

Санкт-Петербургский Политехнический Университет Петра Великого
Институт Компьютерных наук и технологий
Кафедра компьютерных систем и программных технологий

Лабораторная работа 12 Задание 1

Предмет: «Проектирование реконфигурируемых гибридных вычислительных систем»

Тема: «Inline»

Студент: Ерниязов Т.Е.
Гр. № 3540901/81502

Преподаватель: Антонов А.П.

Санкт-Петербург
2019

Оглавление

| | |
|-----------------------------------|----|
| 1. Задание | 4 |
| 2. Исходный код | 6 |
| 3. Моделирование | 7 |
| 4. Первое решение | 7 |
| 4.1. Директивы | 7 |
| 4.2. Синтез | 7 |
| 4.3. Использование ресурсов | 8 |
| 4.4. C/RTL Моделирование | 9 |
| 5. Второе решение | 9 |
| 5.1. Директивы | 9 |
| 5.2. Синтез | 9 |
| 5.3. Использование ресурсов | 10 |
| 5.4. C/RTL Моделирование | 11 |
| 6. Выводы | 11 |

1. Задание

- Создать проект lab12_1
- Микросхема: ха7a12tcsg325-1q
- Создать иерархическую функцию ,

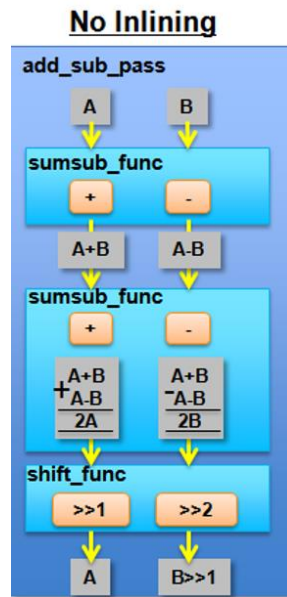
```
int sumsub_func(int *in1, int *in2, int *outSum, int *outSub) {
    *outSum = *in1 + *in2;
    *outSub = *in1 - *in2;
}

int shift_func(int *in1, int *in2, int *outA, int *outB) {
    *outA = *in1 >> 1;
    *outB = *in2 >> 2;
}

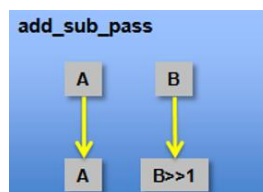
void add_sub_pass(int A, int B, int *C, int *D) {
    int apb, amb;
    int a2, b2;

    sumsub_func(&A,&B,&apb,&amb);
    sumsub_func(&apb,&amb,&a2,&b2);
    shift_func(&a2,&b2,C,D);
}
```

- Создать тест lab12_1_test.c для проверки функции. Осуществить моделирование (с выводом результатов в консоль)
- Исследование:
- Solution_1a
 - задать: clock period 10; clock_uncertainty 0.1
 - установить реализацию ПО УМОЛЧАНИЮ
 - осуществить синтез для:
 - привести в отчете:
 - performance estimates=>summary (timing, latency)
 - utilization estimates=>summary
 - performance Profile
 - Resource profile
 - scheduler viewer (выполнить Zoom to Fit)
 - На скриншоте показать Latency
 - На скриншоте показать Initiation Interval
 - resource viewer (выполнить Zoom to Fit)
 - На скриншоте показать Latency
 - На скриншоте показать Initiation Interval
 - Выполнить cosimulation и привести временную диаграмму
 - Убедиться в том, что требуется 2 сумматора и 2 вычитателя



- Solution_2a
 - задать: clock period 10; clock_uncertainty 0.1
 - установить реализацию Inlining
 - осуществить синтез
 - привести в отчете:
 - performance estimates=>summary (timing, latency)
 - utilization estimates=>summary
 - performance Profile
 - Resource profile
 - scheduler viewer (выполнить Zoom to Fit)
 - На скриншоте показать Latency
 - На скриншоте показать Initiation Interval
 - resource viewer (выполнить Zoom to Fit)
 - На скриншоте показать Latency
 - На скриншоте показать Initiation Interval
 - Выполнить cosimulation и привести временную диаграмму
 - Убедиться в том, что для реализации не требуется ресурсов.



- Сравнить два решения (solution_1a и solution_2a) и сделать выводы

2. Исходный код

```
1 void sumsub_func(int *in1, int *in2, int* outSum, int* outSub) {
2     *outSum = *in1 + *in2;
3     *outSub = *in1 - *in2;
4 }
5
6 void shift_func(int* in1, int* in2, int* outA, int* outB) {
7     *outA = *in1 >> 1;
8     *outB = *in2 >> 2;
9 }
10
11 void add_sub_pass(int A, int B, int* C, int* D) {
12     int apb, amb;
13     int a2, b2;
14
15     sumsub_func(&A, &B, &apb, &amb);
16     sumsub_func(&apb, &amb, &a2, &b2);
17     shift_func(&a2, &b2, C, D);
18 }
```

Рис. 2.1. Source code

```
1 #define N 16
2
3 int main() {
4     int A, B, C, D, C_expected, D_expected;
5     int pass = 1;
6     for (int i = 0; i < N; i++) {
7         A = (i * 123 - 16) / 7;
8         B = (i * A - 11) / 3;
9         C_expected = ((A - B) + (A + B)) >> 1;
10        D_expected = ((A + B) - (A - B)) >> 2;
11        add_sub_pass(A, B, &C, &D);
12        printf("A=%d, B=%d, C=%d, C_expected=%d, D=%d, D_expected=%d\n",
13              A, B, C, C_expected, D, D_expected);
14        if (C != C_expected || D != D_expected) {
15            pass = 0;
16        }
17    }
18    if (pass) {
19        printf("_____Pass!_____\\n");
20    } else {
21        printf("_____Fail!_____\\n");
22    }
23    return 0;
24 }
```

Рис. 2.2. Test code

3. Моделирование

По результатам моделирования видно, что устройство работает

```
Vivado HLS Console
INFO: [HLS 200-10] Running 'C:/Xilinx/Vivado/2019.2/bin/unwrapped/win64.o/
INFO: [HLS 200-10] For user 'Misha' on host 'mikhail' (Windows NT_amd64 ve
INFO: [HLS 200-10] In directory 'C:/Users/Misha/Desktop/university/___maga/
INFO: [APCC 202-3] Tmp directory is apcc_db
INFO: [APCC 202-1] APCC is done.
Generating csim.exe
A=-2, B=-3, C=-2, C_expected=-2, D=-2, D_expected=-2
A=15, B=1, C=15, C_expected=15, D=0, D_expected=0
A=32, B=17, C=32, C_expected=32, D=8, D_expected=8
A=50, B=46, C=50, C_expected=50, D=23, D_expected=23
A=68, B=87, C=68, C_expected=68, D=43, D_expected=43
A=85, B=138, C=85, C_expected=85, D=69, D_expected=69
A=103, B=202, C=103, C_expected=103, D=101, D_expected=101
A=120, B=276, C=120, C_expected=120, D=138, D_expected=138
A=138, B=364, C=138, C_expected=138, D=182, D_expected=182
A=155, B=461, C=155, C_expected=155, D=230, D_expected=230
A=173, B=573, C=173, C_expected=173, D=286, D_expected=286
A=191, B=696, C=191, C_expected=191, D=348, D_expected=348
A=208, B=828, C=208, C_expected=208, D=414, D_expected=414
A=226, B=975, C=226, C_expected=226, D=487, D_expected=487
A=243, B=1130, C=243, C_expected=243, D=565, D_expected=565
A=261, B=1301, C=261, C_expected=261, D=650, D_expected=650
Pass!
INFO: [SIM 211-1] CSim done with 0 errors.
INFO: [SIM 211-3] ***** CSIM finish *****
Finished C simulation.
```

Рис. 3. Successful result of modeling

4. Первое решение

4.1. Директивы

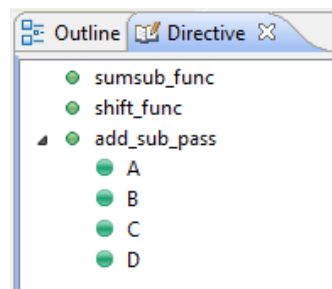


Рис. 4.1. Directives

4.2. Синтез

Performance Estimates

Timing

Summary

| Clock | Target | Estimated | Uncertainty |
|--------|----------|-----------|-------------|
| ap_clk | 10.00 ns | 0 ns | 0.10 ns |

Latency

Summary

| Latency (cycles) | | Latency (absolute) | | Interval (cycles) | | |
|------------------|-----|--------------------|------|-------------------|-----|------|
| min | max | min | max | min | max | Type |
| 0 | 0 | 0 ns | 0 ns | 0 | 0 | none |

Рис. 4.2. Performance estimates

Полученная величина задержки укладывается в заданное значение.

4.3. Использование ресурсов

| Utilization Estimates | | | | | |
|-----------------------|----------|--------|-------|------|------|
| Summary | | | | | |
| Name | BRAM_18K | DSP48E | FF | LUT | URAM |
| DSP | - | - | - | - | - |
| Expression | - | - | - | - | - |
| FIFO | - | - | - | - | - |
| Instance | - | - | - | - | - |
| Memory | - | - | - | - | - |
| Multiplexer | - | - | - | - | - |
| Register | - | - | - | - | - |
| Total | 0 | 0 | 0 | 0 | 0 |
| Available | 40 | 40 | 16000 | 8000 | 0 |
| Utilization (%) | 0 | 0 | 0 | 0 | 0 |

Рис. 4.3.1. Utilization estimates

| Performance Profile | | | | | |
|---------------------|-----------|---------|-------------------|---------------------|------------|
| | Pipelined | Latency | Iteration Latency | Initiation Interval | Trip count |
| add_sub_pass | - | 0 | - | 1 | - |

Рис. 4.3.2. Performance profile

| Resource Profile | | | | | | | |
|------------------|------|-----|----|-----|---------|---------|---------|
| | BRAM | DSP | FF | LUT | Bits P0 | Bits P1 | Bits P2 |
| add_sub_pass | 0 | 0 | 0 | 0 | | | |
| I/O Ports(4) | | | | | 128 | | |
| Instances(0) | 0 | 0 | 0 | 0 | | | |
| Memories(0) | 0 | | 0 | 0 | 0 | | 0 |
| Expressions(0) | 0 | 0 | 0 | 0 | 0 | 0 | |
| Registers(0) | | | 0 | 0 | | | |
| Channels(0) | 0 | | 0 | 0 | | | 0 |
| Multiplexers(0) | 0 | | 0 | 0 | | | 0 |
| DSP(0) | | 0 | | | | | |

Рис. 4.3.3 Resource profile

| Operation\Control Step | |
|------------------------|---|
| B_read(read) | |
| A_read(read) | |
| C_write_In17(write) | |
| D_write_In17(write) | |
| | 0 |

Рис. 4.3.4. Operation\Control Step

| Operation\Control Step | |
|------------------------|-------|
| [+] I/O Ports | |
| A | |
| C | |
| D | |
| B | |
| | 0 |
| | read |
| | write |
| | write |
| | read |

Рис. 4.3.5. Operation\Control Step

4.4. C/RTL Моделирование

По результат моделирование видно, что среда разработки выполнило inlining, хотя это не было указано явно.

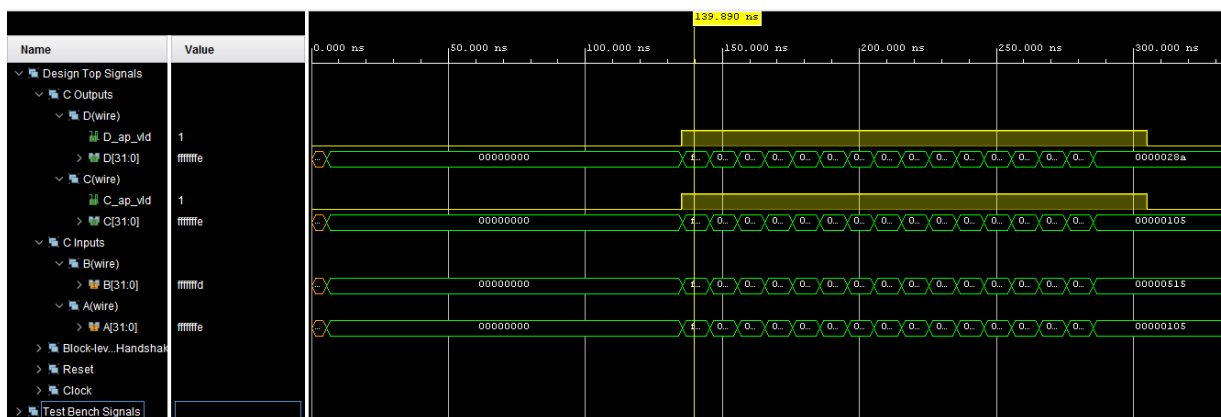


Рис. 4.4. Modeling result

5. Второе решение

5.1. Директивы

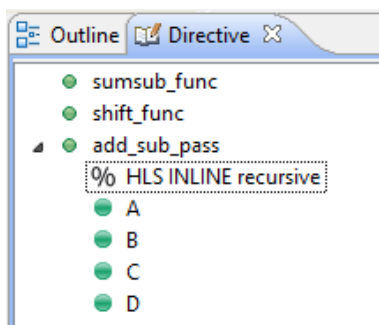


Рис.5.1 Directive

5.2. Синтез

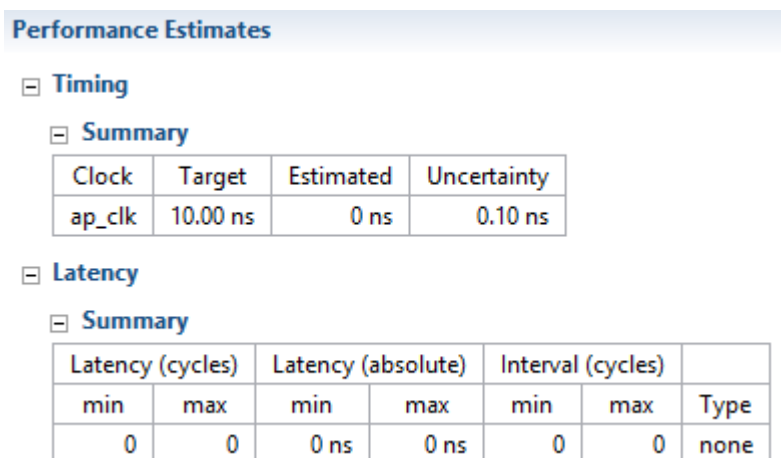


Рис. 5.2. Performance estimates

5.3. Использование ресурсов

| Utilization Estimates | | | | | |
|-----------------------|----------|--------|-------|------|------|
| Summary | | | | | |
| Name | BRAM_18K | DSP48E | FF | LUT | URAM |
| DSP | - | - | - | - | - |
| Expression | - | - | - | - | - |
| FIFO | - | - | - | - | - |
| Instance | - | - | - | - | - |
| Memory | - | - | - | - | - |
| Multiplexer | - | - | - | - | - |
| Register | - | - | - | - | - |
| Total | 0 | 0 | 0 | 0 | 0 |
| Available | 40 | 40 | 16000 | 8000 | 0 |
| Utilization (%) | 0 | 0 | 0 | 0 | 0 |

Рис. 5.3.1. Utilization estimates

| Performance Profile | | | | | |
|---------------------|-----------|---------|-------------------|---------------------|------------|
| | Pipelined | Latency | Iteration Latency | Initiation Interval | Trip count |
| ● add_sub_pass | - | 0 | - | 1 | - |

Рис. 5.3.2. Performance profile

| Resource Profile | | | | | | | | |
|------------------|------|-----|----|-----|---------|---------|---------|-------------|
| | BRAM | DSP | FF | LUT | Bits P0 | Bits P1 | Bits P2 | Banks/Depth |
| ● add_sub_pass | 0 | 0 | 0 | 0 | | | | |
| ▶ I/O Ports(4) | | | | | 128 | | | |
| Instances(0) | 0 | 0 | 0 | 0 | | | | |
| Memories(0) | 0 | 0 | 0 | 0 | | | | 0 |
| Expressions(0) | 0 | 0 | 0 | 0 | 0 | 0 | | |
| Registers(0) | | | 0 | | 0 | | | |
| Channels(0) | 0 | | 0 | 0 | 0 | | | 0 |
| Multiplexers(0) | 0 | | 0 | 0 | 0 | | | 0 |
| DSP(0) | | 0 | | | | | | |

Рис. 5.3.3 Resource profile

| Operation\Control Step | |
|------------------------|---|
| B_read(read) | 0 |
| A_read(read) | |
| C_write_In17(write) | |
| D_write_In17(write) | |

Рис. 5.3.4. Operation\Control Step

| Operation\Control Step | |
|------------------------|-------|
| ▲ [+]I/O Ports | 0 |
| A | read |
| C | write |
| D | write |
| B | read |

Рис. 5.3.5. Operation\Control Step

5.4. C/RTL Моделирование

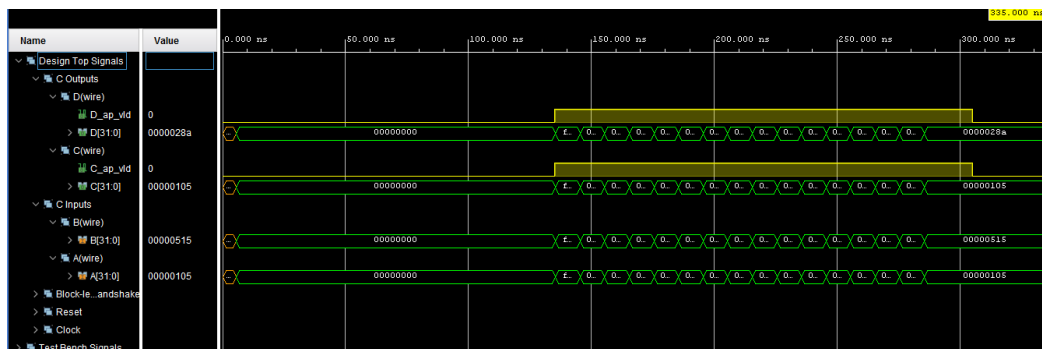


Рис. 5.4. Modeling result

Данное решение не отличается от первого, так как в предыдущем был применен `inlining` без явного указания, а во втором решение после явного указания – ничего не изменилось.

6. Выводы

В ходе выполнения работы не удалось сравнить производительность устройства после применения `inlining` и без применения, так как среда разработки применяет его, даже если это не указано, по причине того, что для данного устройства это однозначно приводит к улучшению характеристик.