Санкт-Петербургский политехнический университет Петра Великого
Институт компьютерных наук и технологий
Кафедра компьютерных систем и программных технологий

Лабораторная №12

Предмет: Проектирование реконфигурируемых гибридных вычислительных систем

Тема: Inline

Задание 1

Студенты:
Соболь В.
Темнова А.С.
Группа: 13541/3
Преподаватель:
Антонов А.П.

Санкт-Петербург
2019

Содержание

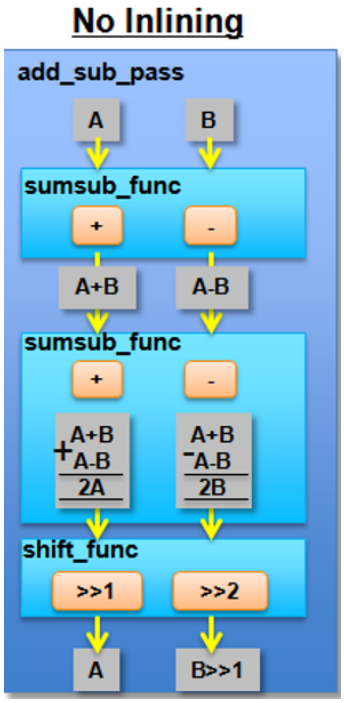
1. Задание

1. Создать проект lab12_1
2. Микросхема: xa7a12tcsg325-1q
3. 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);

}
```

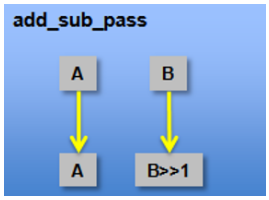
4. Создать тест lab12_1_test.c для проверки функции. Осуществить моделирование (с выводом результатов в консоль)
5. Исследование:
6. Solution_1a
   - задать: clock period 10; clock_uncertainty 0.1
   - установить реализацию ПО УМОЛЧАНИЮ
   - осуществить синтез для:
     - привести в отчете:
       * 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



7. 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.



8. Сравнить два решения (solution_1a и solution_2a) и сделать выводы

2. Исходный код

Ниже приведен исходный код устройства и теста.

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

Рис. 2.1. Исходный код устройства

```
#define N 16
int main()
{
  int A,B,C,D,C_Exp,D_Exp;
  int pass = 1;
  for(int i = 0;i < N; i++)
  {
    A = (i*123 - 16) / 7;
    B = (i*A - 11) / 3;
    C_Exp = ((A - B) + (A + B)) >> 1;
    D_Exp = ((A + B) - (A - B)) >> 2;
    add_sub_pass(A,B,&C,&D);
    printf("A = %d B = %d C = %d C_Exp = %d D = %d D_Exp = %d\n",A,B,C,C_Exp,D,
    ↪ D_Exp);
    if(C != C_Exp || D != D_Exp)
      pass = 0;
  }
  if(pass)
  {
    printf("----------Test passed----------\n");
  } else {
    printf("----------Test failed----------\n");
  }
  return 0;
}
```

Рис. 2.2. Исходный код теста

3. Скрипт

Ниже приводится скрипт, для автоматизации выполнения лабораторной работы.

```
open_project -reset lab12_1

add_files lab12_1.c
add_files -tb lab12_1_test.c
set_top add_sub_pass

set solutions [list 1a 2a]

foreach sol $solutions {
  open_solution solution_$sol -reset
  set_part {xa7a12tcsg325-1q}
  create_clock -period 10ns
  set_clock_uncertainty 0.1

  if {$sol == "2a"} {
    set_directive_inline -recursive add_sub_pass
  }

  csim_design
  csynth_design
  cosim_design -trace_level all
}

exit
```

Рис. 3.1. Скрипт

4. Моделирование

Ниже приведены результаты моделирования.

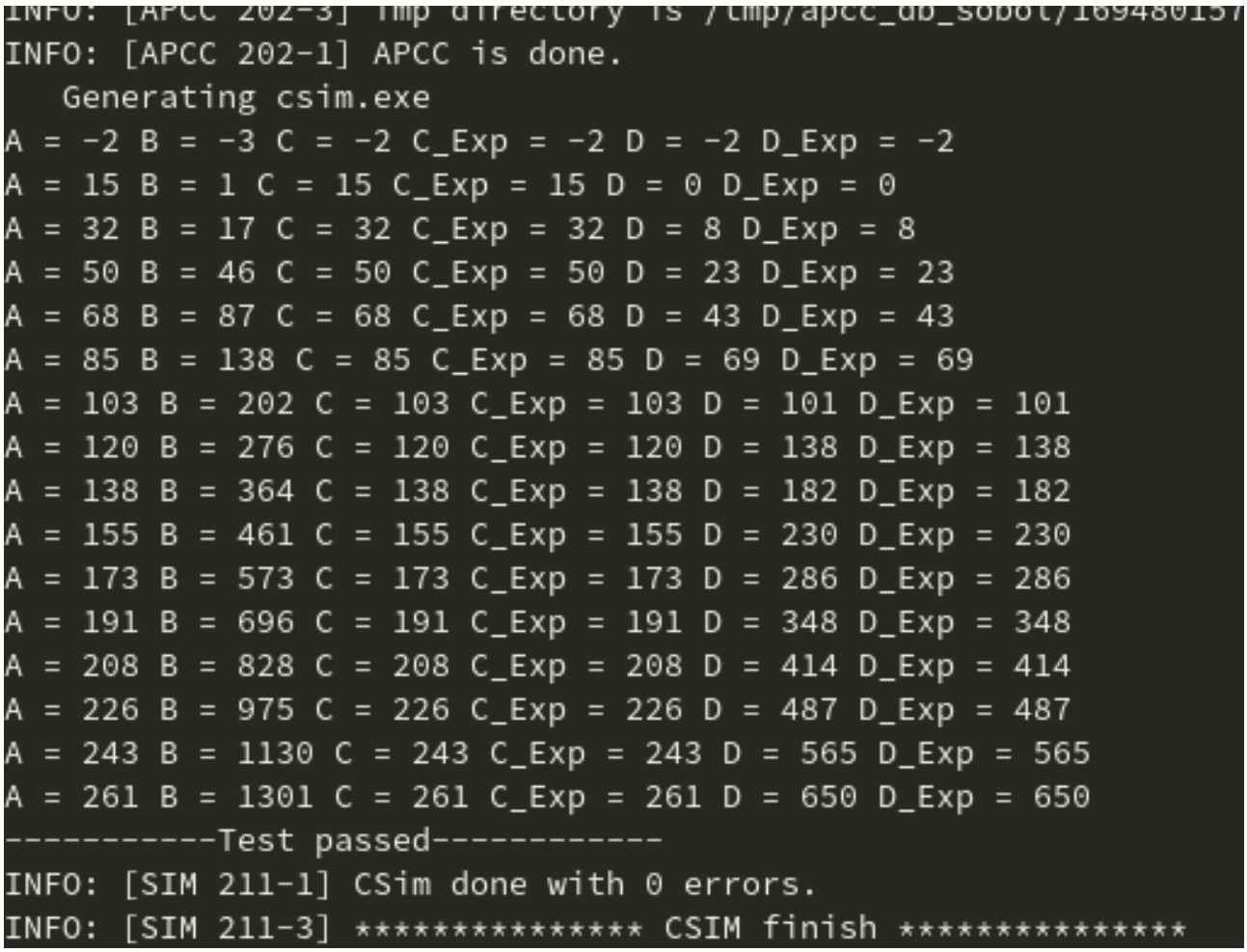


Рис. 4.1. Результаты моделирования

По результатам моделирования видно, что устройство работает корректно.

5. Решение 1a

5.1. Директивы

В данном решения были установлены директивы, приведённые ниже.

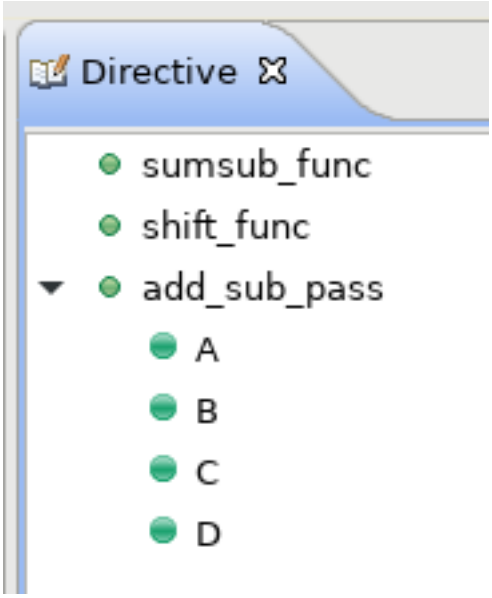


Рис. 5.1. Директивы

5.2. Синтез

По оценке производительности видно, что устройство соответствует заданным критериям.

Performance Estimates

Timing (ns)

Summary

Clock	Target	Estimated	Uncertainty
ap_clk	10.00	0.000	0.10

Latency (clock cycles)

Summary

Latency		Interval		
min	max	min	max	Type
0	0	0	0	none

Рис. 5.2. Performance estimates

Utilization Estimates

Summary

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

Рис. 5.3. Utilization estimates

Performance Profile | Resource Profile

	Pipelined	Latency	Iteration Latency	Initiation Interval	Trip count
add_sub_pass	-	0	-	1	-

Рис. 5.4. Performance profile

Рис. 5.5. Scheduler viewer

	Resource\Contro...	C0
1	⊟ I/O Ports	
2	A	read
3	B	read
4	C	write
5	D	write

Рис. 5.6. Resource viewer

5.3. C/RTL моделирование

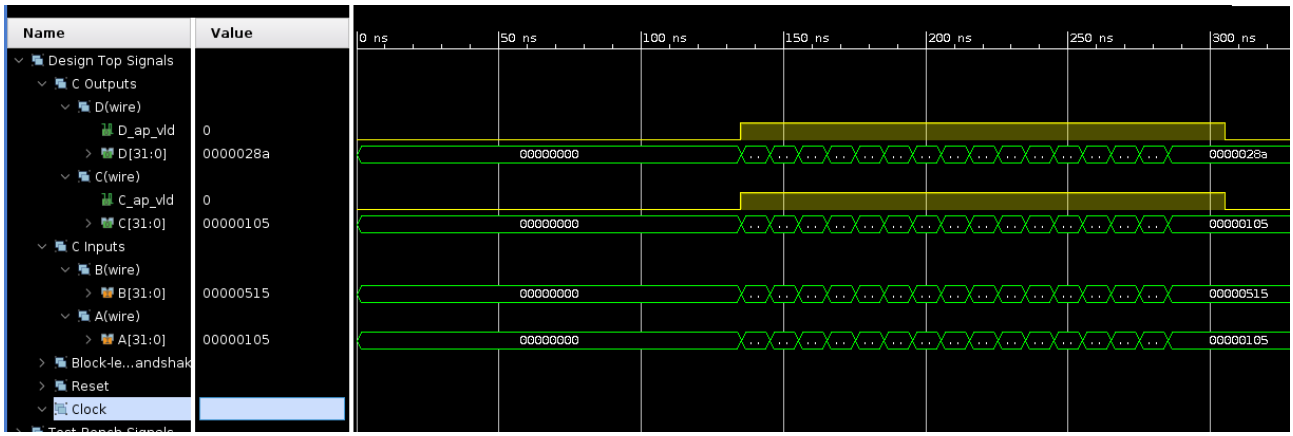


Рис. 5.7. Временная диаграмма

По результатам выше видно, что среда разработки выполнила inlining, хотя это и не было указано явно.

6. Решение 2a

6.1. Директивы

В данном решения были установлены директивы, приведённые ниже.

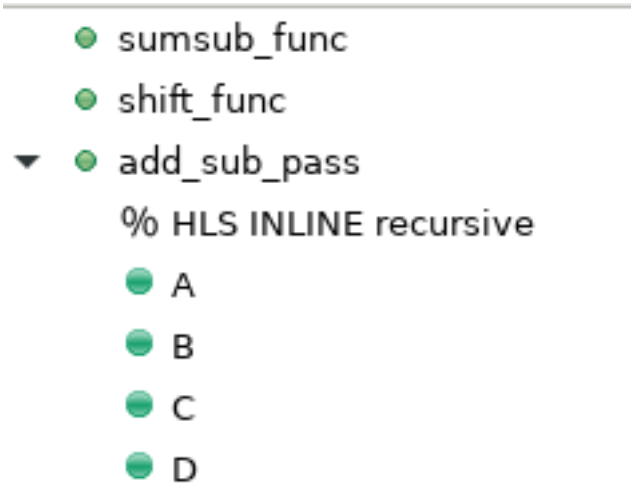


Рис. 6.1. Директивы

6.2. Синтез

Данное решение не отличается от предыдущего, так как в предыдущем, inlining был применён без явного указания, а в данном решении, после явного указания, ничего не поменялось.

7. Вывод

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