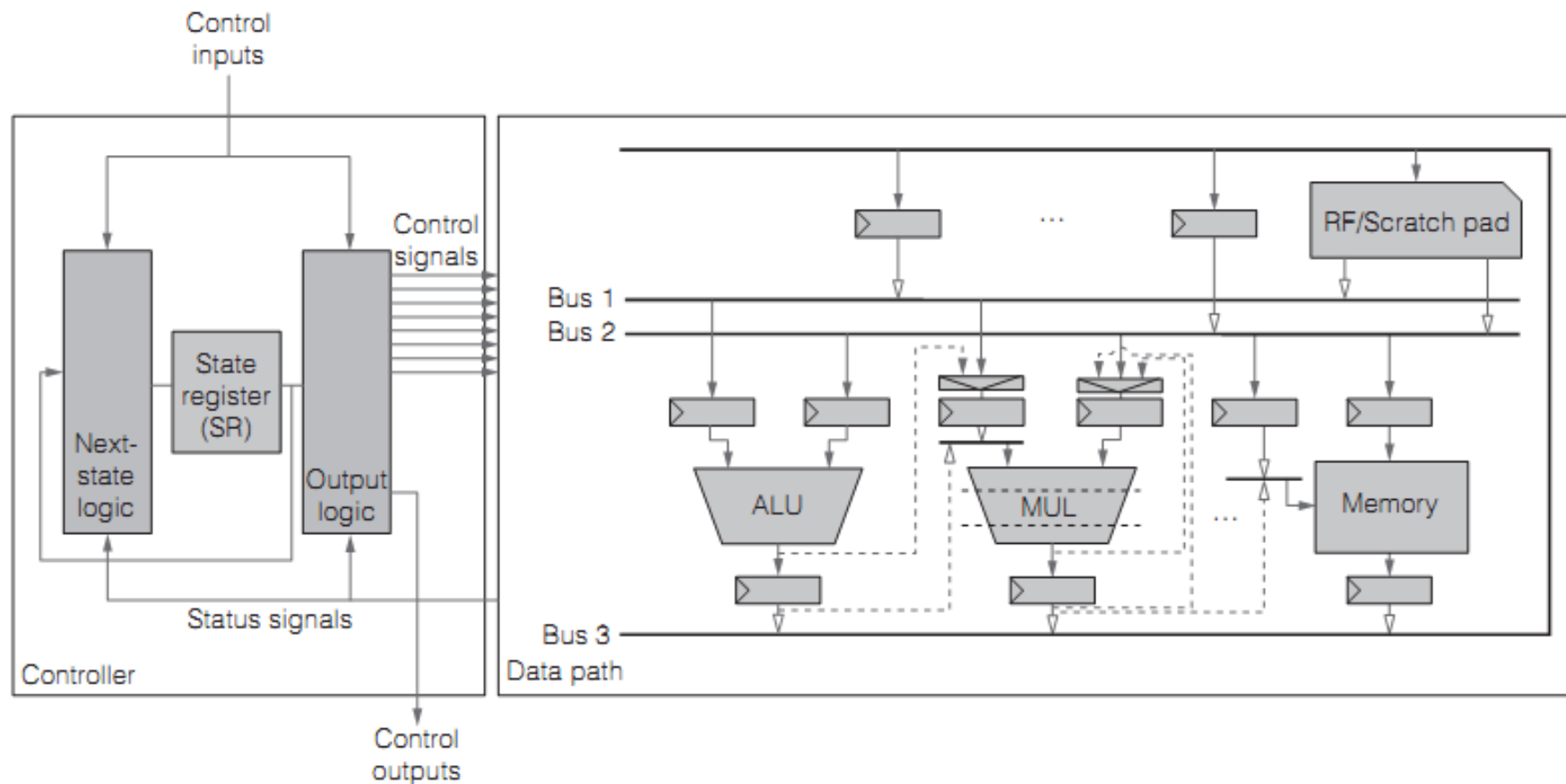
## Пример 1

```
void foo (int in[3], char a, char b,  
char c, int out[3]) {  
    int x, y;  
    for (int i = 0; i < 3; i++) {  
        x = in[i];  
        y = a*x + b + c;  
        out[i] = y;  
    }  
}
```

## Извлечение логики управления и синтез RTL



# Типовой шаблон синтеза



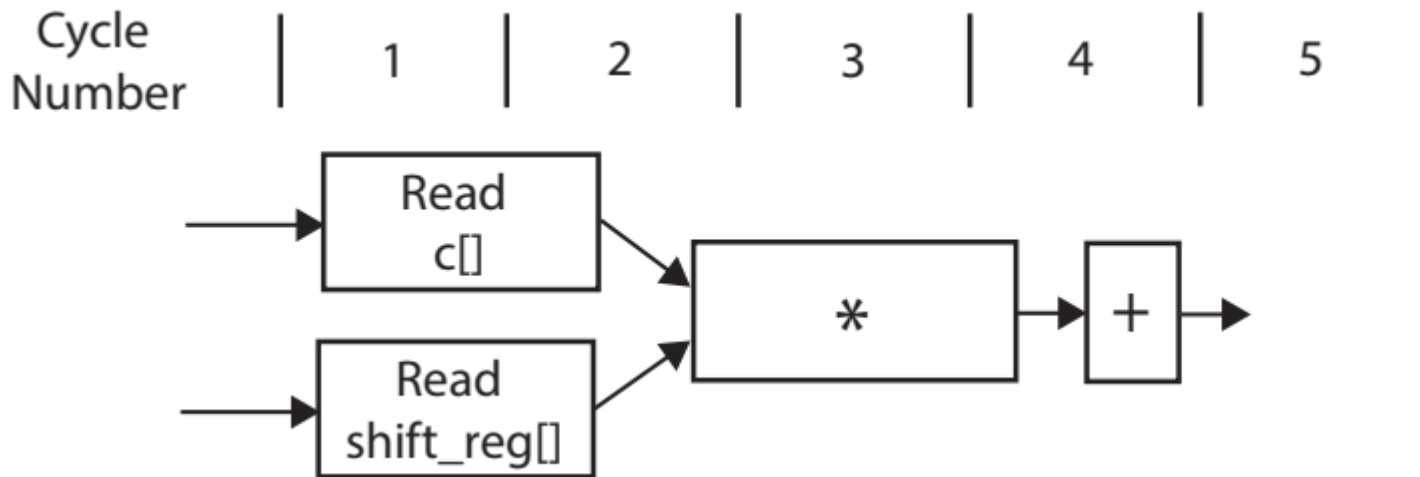
## Пример 2. Потактовая реализация

```

for (i = N - 1; i >= 0; i--) {
    acc += shift_reg[i] * c[i];
}
    
```

Latency =  $4 * N$

Interval =  $4 * N$





## Временные характеристики

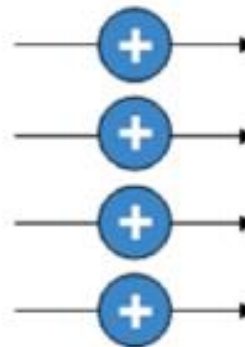
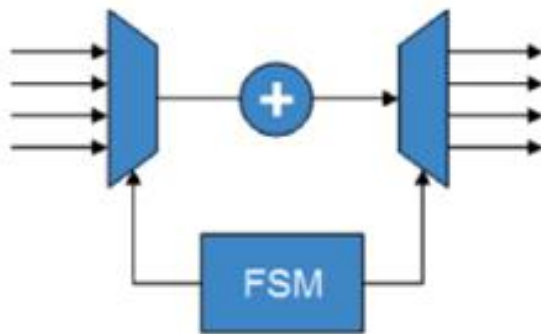
- ✓ **Latency** – время между подачей входных данных и получением результата
- ✓ **II** (Iteration interval или Initiation Interval) – минимальное время между посылками новых порций входных данных
- ✓ Времена измеряются в тактах сигнала синхронизации!

# Оптимизации циклов

- Развертка циклов (Unrolling)
- Слияние циклов (loops merging)
- Конвейеризация циклов (pipelining)

```
for (i = 0; i<4; i++)
{
    r[i] = a[i] + b[i];
}
```

```
r[0] = a[0] + b[0];
r[1] = a[1] + b[1];
r[2] = a[2] + b[2];
r[3] = a[3] + b[3];
```



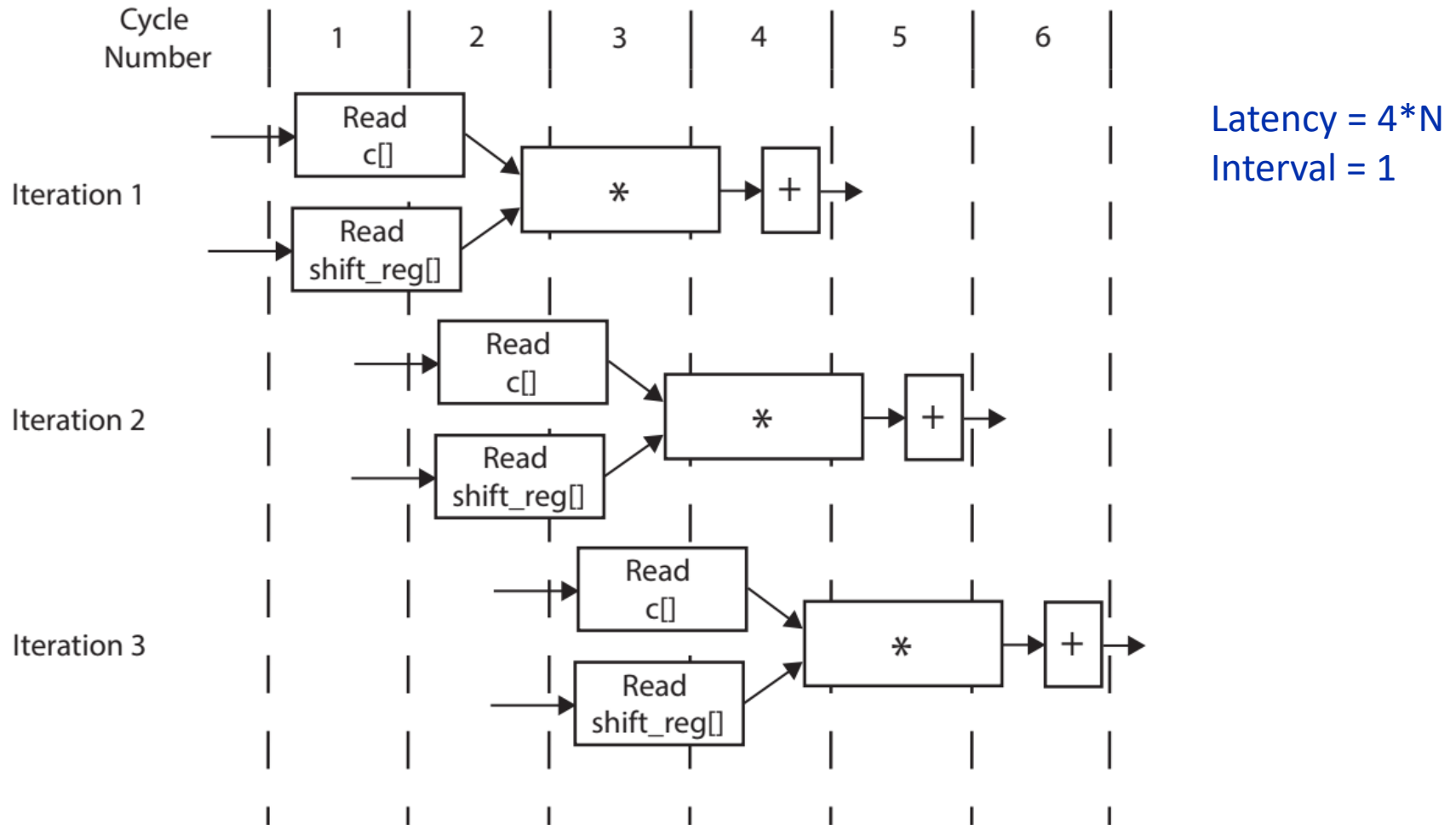
## Оптимизации | Развертка циклов

```
for (i = N - 1; i >= 0; i--) {  
    acc += shift_reg[i] * c[i];  
}
```



```
for (i = N - 1; i >= 3; i -= 4) {  
    acc += shift_reg[i] * c[i] + shift_reg[i - 1] * c[i - 1] +  
    shift_reg[i - 2] * c[i - 2] + shift_reg[i - 3] * c[i - 3];  
}
```

# Оптимизации | Конвейеризация циклов



# Интерфейсы. Допустимые типы аргументов

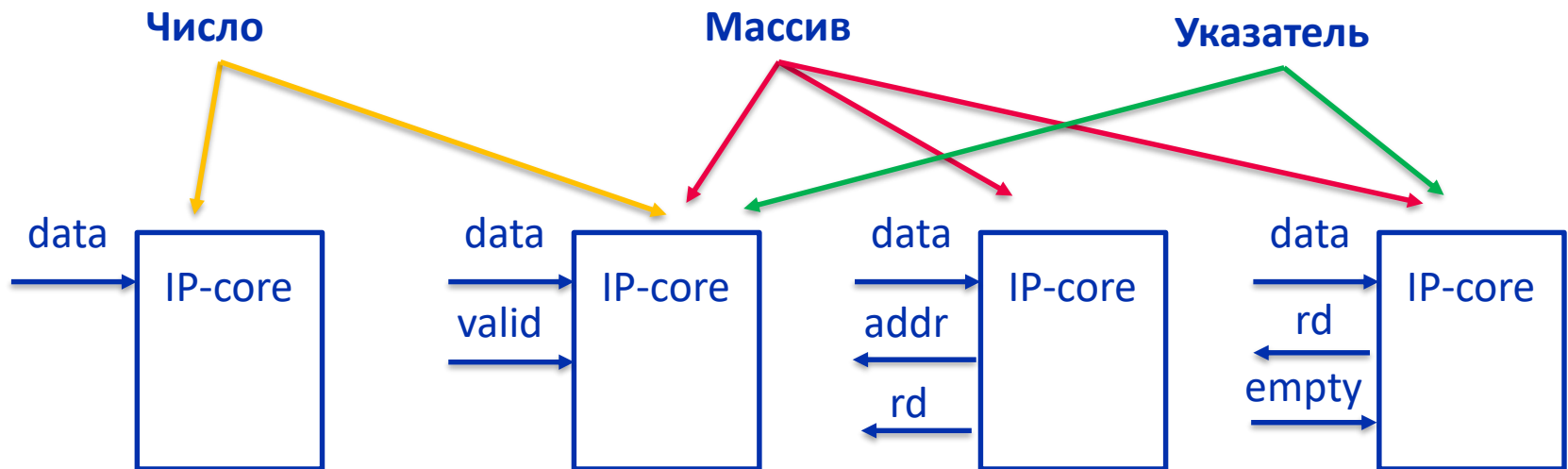
Допустимые типы аргументов:

- Число (int data)
- Массив (int data[N])
- Указатель (int \* data)

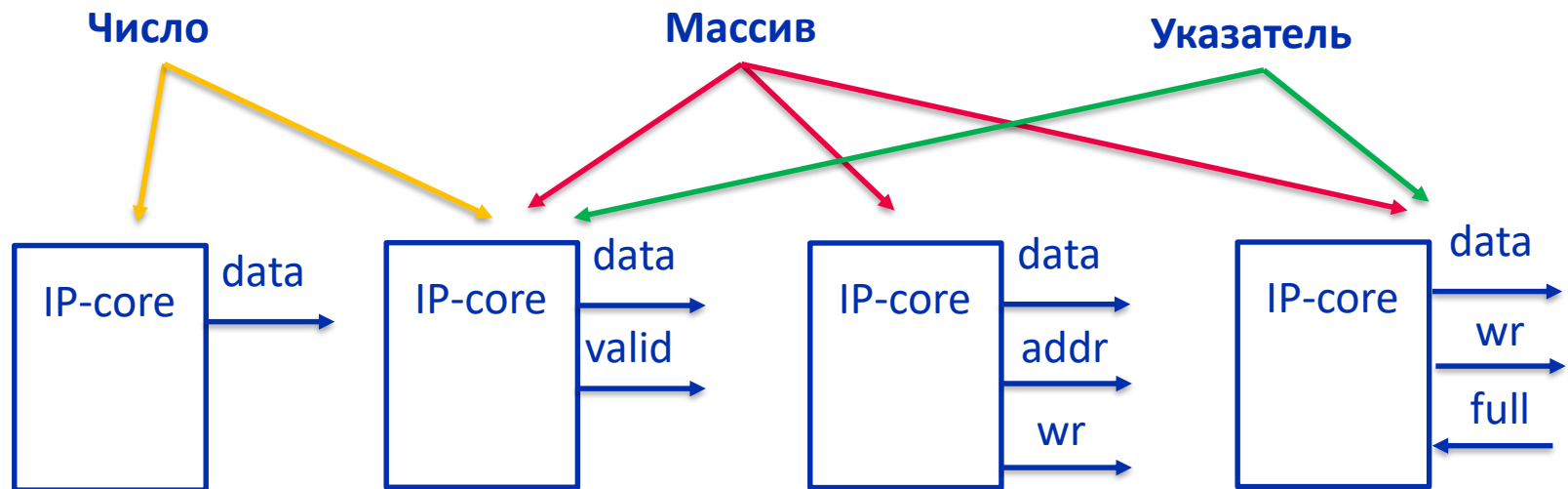
Возможные интерфейсы:

- Без логики управления (непосредственная передача значения)
- Интерфейс с протоком рукопожатия (handshake)
- Интерфейс памяти
- Интерфейс очереди
- Стандартизированный интерфейс коммуникационной шины (AXI, Wishbone и др.)

# Отображение аргументов на входные порты

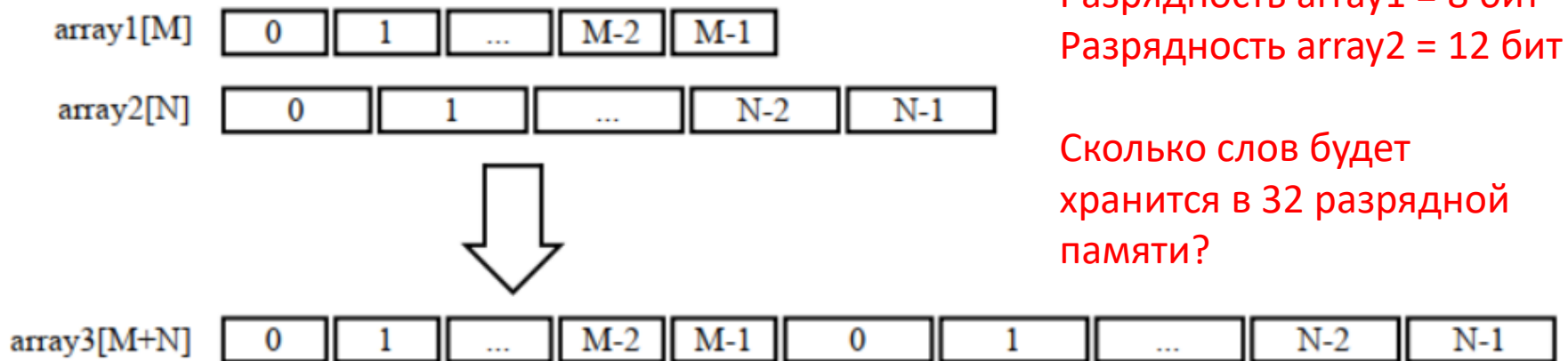


## Отображение аргументов на выходные порты



## Оптимизация массивов. Склеивание массивов

### Горизонтальное отображение



Примем:

$M = 5$ ,

$N = 8$ ,

Разрядность array1 = 8 бит

Разрядность array2 = 12 бит

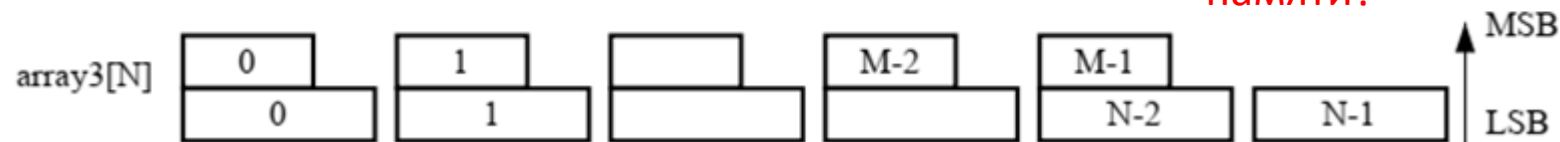
Сколько слов будет  
хранится в 32 разрядной  
памяти?

Используется для уменьшения количества **блоков** блочной памяти. Актуально при наличии множества мелких массивов.



# Оптимизация массивов. Склеивание массивов

## Вертикальное отображение



Примем:

$M = 5$ ,

$N = 8$ ,

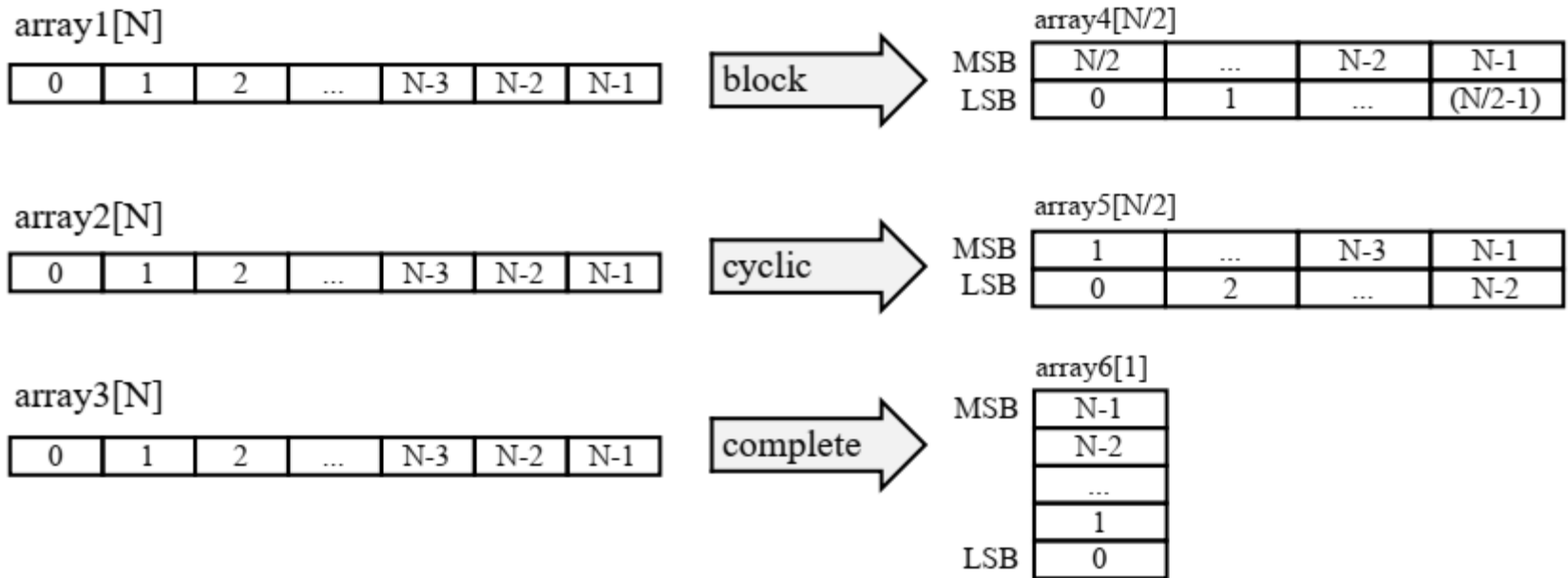
Разрядность array1 = 8 бит

Разрядность array2 = 12 бит

Сколько слов будет хранится в 32 разрядной памяти?

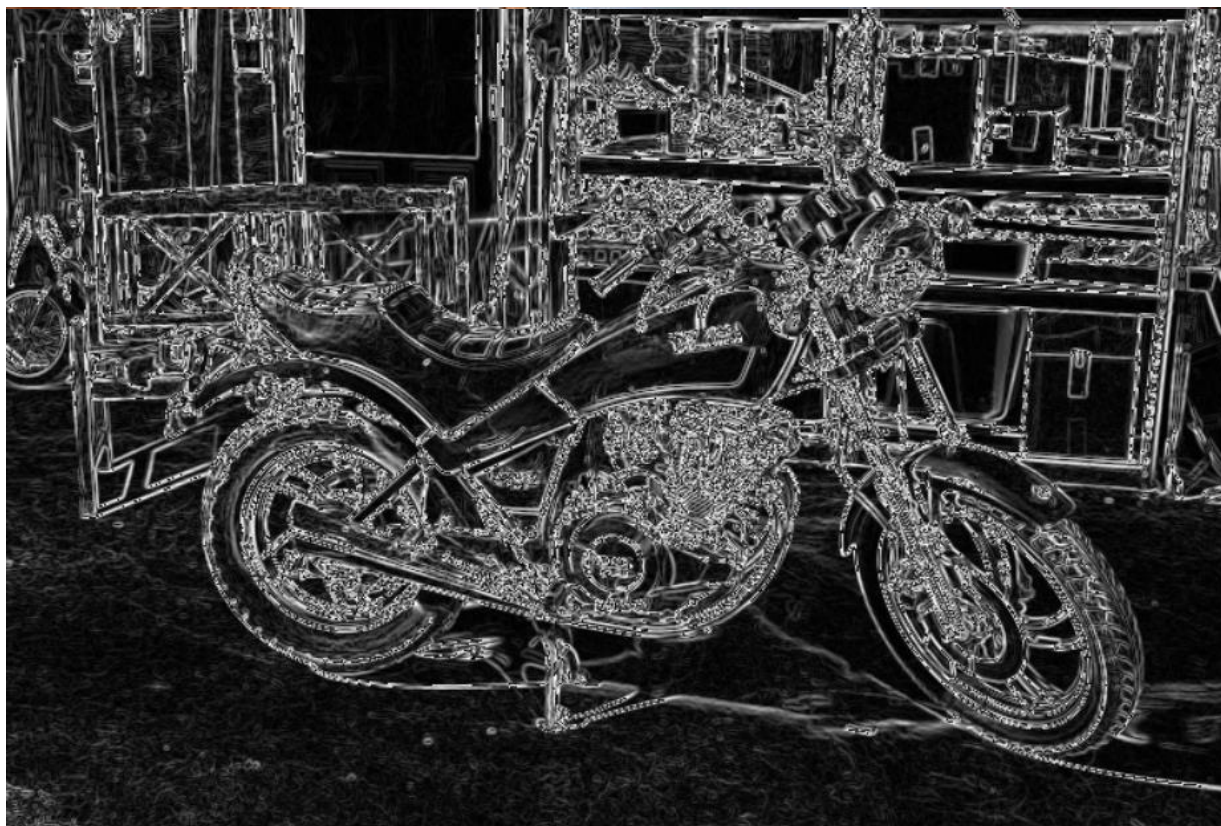
Используется для уменьшения количества слов блочной памяти. Увеличивается разрядность слова, но уменьшается количество слов.

# Оптимизация массивов. Переупорядочивание содержимого (reshaping)



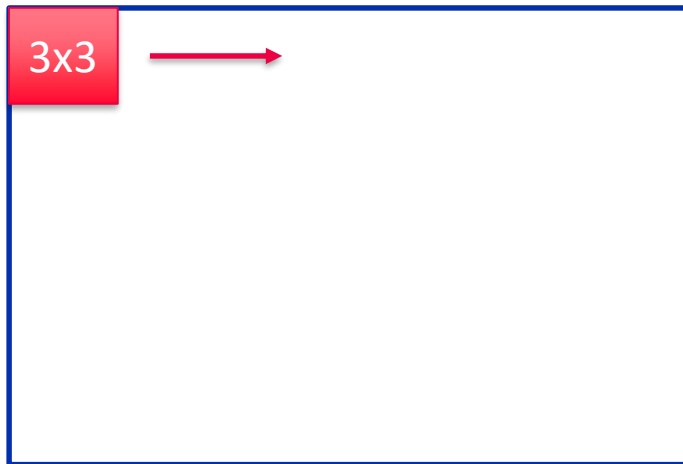
Используется для оптимизации скорости доступа к разным элементам массива.

## Фильтр Собеля



Используется для определения границ на изображении

# Фильтр Собеля | Алгоритм



```
int gx = {-1, 0, 1,  
          -2, 0, 2,  
          -1, 0, 1 };  
int gy = {-1, -2, -1,  
          0,  0,  0,  
          1,  2,  1 };
```

```
int x = 0, y = 0;
```

```
for(int i = 0; i < 9; i++){  
    x += in[i]*gx[i];  
    y += in[i]*gy[i];  
}
```

$$out = \sqrt{x^2 + y^2}$$

## Фильтр Собеля | Вопросы и задачи

- Что является входными и выходными данными? Какой будет формат их представления?
- Проведите синтез аппаратного блока фильтра Собеля без развертки циклов и без конвейеризации вычислений.
- За сколько тактов выполнится свертка участка изображения 3x3 пикселя?
- Сколько состояний будет у управляющего автомата?
- Какие ресурсы будут использованы при реализации аппаратного блока фильтра Собеля?



УНИВЕРСИТЕТ ИТМО

**Спасибо за внимание!**

[sergei\\_bykovskii@itmo.ru](mailto:sergei_bykovskii@itmo.ru)

[142291@itmo.ru](mailto:142291@itmo.ru)