Министерство науки и высшего образования Российской Федерации

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
“НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО”

Факультет Программной инженерии и компьютерной техники

Отчет

По лабораторной работе № 2

По предмету: Системы на кристалле

Вариант 1

Студенты:

Андрейченко Леонид Вадимович

Степанов Михаил Алексеевич

Группа P34301

Преподаватель:

Быковский Сергей Вячеславович

Санкт-Петербург

2023

Цель работы

Получить базовые навыки использования средств высокоуровневого синтеза в процессе проектирования СнК.

Задание

1. Спроектировать и описать функциональность аппаратного ускорителя для алгоритма из лабораторной работы №1 на языках C/C++, пригодную для синтеза в аппаратный блок.

2. Провести синтез аппаратного ускорителя.

3. Разработать тестовое окружение для проверки функциональности синтезированного аппаратного ускорителя.

4. Оценить следующие характеристики:

    4.1. Время выполнения алгоритма при частоте тактового сигнала в 100 МГц.

    4.2. Число занимаемых ресурсов ПЛИС (XC7A100T-1CSG324C).

    4.3. Время и занимаемые ресурсы ПЛИС с использованием следующих оптимизаций: раскрутка циклов, конвейеризация циклов.

Выполнение

Разработанный алгоритм был перенесен в Vivado HLS, и произведен его синтез. Получившийся код:

```
--------------------------------------------------------------------------------
#include "fir.h"

int permutations[7];

int get_index(int raw, int val) {
	return !raw ? 0 : (raw & 1 ? val : get_index(raw / 2, val + 1));
}

void swap(int i, int j) {
	int temp = permutations[i];
	permutations[i] = permutations[j];
	permutations[j] = temp;
}

int next_set(int n) {
	int j = n - 2;
	while (j != -1 && permutations[j] >= permutations[j + 1]) j--;
	if (j == -1)
		return 0;
	int k = n - 1;
	while (permutations[j] >= permutations[k]) k--;
	swap(j, k);
	int l = j + 1, r = n - 1;
	while (l<r)
		swap(l++, r--);
	return 1;
}
```

```
int get_element(int i, int j, int exc, int matrix[7][7], int y[7]) {
	return j == exc ? y[i] : matrix[i][j];
}

int count_inversion(int n) {
	int sum = 0;
	for (int i = 0; i < n; i++) {
		for (int j = i + 1; j < n; j++) {
			if (permutations[i] > permutations[j]) sum++;
		}
	}
	return sum;
}

void clear_perm(){
	for(int i = 0; i < 7; i++)
		permutations[i] = i;
}

int eval_elem(int param, int n, int matrix[7][7], int y[7]) {
	int res = count_inversion(n) % 2 ? -1 : 1;
	for(int i = 0; i < n; i++) {
		res *= get_element(i, permutations[i], param, matrix, y);
	}
	return  res;
}

int calc_det(int param, int n, int matrix[7][7], int y[7]){
	clear_perm();
	int res = eval_elem(param, n, matrix, y);
	while(next_set(n)){
		res += eval_elem(param, n, matrix, y);
	}
	return res;
}

void calc_result(int n, int matrix[7][7], int y[7], int result[7]){
	int main_delta = calc_det(n + 1, n, matrix, y);
	for (int i = 0; i < n; i++) {
		int delta = calc_det(i, n, matrix, y);
		result[i] = delta / main_delta;
	}
}
```

Далее были написаны тесты для проверки корректности работы получившегося аппаратного блока. Результаты тестирования:

```
   Generating csim.exe
----- START ALL TEST -----


----- START TEST 3 -----
STEP 0 CORRECT
STEP 1 CORRECT
STEP 2 CORRECT
----- END TEST 3 -----


----- START TEST 4 -----
STEP 0 CORRECT
STEP 1 CORRECT
STEP 2 CORRECT
STEP 3 CORRECT
----- END TEST 4 -----


----- START TEST 5 -----
STEP 0 CORRECT
STEP 1 CORRECT
STEP 2 CORRECT
STEP 3 CORRECT
STEP 4 CORRECT
----- END TEST 5 -----


----- END ALL TEST -----INFO: [SIM 211-1] CSim done with 0 er
INFO: [SIM 211-3] *************** CSIM finish ***************
Finished C simulation.
```

Стандартная реализация

Clock	Target	Estimated	Uncertainty
ap_clk	10.00	8.470	1.25

Name	BRAM_18K	DSP48E	FF	LUT	URAM
DSP	-	-	-	-	-
Expression	-	-	0	95	-
FIFO	-	-	-	-	-
Instance	2	3	1222	1455	0
Memory	-	-	-	-	-
Multiplexer	-	-	-	219	-
Register	-	-	197	-	-
Total	2	3	1419	1769	0
Available	270	240	126800	63400	0
Utilization (%)	~0	1	1	2	0

Время выполнения алгоритма 3 =12.3 us, 4 = 74 us, 5 = 554 us

---

pipeline

Name	BRAM_18K	DSP48E	FF	LUT	URAM
DSP	-	-	-	-	-
Expression	-	-	0	173	-
FIFO	-	-	-	-	-
Instance	-	3	2979	2692	-
Memory	1	-	0	0	0
Multiplexer	-	-	-	534	-
Register	-	-	246	-	-
Total	1	3	3225	3399	0
Available	270	240	126800	63400	0
Utilization (%)	~0	1	2	5	0

Clock	Target	Estimated	Uncertainty
ap_clk	10.00	8.470	1.25

Время выполнения алгоритма 3 = 5.5 us, 4 = 24 us, 5 = 172 us

---

unroll

Name	BRAM_18K	DSP48E	FF	LUT	URAM
DSP	-	-	-	-	-
Expression	-	-	0	173	-
FIFO	-	-	-	-	-
Instance	-	3	1090	1192	-
Memory	1	-	0	0	0
Multiplexer	-	-	-	534	-
Register	-	-	246	-	-
Total	1	3	1336	1899	0
Available	270	240	126800	63400	0
Utilization (%)	~0	1	1	2	0

Clock	Target	Estimated	Uncertainty
ap_clk	10.00	8.470	1.25

Время выполнения алгоритма 3 = 8us, 4 = 24 us, 5 = 163 us

Выводы

Самая быстрая реализация – разворачивание цикла.