

Университет ИТМО

**Системы на кристалле**  
**Лабораторная работа №1**  
Вариант 3

Выполнили: Калугина Марина  
Саржевский Иван  
Группа: Р3402

г. Санкт-Петербург

2020 г.

# Задание

Реализовать КИХ фильтр цифрового сигнала.

## Дизайн системы

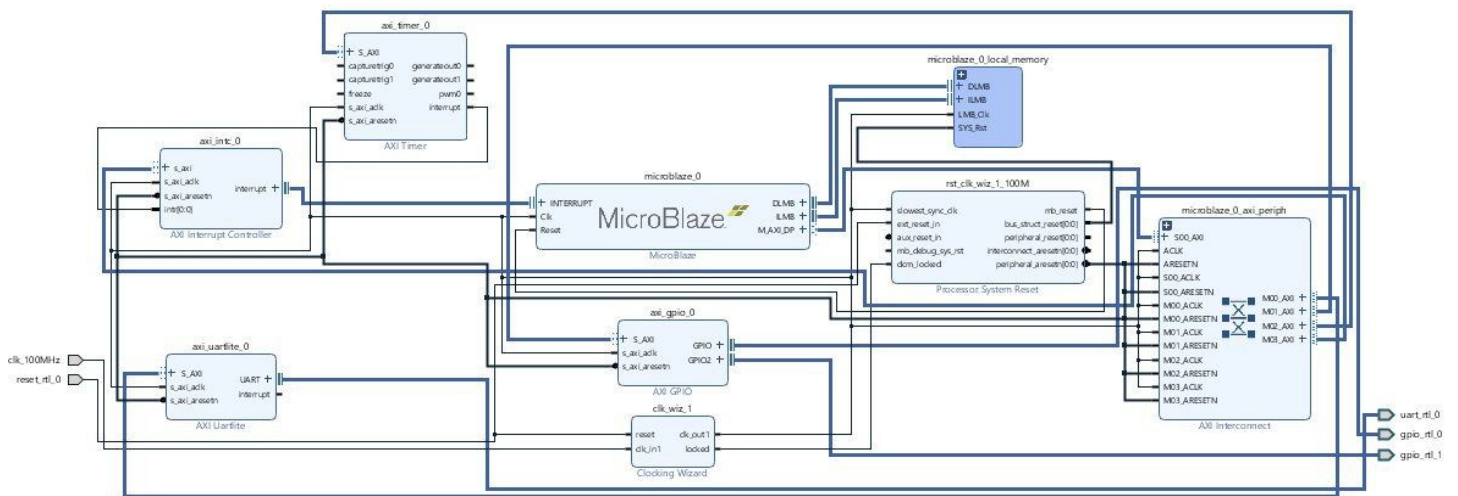


Рисунок 1

Для обработки прерываний был использован таймер и обработчик прерываний. Также были использованы: soft-процессор MicroBlaze, контроллер асинхронного передатчика UART, контроллер дискретных портов ввода-вывода GPIO, коммуникационную матрицу шины AXI.

Схема описанных выше элементов изображена на рисунке 1.

# Характеристики созданной системы

Число занимаемых ресурсов ПЛИС (XC7A100T-1CSG324C)

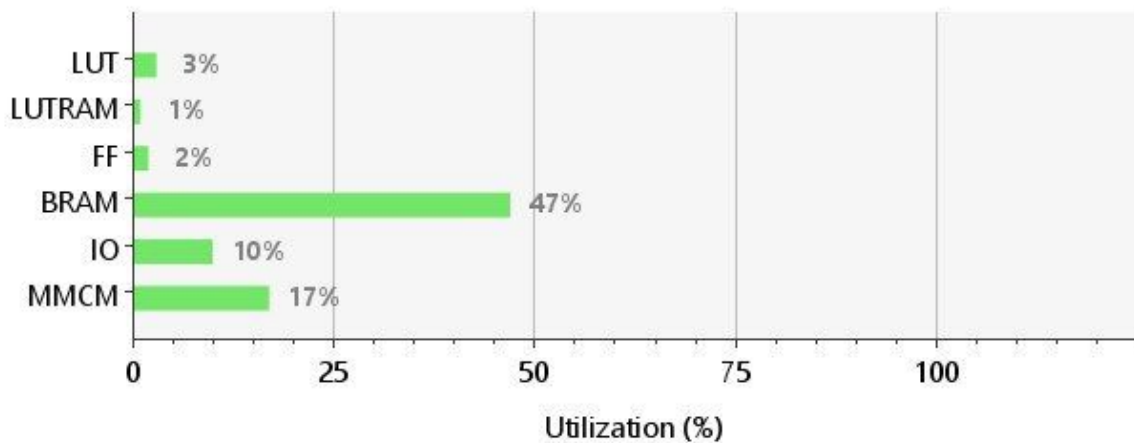


Рисунок 2

Resource	Utilization	Available	Utilization %
LUT	1782	63400	2.81
LUTRAM	160	19000	0.84
FF	1977	126800	1.56
BRAM	64	135	47.41
IO	21	210	10.00
MMCM	1	6	16.67

Рисунок 3

## Вывод программы

Вывод программы представлен на рисунке 4.

Первые 10 чисел примера программы - входной массив. При дефолтном значении  $p = 3$  результат программы представлен в виде массива: 7, 4, 4, 4, 3, 3, 4, 5.

Следующая строка ввода -- смена Р. Смена Р осуществляется при изменении тумблера J15. Значение Р было изменено на 10 и следующие 10 чисел -- новый массив входных данных. Результатом стал массив из одного элемента - 5.

13  
5  
3  
4  
6  
2  
3  
5  
5  
5  
—  
7  
4  
4  
4  
3  
3  
4  
5  
—  
10  
13  
5  
3  
4  
6  
2  
3  
5  
5  
5  
—  
5  
—

Рисунок 4

## Зависимость времени исполнения от варьируемого параметра

[p = 3]::22us  
[p = 4]::23us  
[p = 5]::24us  
[p = 6]::24us  
[p = 7]::21us  
[p = 8]::18us

[p = 9>::15us  
[p = 10>::11us

## Код

### Алгоритм

```
void fir_filter(u16 x[10], u16 y[8], u16 p)
{
    for (int n = p - 1, i = 0; n < 10; ++n, ++i) {
        for (int j = 0; j < p; ++j)
            y[i] += x[n - j];
        y[i] /= p;
    }
}
```

### Прерывания

Для вывода чисел раз в 1 секунду был применен механизм прерываний. Для этого были добавлены элементы AXI Timer и Interrupt Controller, вывод прерывания таймера был подключен на вход контроллера прерываний, который был подсоединен к процессору через шину прерываний.

Код обработчика прерываний:

```
void TimerCounterHandler(void *CallBackRef, u8 TmrCtrNumber) {
    if (XTmrCtr_IsExpired(&TimerCounterInst, 0)) {
        if (interrupt_flag == 0) return;
        Xil_Out16(ADDR_GPIO_BASE, output[current_idx++]);
        if (current_idx == 10 - p + 1)
        {
            current_idx = 0;
            interrupt_flag = 0;
        }
    }
}
```

Мы имеем специальный флаг, который позволяет нам игнорировать нежелательные прерывания, в основном коде мы просто выставляем его в ненулевое значение. После вывода необходимого количества чисел, он снова зануляется, блокируя все прерывания.

## Вывод

В ходе выполнения лабораторной работы были получены базовые навыки создания микропроцессорной системы на кристалле на базе сложнофункциональных блоков.