

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

Лабораторная работа 13 Задание 1

Предмет: «Проектирование реконфигурируемых гибридных вычислительных систем»

Тема: «Сравнение типов данных»

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

Преподаватель: Антонов А.П.

Санкт-Петербург
2019

Оглавление

1. Задание	4
2. Первое решение	6
2.1. Исходный код.....	6
2.2. Моделирование	7
2.3. Синтез	7
2.4. Использование ресурсов	7
3. Второе решение.....	9
3.1. Исходный код.....	9
3.2. Моделирование	10
3.3. Синтез	10
3.4. Использование ресурсов	10
4. Третье решение	12
4.1. Исходный код.....	12
4.2. Моделирование	13
4.3. Синтез	13
4.4. Использование ресурсов	13
5. Выводы	15

1. Задание

- Создать проект lab13_1
- Микросхема: xa7a12tcs325-1q
- В папке Source имеется 3 папки с описанием одной функции, но разными типами данных
- Ознакомиться с описаниями функций
- Ознакомиться с тестами
- Исследование:
- Solution_1a – для функции types_standard
 - Осуществить моделирование (с выводом результатов в консоль)
 - 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
- Solution_2a – для функции types_float_double
 - Осуществить моделирование (с выводом результатов в консоль)
 - задать: clock period 10; clock_uncertainty 0.1
 - установить реализацию ПО УМОЛЧАНИЮ
 - 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
- Сравнить два решения (solution_1a и solution_2a) и сделать выводы
- Solution_3a – для функции apint_arith
 - Осуществить моделирование (с выводом результатов в консоль)
 - задать: clock period 10; clock_uncertainty 0.1
 - установить реализацию ПО УМОЛЧАНИЮ
 - 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
 - Сравнить два решения (solution_1a и solution_3a) и сделать выводы
 - Сравнить два решения (solution_2a и solution_3a) и сделать выводы

2. Первое решение

2.1. Исходный код

```
1  #include "source.h"
2
3  void types_standard(
4      din_A inA,
5      din_B inB,
6      din_C inC,
7      din_D inD,
8      dout_1 *out1,
9      dout_2 *out2,
10     dout_3 *out3,
11     dout_4 *out4) {
12
13     *out1 = inA * inB;
14     *out2 = inB + inA;
15     *out3 = inC / inA;
16     *out4 = inD % inA;
17 }
```

Рис. 2.1.1. Source code

```
1  #ifndef _TYPES_STANDARD_H_
2  #define _TYPES_STANDARD_H_
3
4  #include <stdio.h>
5  #include <stdint.h>
6
7  #define N 9
8
9  typedef char din_A;
10 typedef short din_B;
11 typedef int din_C;
12 typedef long long din_D;
13
14 typedef int dout_1;
15 typedef unsigned char dout_2;
16 typedef int32_t dout_3;
17 typedef int64_t dout_4;
18
19 void type_standard(din_A inA, din_B inB, din_C inC, din_D inD,
20     dout_1 *out1, dout_2 *out2, dout_3 *out3, dout_4 *out4);
21
22 #endif
```

Рис. 2.1.2. Source code - header

```
1  #include "source.h"
2
3  int main() {
4      din_A inA;
5      din_B inB;
6      din_C inC;
7      din_D inD;
8      dout_1 out1;
9      dout_2 out2;
10     dout_3 out3;
11     dout_4 out4;
12
13     int i, retval = 0;
14     FILE *fp;
15
16     fp = fopen("result.dat", "w");
17
18     for (i = 0; i < N; i++) {
19         inA = i + 2;
20         inB = i + 23;
21         inC = i + 234;
22         inD = i + 2345;
23
24         types_standard(inA, inB, inC, inD, &out1, &out2, &out3, &out4);
25 #ifdef __MINGW32__
26         fprintf(fp, "%d*%d=%d; %d+%d=%d; %d/%d=%d; %lld mod %d=%lld \n",
27             inA, inB, out1, inB, inA, out2, inC, inA, out3, inD, inA, out4);
28 #else
29         fprintf(fp, "%d*%d=%d; %d+%d=%d; %d/%d=%d; %lld mod %d=%lld \n",
30             inA, inB, out1, inB, inA, out2, inC, inA, out3, inD, inA, out4);
31 #endif
32     }
33
34     fclose(fp);
35
36     retval = system("diff --brief -w result.dat result.golden.dat");
37     if (retval == 0) {
38         printf("____Pass!____ \n");
39     } else {
40         printf("____Fail!____ \n");
41         retval = 1;
42     }
43
44     return retval;
45 }
```

Рис. 2.1.3. Test code

2.2. Моделирование

По результатам моделирования видно, что устройство работает корректно.

```
INFO: [HLS 200-10] Setting target device to 'xa7a12t-csg325-1Q'
INFO: [SIM 211-2] ***** CSIM start *****
INFO: [SIM 211-4] CSIM will launch GCC as the compiler.
    Compiling(apcc) ../../../../test.c in debug mode
INFO: [HLS 200-10] Running 'C:/Xilinx/Vivado/2019.2/bin/unwrapped/win64.o/apcc.exe'
INFO: [HLS 200-10] For user 'Misha' on host 'mikhail' (Windows NT_amd64 version 6.2)
INFO: [HLS 200-10] In directory 'C:/Users/Misha/Desktop/university/___maga/3sem/anton'
INFO: [APCC 202-3] Tmp directory is apcc_db
INFO: [APCC 202-1] APCC is done.
    Generating csim.exe
    Pass!
INFO: [SIM 211-1] CSim done with 0 errors.
INFO: [SIM 211-3] ***** CSIM finish *****
Finished C simulation.
```

Рис. 2.2. Modeling result

2.3.Синтез

Performance Estimates

Timing

Summary

Clock	Target	Estimated	Uncertainty
ap_clk	10.00 ns	7.180 ns	0.10 ns

Latency

Summary

Latency (cycles)		Latency (absolute)		Interval (cycles)		
min	max	min	max	min	max	Type
67	67	0.670 us	0.670 us	67	67	none

Рис. 2.3. Performance estimates

Полученная величина задержки укладывается в заданное значение.

2.4. Использование ресурсов

Utilization Estimates					
Summary					
Name	BRAM_18K	DSP48E	FF	LUT	URAM
DSP	-	1	-	-	-
Expression	-	-	0	15	-
FIFO	-	-	-	-	-
Instance	-	-	1173	707	-
Memory	-	-	-	-	-
Multiplexer	-	-	-	309	-
Register	-	-	68	-	-
Total	0	1	1241	1031	0
Available	40	40	16000	8000	0
Utilization (%)	0	2	7	12	0

Рис. 2.4.1. Utilization estimates

	Pipelined	Latency	Iteration Latency	Initiation Interval	Trip count
types_standard	-	67	-	68	-

Рис. 2.4.2. Performance profile

	BRAM	DSP	FF	LUT	Bits P0	Bits P1	Bits P2	Banks/Dep
types_standard	0	1	1241	1031				
I/O Ports(8)					256			
Instances(2)	0	0	1173	707				
Memories(0)	0		0	0	0			0
Expressions(1)	0	0	0	15	8	8	0	
Registers(1)			68	68				
Channels(0)	0		0	0	0			0
Multiplexers(1)	0		0	309	1			0
DSP(3)		1						

Рис. 2.4.3 Resource profile

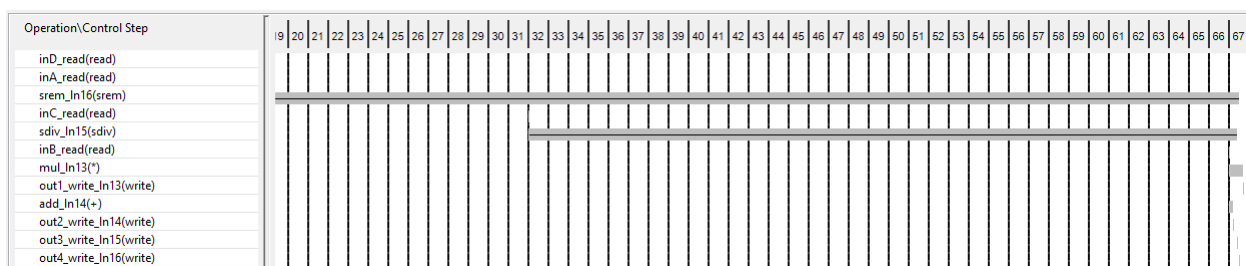


Рис. 2.4.4. Schedule viewer

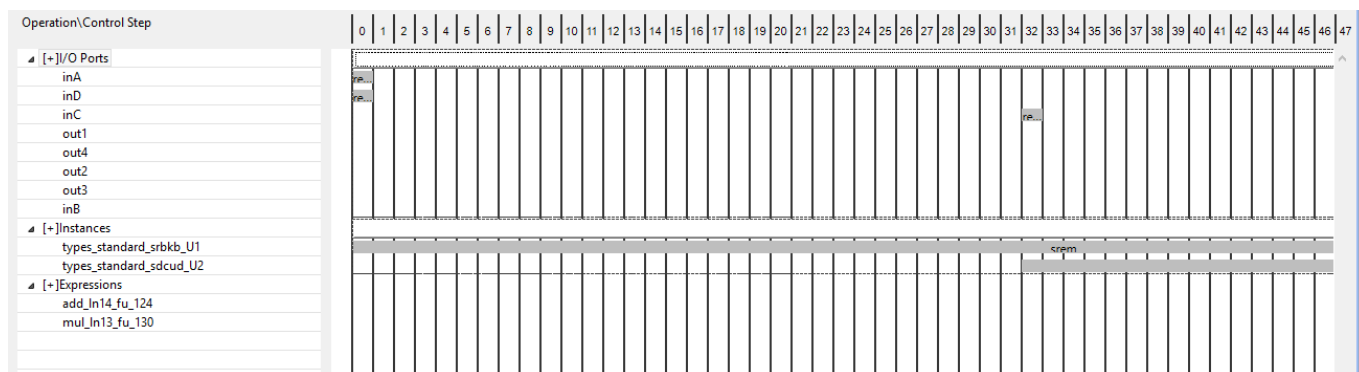


Рис. 2.4.5. Resource viewer

Исходя из диаграммы видно, что дольше всего выполняются операции деление нацело и деление с остатком – это связано с используемым типом данных.

3. Второе решение

3.1. Исходный код

```
1  #include "source_float.h"
2
3  void types_float_double(
4      din_A inA,
5      din_B inB,
6      din_C inC,
7      din_D inD,
8      dout_1 *out1,
9      dout_2 *out2,
10     dout_3 *out3,
11     dout_4 *out4) {
12
13     *out1 = inA * inB;
14     *out2 = inB + inA;
15     *out3 = inC / inA;
16     *out4 = sqrtf(inD);
17 }
```

Рис.3.1.1 Source code

```
1  #ifndef _SOURCE_FLOAT_H_
2  #define _SOURCE_FLOAT_H_
3
4  #include <math.h>
5  #include <stdio.h>
6  #include <stdint.h>
7
8  #define N 9
9
10 typedef double din_A;
11 typedef double din_B;
12 typedef double din_C;
13 typedef float din_D;
14
15 typedef double dout_1;
16 typedef double dout_2;
17 typedef double dout_3;
18 typedef float dout_4;
19
20 void types_float_double(din_A inA, din_B inB, din_C inC, din_D inD,
21     dout_1 *out1, dout_2 *out2, dout_3 *out3, dout_4 *out4);
22
23 #endif
```

Рис.3.1.2 Source code – header

```
1  #include "source_float.h"
2
3  int main() {
4      din_A inA;
5      din_B inB;
6      din_C inC;
7      din_D inD;
8      dout_1 out1;
9      dout_2 out2;
10     dout_3 out3;
11     dout_4 out4;
12
13     int i, retval = 0;
14     FILE *fp;
15
16     fp = fopen("result.dat", "w");
17
18     for (i = 0; i < N; i++) {
19         inA = i + 12.3;
20         inB = i + 12.34;
21         inC = i + 123.456;
22         inD = i + 1234.5678;
23
24         types_float_double(inA, inB, inC, inD, &out1, &out2, &out3, &out4);
25         fprintf(fp, "%f*%f=%f; %f+%f=%f; %f/%f=%f; %f sqrt =%.5f \n",
26             inA, inB, out1, inB, inA, out2, inC, inA, out3, inD, out4);
27     }
28     fclose(fp);
29
30     retval = system("diff --brief -w result.dat result.golden.dat");
31     if (retval == 0) {
32         printf("____Pass!____\n");
33     } else {
34         printf("____Fail!____\n");
35         retval = 1;
36     }
37
38     return retval;
39 }
```

Рис.3.1.3 Test code

3.2. Моделирование

По результатам моделирования видно, что устройство работает корректно.

```
Starting C simulation ...
C:/Xilinx/Vivado/2019.2/bin/vivado_hls.bat C:/Users/Misha/Desktop/university/___maga
INFO: [HLS 200-10] Running 'C:/Xilinx/Vivado/2019.2/bin/unwrapped/win64.o/vivado_hl
INFO: [HLS 200-10] For user 'Misha' on host 'mikhail' (Windows NT_amd64 version 6.2
INFO: [HLS 200-10] In directory 'C:/Users/Misha/Desktop/university/___maga/3sem/antc
Sourcing Tcl script 'C:/Users/Misha/Desktop/university/___maga/3sem/antonov/lab13/1s
INFO: [HLS 200-10] Opening project 'C:/Users/Misha/Desktop/university/___maga/3sem/s
INFO: [HLS 200-10] Opening solution 'C:/Users/Misha/Desktop/university/___maga/3sem/
INFO: [SYN 201-201] Setting up clock 'default' with a period of 10ns.
INFO: [SYN 201-201] Setting up clock 'default' with an uncertainty of 0.1ns.
WARNING: [HLS 200-40] Cannot find library 'C:/Xilinx/Vivado/2019.2/common/technolo
WARNING: [HLS 200-40] Cannot find library 'xilinx/aartix7/aartix7'.
INFO: [HLS 200-10] Setting target device to 'xa7a12t-csg325-1Q'
INFO: [SIM 211-2] ***** CSIM start *****
INFO: [SIM 211-4] CSIM will launch GCC as the compiler.
make: 'csim.exe' is up to date.
Pass!
INFO: [SIM 211-1] CSim done with 0 errors.
INFO: [SIM 211-3] ***** CSIM finish *****
Finished C simulation.
```

Рис.3.2 Modeling result

3.3. Синтез

Полученная величина задержки укладывается в заданное значение.

Performance Estimates

Timing

Summary

Clock	Target	Estimated	Uncertainty
ap_clk	10.00 ns	8.997 ns	0.10 ns

Latency

Summary

Latency (cycles)		Latency (absolute)		Interval (cycles)		
min	max	min	max	min	max	Type
30	30	0.300 us	0.300 us	30	30	none

Рис. 3.3. Performance estimates

3.4. Использование ресурсов

Utilization Estimates					
Summary					
Name	BRAM_18K	DSP48E	FF	LUT	URAM
DSP	-	-	-	-	-
Expression	-	-	-	-	-
FIFO	-	-	-	-	-
Instance	-	14	4379	4779	-
Memory	-	-	-	-	-
Multiplexer	-	-	-	145	-
Register	-	-	31	-	-
Total	0	14	4410	4924	0
Available	40	40	16000	8000	0
Utilization (%)	0	35	27	61	0

Рис. 3.4.1. Utilization estimates

	Pipelined	Latency	Iteration Latency	Initiation Interval	Trip count
types_float_double	-	30	-	31	-

Рис. 3.4.2. Performance profile

	BRAM	DSP	FF	LUT	Bits P0	Bits P1	Bits P2	Banks/Depth	Wor
types_float_double	0	14	4410	4924					
I/O Ports(8)				448					
Instances(4)	0	14	4379	4779					
Memories(0)	0		0	0	0		0		0
Expressions(0)	0	0	0	0	0	0	0		
Registers(1)			31	31					
Channels(0)	0		0	0	0		0		0
Multiplexers(1)	0		0	145	1		0		
DSP(4)		0							

Рис. 3.4.3 Resource profile

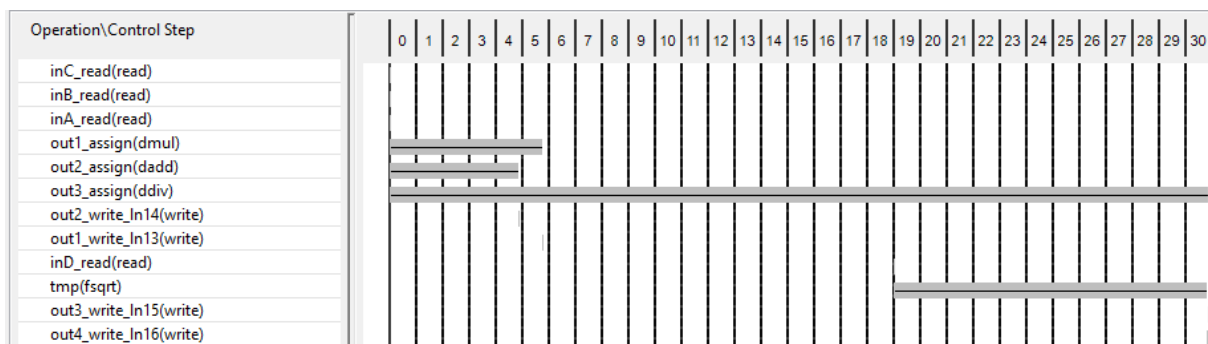


Рис. 3.4.4. Schedule viewer

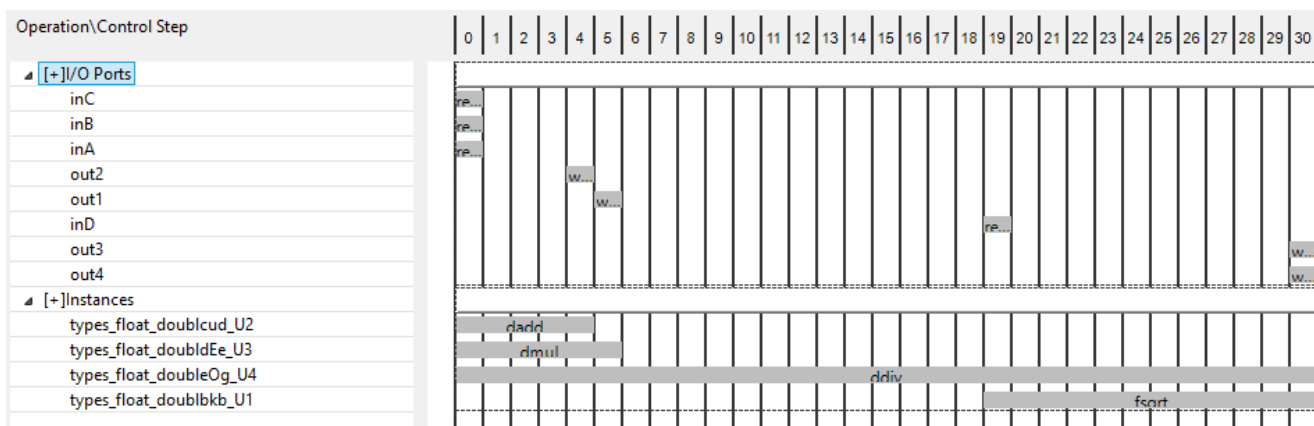


Рис. 3.4.5. Resource viewer

В данном проекте используется наибольшее количество ресурсов в связи с тем что используются 64 и 32-битные типы данных для синтеза которых требуется больше триггером и LUT, чем для типов данных char и short.

4. Третье решение

4.1. Исходный код

```
1  #include "apint_arith.h"
2
3  void apint_arith(
4      dinA_t inA,
5      dinB_t inB,
6      dinC_t inC,
7      dinD_t inD,
8      dout1_t *out1,
9      dout2_t *out2,
10     dout3_t *out3,
11     dout4_t *out4) {
12
13     *out1 = inA * inB;
14     *out2 = inB + inA;
15     *out3 = inC / inA;
16     *out4 = inD % inA;
17 }
```

Рис.4.1.1 Source code

```
1  #ifndef _APINT_ARITH_H_
2  #define _APINT_ARITH_H_
3
4  #include <stdio.h>
5  #include <ap_cint.h>
6
7  #define N 9
8
9  typedef int6 dinA_t;
10 typedef int12 dinB_t;
11 typedef int22 dinC_t;
12 typedef int33 dinD_t;
13
14 typedef int18 dout1_t;
15 typedef uint13 dout2_t;
16 typedef int22 dout3_t;
17 typedef int6 dout4_t;
18
19 void apint_arith(dinA_t inA, dinB_t inB, dinC_t inC, dinD_t inD,
20     dout1_t *out1, dout2_t *out2, dout3_t *out3, dout4_t *out4);
21
22 #endif
```

Рис.4.1.2 Source code – header

```
1  #include "apint_arith.h"
2
3  int main() {
4      dinA_t inA;
5      dinB_t inB;
6      dinC_t inC;
7      dinD_t inD;
8      dout1_t out1;
9      dout2_t out2;
10     dout3_t out3;
11     dout4_t out4;
12
13     int i, retval = 0;
14     FILE *fp;
15
16     fp = fopen("result.dat", "w");
17
18     for (i = 0; i < N; i++) {
19         inA = i + 2;
20         inB = i + 23;
21         inC = i + 234;
22         inD = i + 2345;
23
24         apint_arith(inA, inB, inC, inD, &out1, &out2, &out3, &out4);
25     #ifndef __MINGW32__
26         fprintf(fp, "%d*%d=%d; %d*%d=%d; %d/%d=%d; %lld mod %d =%d \n",
27             inA, inB, out1, inB, inA, out2, inC, inA, out3, inD, inA, out4);
28     #else
29         fprintf(fp, "%d*%d=%d; %d*%d=%d; %d/%d=%d; %d mod %d =%d \n",
30             inA, inB, out1, inB, inA, out2, inC, inA, out3, inD, inA, out4);
31     #endif
32     }
33     fclose(fp);
34
35     retval = system("diff --brief -w result.dat result.golden.dat");
36     if (retval == 0) {
37         printf("____Pass!____\n");
38     } else {
39         printf("____Fail!____\n");
40         retval = 1;
41     }
42
43     return retval;
44 }
```

Рис.4.1.3 Test code

4.2. Моделирование

По результатам моделирования видно, что устройство работает корректно.

```
Vivado HLS Console
INFO: [HLS 200-10] In directory 'C:/Users/Misha/Desktop/university/___maga/3sem/antonov/lab13/lab13'
Sourcing Tcl script 'C:/Users/Misha/Desktop/university/___maga/3sem/antonov/lab13/lab13'
INFO: [HLS 200-10] Opening project 'C:/Users/Misha/Desktop/university/___maga/3sem/antonov/lab13/lab13'
INFO: [HLS 200-10] Opening solution 'C:/Users/Misha/Desktop/university/___maga/3sem/antonov/lab13/lab13'
INFO: [SYN 201-201] Setting up clock 'default' with a period of 10ns.
INFO: [SYN 201-201] Setting up clock 'default' with an uncertainty of 0.1ns.
WARNING: [HLS 200-40] Cannot find library 'C:/Xilinx/Vivado/2019.2/common/technology/xilinx/aartix7/aartix7'.
WARNING: [HLS 200-40] Cannot find library 'xilinx/aartix7/aartix7'.
INFO: [HLS 200-10] Setting target device to 'xa7a12t-csg325-1Q'
INFO: [SIM 211-2] ***** CSIM start *****
INFO: [SIM 211-4] CSIM will launch GCC as the compiler.
make: 'csim.exe' is up to date.
Pass!
INFO: [SIM 211-1] CSim done with 0 errors.
INFO: [SIM 211-3] ***** CSIM finish *****
Finished C simulation.
```

Рис.4.2 Modeling result

4.3. Синтез

Полученная величина задержки укладывается в заданное значение.

Performance Estimates

Timing

Summary

Clock	Target	Estimated	Uncertainty
ap_clk	10.00 ns	7.180 ns	0.10 ns

Latency

Summary

Latency (cycles)		Latency (absolute)		Interval (cycles)		
min	max	min	max	min	max	Type
36	36	0.360 us	0.360 us	36	36	none

Рис.4.3 Performance estimates

4.4. Использование ресурсов

Utilization Estimates					
Summary					
Name	BRAM_18K	DSP48E	FF	LUT	URAM
DSP	-	1	-	-	-
Expression	-	-	0	20	-
FIFO	-	-	-	-	-
Instance	-	-	681	415	-
Memory	-	-	-	-	-
Multiplexer	-	-	-	169	-
Register	-	-	37	-	-
Total	0	1	718	604	0
Available	40	40	16000	8000	0
Utilization (%)	0	2	4	7	0

Рис.4.4.1 Utilization estimates

	Pipelined	Latency	Iteration Latency	Initiation Interval	Trip count
apint_arith	-	36	-	37	-

Рис.4.4.2 Performance profile

	BRAM	DSP	FF	LUT	Bits P0	Bits P1	Bits P2	Banks/Depth
apint_arith	0	1	718	604				
I/O Ports(8)					132			
Instances(2)	0	0	681	415				
Memories(0)	0		0	0	0			0
Expressions(1)	0	0	0	20	13	13	0	
Registers(1)			37		37			
Channels(0)	0		0	0	0			0
Multiplexers(1)	0		0	169	1			0
DSP(3)		1						

Рис.4.4.3 Resource profile

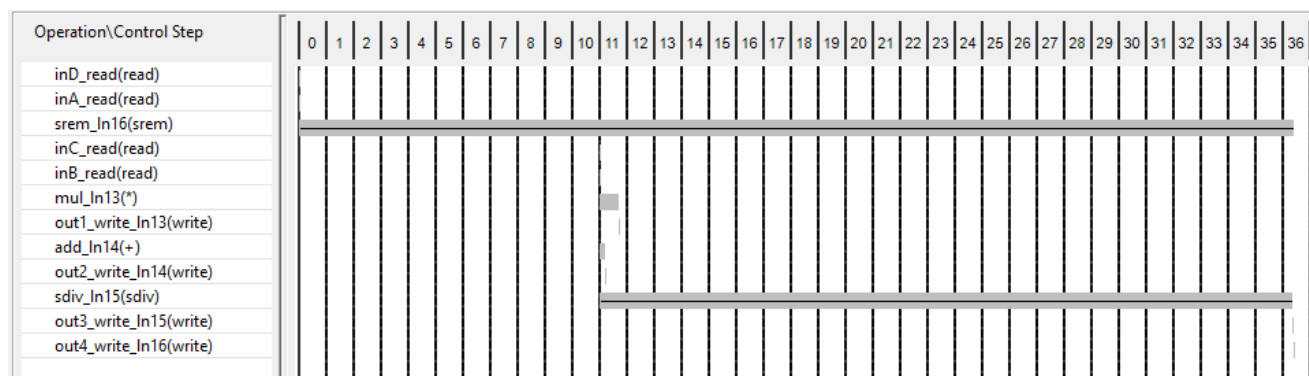


Рис.4.4.4 Schedule viewer

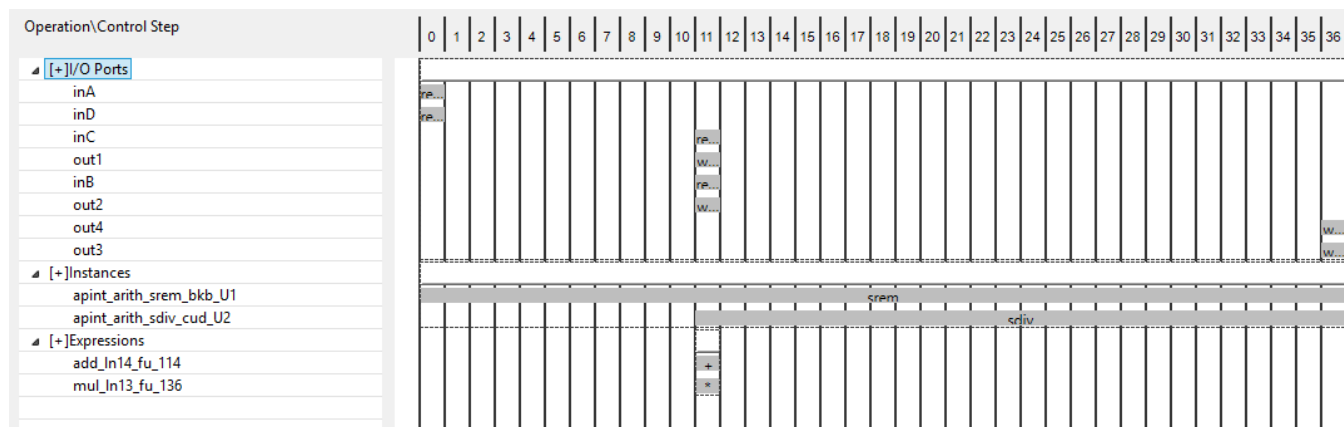


Рис.4.4.5 Resource viewer

В данном решении значение latency чуть больше, чем в лучшем решении и количество требуемых ресурсов чуть больше, чем в лучшем решении. Это связано с тем, что используются оптимальные типы данных для задачи. В связи с этим требуется оптимальное количество ресурсов.

5. Выводы

В ходе выполнения работы была исследована одна функция с разными типами данных. Было установлено, что чем лучше подобраны типы данных, на основе анализа возможных значений, тем оптимальнее синтезированная схема.