

## **SIMD инструкции (продолжение)**

# Редукция

- Редукция – вычисления скалярного значения по вектору значений
- Примеры: вычисление суммы, минимума, максимума, произведения, и т.п. элементов одномерного массива

# Редукция

- Общий алгоритм редукции на SIMD архитектурах:

1. Разбить данные на части, для которых значение может быть вычислено независимо (вертикально)

- количество частей = длина вектора, обрабатываемого SIMD инструкцией
- вычислить вектор частичных значений

# Редукция

2. Произвести редукцию полученного вектора. Варианты:

- а) последовательно. Число шагов = длина
- б) параллельно (горизонтальными инструкциями), за каждый шаг сокращая его длину в 2 раза. Число шагов =  $\log_2(\text{длина})$

Зори СА "АР", 2019

# данные 1

[illegible]

\* \* \* \* \*

# данные 1

[illegible]

=====

# промежуточные результаты

[illegible]

\* \* \* \* \*

## данные 2

[illegible]

=====

## промежуточные результаты

[illegible]

\* \* \* \* \*

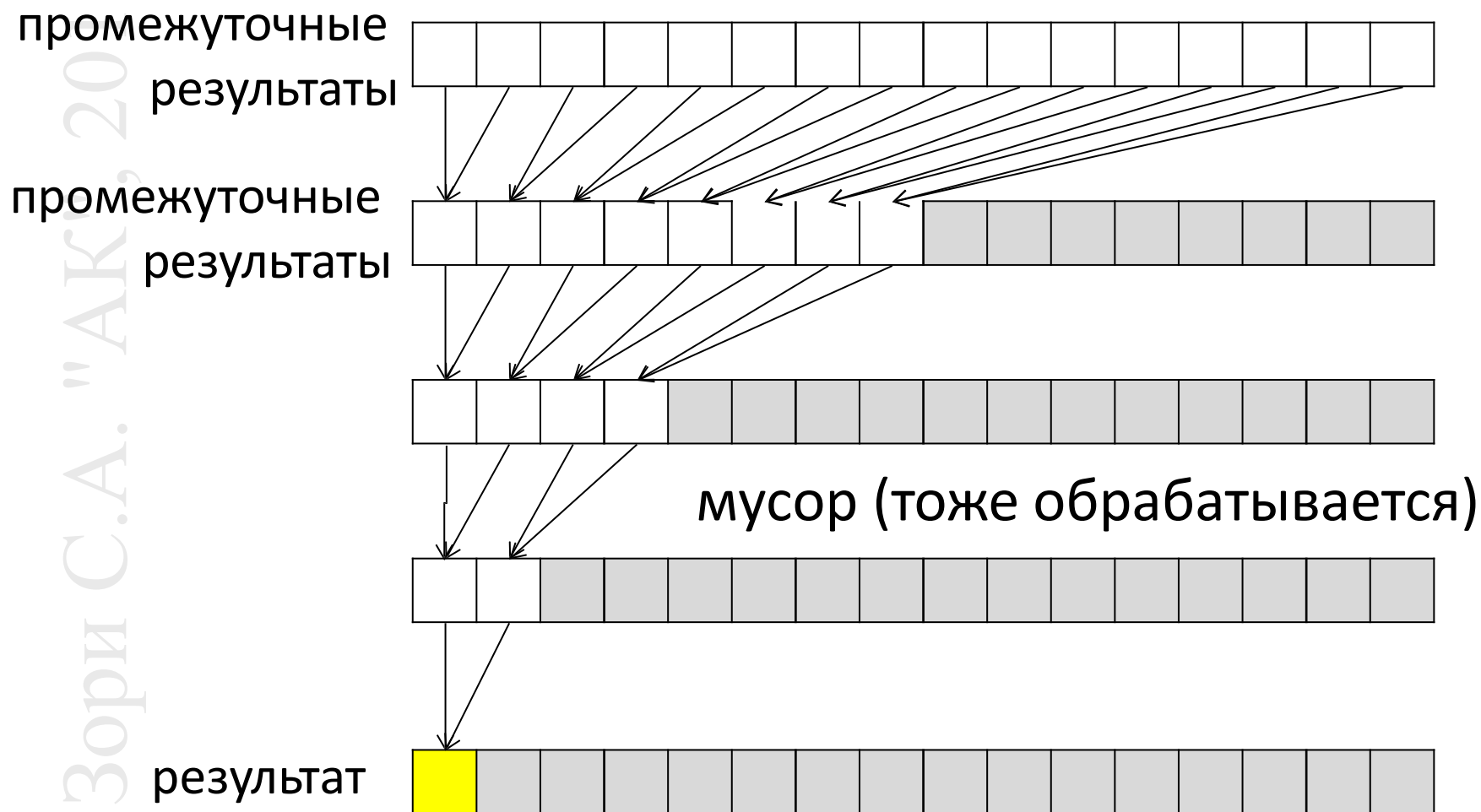
...

## промежуточные результаты

[illegible]

=====

# Редукция – горизонтальные вычисления



# Пример

- Вычислить сумму  $a_i \dots$

```
xor      xmm1, xmm1      ; sum = 0
fori:
    movaps  xmm0, a[esi]
; вычисляем параллельно 4 суммы s3, s2, s1, s0
    addps   xmm1, xmm0
    add esi, 16
loop
```

# Пример (продолжение)

`; xmm1 = s3, s2, s1, s0`

**`haddps xmm1, xmm1`**

`; xmm1 = s3+s2, s1+s0, s3+s2, s1+s0`

**`haddps xmm1, xmm1`**

`; xmm1 = ..., ..., ..., s3+s2+s1+s0`

**`movss result, xmm1`**