

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

Лабораторная работа 6

Предмет: Проектирование реконфигурируемых гибридных вычислительных систем

Тема: Port-Level IO Protocols

Задание 2

Студент: Ернязов Т.Е.

Гр. 3540901/81501

Преподаватель: Антонов А.П.

Санкт-Петербург  
2019

## Оглавление

Решение 1 .....	3
Решение 2 .....	7
Решение 3 .....	11
Решение 4 .....	15
Выводы .....	21

## Решение 1

### Создание файла с исходным кодом

```
void foo (int *d){
    static int acc = 0;
    int i = 0;
    acc += d[i];
    d[i] = acc;
}
```

### Создание тестового файла

```
#include <stdio.h>

int main()
{
    int d[1];
    d[0] = 0;

    foo(d);

    int res = 0;

    int pass;

    fprintf(stdout, "Expected    Actual\n");

        if (res == d[0]) {
            pass = 1;
        } else {
            pass = 0;
        }

    if (pass) {
        fprintf(stdout, "-----Pass!-----\n");
        return 0;
    } else {
        fprintf(stderr, "-----Fail!-----\n");
        return 1;
    }
}
```

### Создание решения с заданными параметрами

## Solution Configuration

Create Vivado HLS solution for selected technology

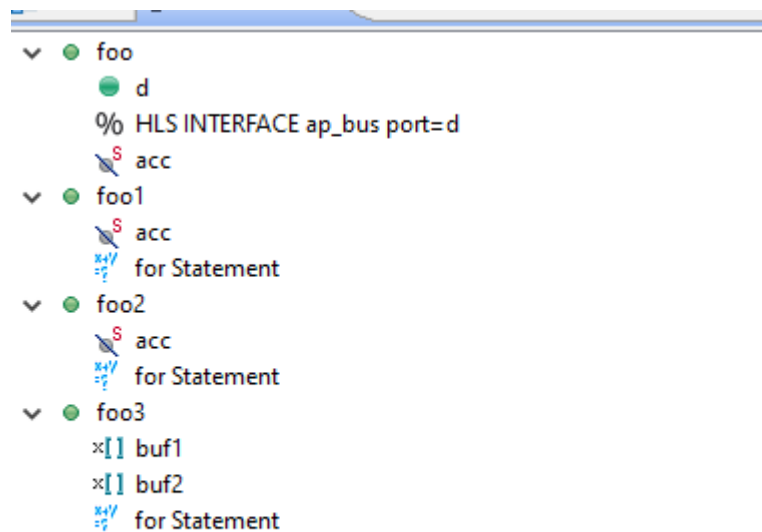


Solution Name:

Clock  
Period:  Uncertainty:

Part Selection  
Part: **xa7a12tcsg325-1q**

Задание интерфейса ap\_bus



## Моделирование

```
INFO: [APCC 202-3] Tmp directory is apcc_db
INFO: [APCC 202-1] APCC is done.
Generating csim.exe
Expected Actual
res: 0 == d[0]: 0
-----Pass!-----
INFO: [SIM 211-1] CSim done with 0 errors.
INFO: [SIM 211-3] ***** CSIM finish *****
Finished C simulation.
```

Результат моделирования успешный, заданное и полученное значения совпадают.

## Синтез

## Производительность

### Performance Estimates

#### Timing

##### Summary

Clock	Target	Estimated	Uncertainty
ap_clk	10.00 ns	9.900 ns	0.10 ns

#### Latency

##### Summary

Latency (cycles)		Latency (absolute)		Interval (cycles)		
min	max	min	max	min	max	Type
4	4	40.000 ns	40.000 ns	4	4	none

##### Detail

###### + Instance

###### + Loop

Полученная величина задержки укладывается в целевое значение.

Использование ресурсов

### Utilization Estimates

#### Summary

Name	BRAM_18K	DSP48E	FF	LUT	URAM
DSP	-	-	-	-	-
Expression	-	-	0	39	-
FIFO	-	-	-	-	-
Instance	-	-	-	-	-
Memory	-	-	-	-	-
Multiplexer	-	-	-	33	-
Register	-	-	101	-	-
Total	0	0	101	72	0
Available	40	40	16000	8000	0
Utilization (%)	0	0	~0	~0	0

Данное решение потребует на микросхеме 101 регистр и 72 LUT.

Интерфейсы соответствуют заданному ap\_bus при создании решения:

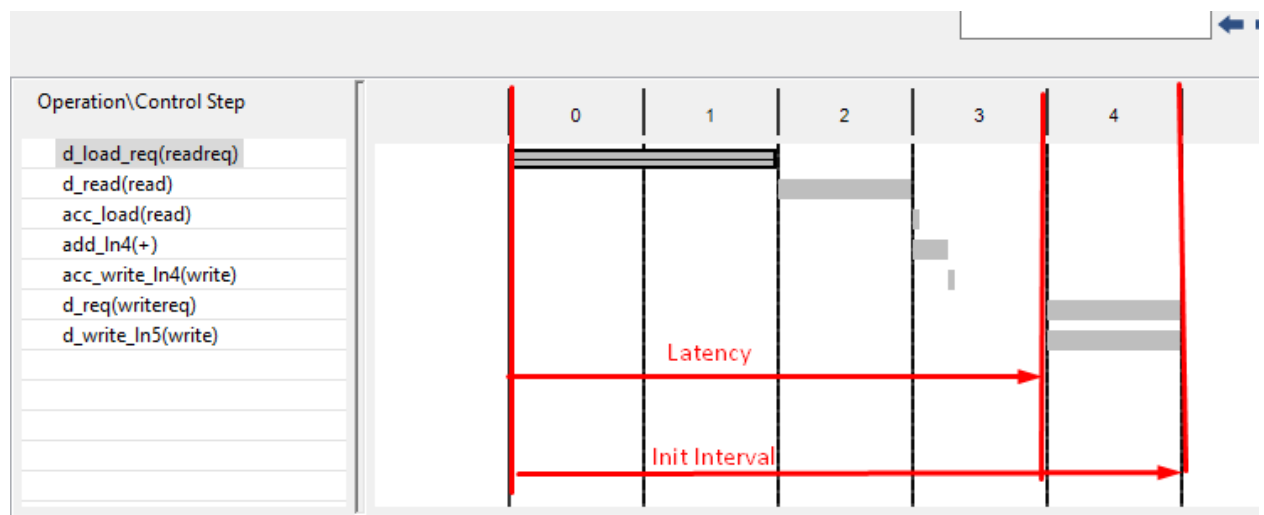
## Interface

### Summary

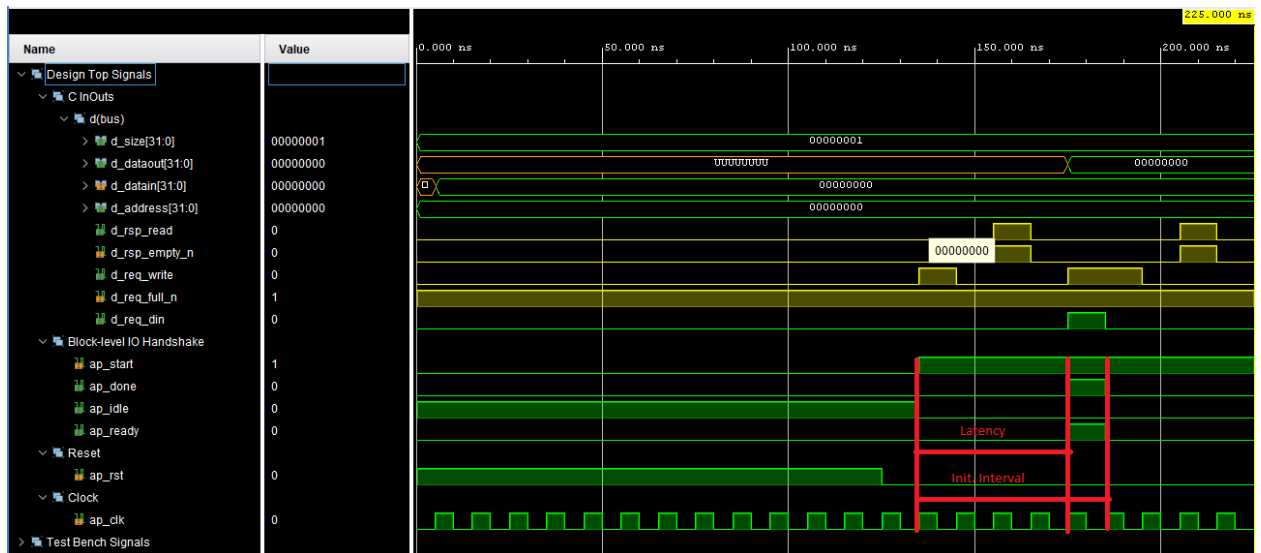
RTL Ports	Dir	Bits	Protocol	Source Object	C Type
ap_clk	in	1	ap_ctrl_hs	foo	return value
ap_rst	in	1	ap_ctrl_hs	foo	return value
ap_start	in	1	ap_ctrl_hs	foo	return value
ap_done	out	1	ap_ctrl_hs	foo	return value
ap_idle	out	1	ap_ctrl_hs	foo	return value
ap_ready	out	1	ap_ctrl_hs	foo	return value
d_req_din	out	1	ap_bus	d	pointer
d_req_full_n	in	1	ap_bus	d	pointer
d_req_write	out	1	ap_bus	d	pointer
d_rsp_empty_n	in	1	ap_bus	d	pointer
d_rsp_read	out	1	ap_bus	d	pointer
d_address	out	32	ap_bus	d	pointer
d_datain	in	32	ap_bus	d	pointer
d_dataout	out	32	ap_bus	d	pointer
d_size	out	32	ap_bus	d	pointer

Performance Profile		Resource Profile			
	Pipelined	Latency	Iteration Latency	Initiation Interval	Trip count
foo	-	4	-	5	-

Задержка составляет 4 такта, интервал инициализации составляет 5 тактов.



C/RTL моделирование



На временной диаграмме отображен интервал инициализации и задержка.

## Решение 2

### Создание исходного файла

```
void fool (int *d){
    static int acc = 0;
    for (int i = 0; i < 4; i++){
        acc += d[i];
        d[i] = acc;
    }
}
```

### Создание тестового файла

```
#include <stdio.h>

int main()
{
    int d[4] = {0, 1, 2, 3};

    fool(d);

    int res[4] = {0, 1, 3, 6};

    int pass;

    fprintf(stdout, "Expected    Actual\n");

    for (int i = 0; i < 4; i++)    {
        fprintf(stdout, "res[%d]: %d == d[%d]: %d\n", i, res[i], i,
d[i]);

        if (res[i] == d[i]) {
            pass = 1;
        } else {
            pass = 0;
            break;
        }
    }
}
```

```

if (pass) {
    fprintf(stdout, "-----Pass!-----\n");
    return 0;
} else {
    fprintf(stderr, "-----Fail!-----\n");
    return 1;
}
}

```

## Настройка решения

### Solution Configuration

Create Vivado HLS solution for selected technology



Solution Name:

Clock  
Period:  Uncertainty:

Part Selection  
Part: **xa7a12tcsq325-1q** ...

Options  
☐ Copy directives and constraints from solution:  ▼

## Установка интерфейса

```

▼ ● foo1
  ● d
  % HLS INTERFACE ap_bus depth=4 port=d
  S acc
  for Statement

```

## Моделирование

```

-----g-----
Expected  Actual
res[0]: 0 == d[0]: 0
res[1]: 1 == d[1]: 1
res[2]: 3 == d[2]: 3
res[3]: 6 == d[3]: 6
-----Pass!-----
INFO: [SIM 211-1] CSim done with 0 errors.
INFO: [SIM 211-3] ***** CSIM finish *****
Finished C simulation.

```

## Синтез

## Производительность



## Performance Estimates

### Timing

#### Summary

Clock	Target	Estimated	Uncertainty
ap_clk	10.00 ns	9.900 ns	0.10 ns

### Latency

#### Summary

Latency (cycles)		Latency (absolute)		Interval (cycles)		Type
min	max	min	max	min	max	
14	14	0.140 us	0.140 us	14	14	none

#### Detail

##### + Instance

##### + Loop

Полученное значение задержки укладывается в заданное и соответствует решению 1.

## Использование ресурсов

## Utilization Estimates

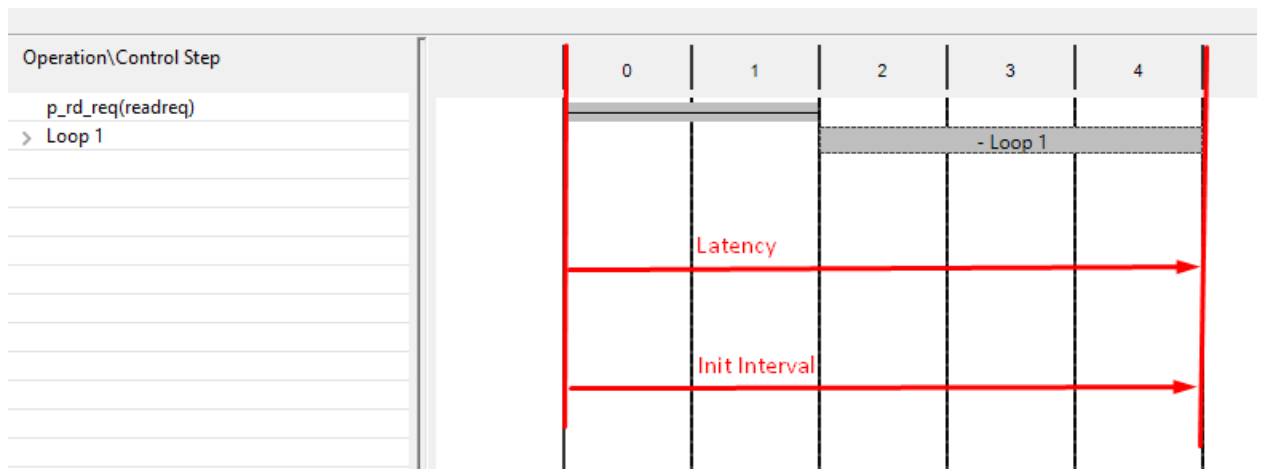
### Summary

Name	BRAM_18K	DSP48E	FF	LUT	URAM
DSP	-	-	-	-	-
Expression	-	-	0	71	-
FIFO	-	-	-	-	-
Instance	-	-	-	-	-
Memory	-	-	-	-	-
Multiplexer	-	-	-	42	-
Register	-	-	108	-	-
Total	0	0	108	113	0
Available	40	40	16000	8000	0
Utilization (%)	0	0	~0	1	0

Данное решение потребует на микросхеме 108 регистров и 113 элементов LUT. По сравнению с решением 1 значительно увеличилось число LUT.

Performance Profile		Resource Profile			
	Pipelined	Latency	Iteration Latency	Initiation Interval	Trip count
✓ ● foo1	-	14	-	15	-
● Loop 1	no	12	3	-	4

Задержка на одной итерации занимает 3 такта, для всего цикла – 12 тактов (так как производится 4 итерации). Интервал инициализации занимает 15 тактов.

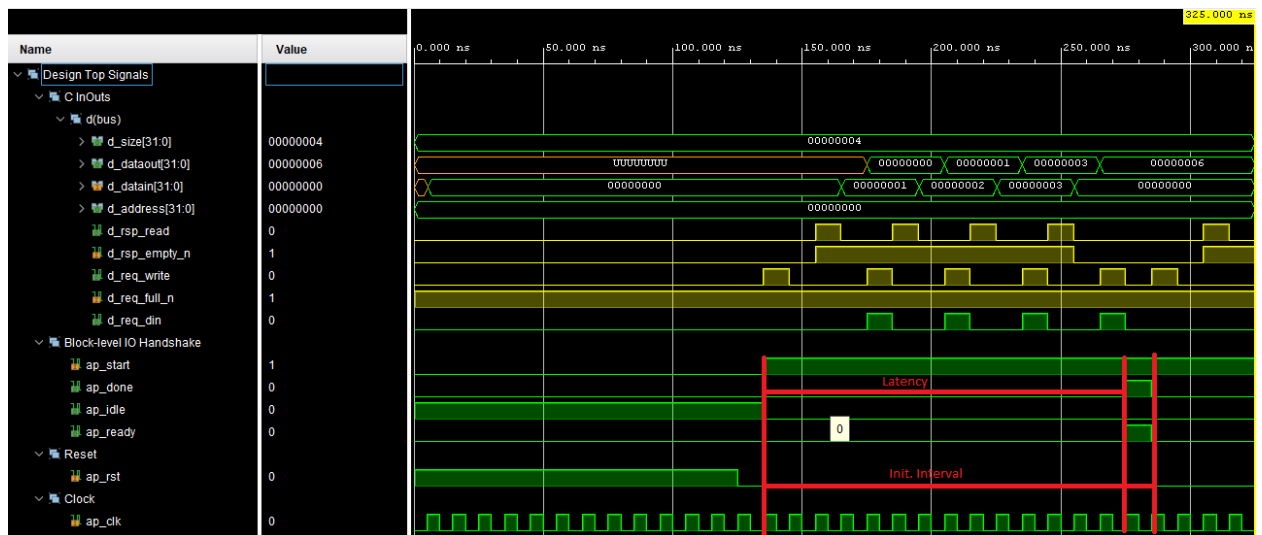


## Интерфейс

Interface						
Summary						
RTL Ports	Dir	Bits	Protocol	Source Object	C Type	
ap_clk	in	1	ap_ctrl_hs	foo1	return value	
ap_rst	in	1	ap_ctrl_hs	foo1	return value	
ap_start	in	1	ap_ctrl_hs	foo1	return value	
ap_done	out	1	ap_ctrl_hs	foo1	return value	
ap_idle	out	1	ap_ctrl_hs	foo1	return value	
ap_ready	out	1	ap_ctrl_hs	foo1	return value	
d_req_din	out	1	ap_bus	d	pointer	
d_req_full_n	in	1	ap_bus	d	pointer	
d_req_write	out	1	ap_bus	d	pointer	
d_rsp_empty_n	in	1	ap_bus	d	pointer	
d_rsp_read	out	1	ap_bus	d	pointer	
d_address	out	32	ap_bus	d	pointer	
d_datain	in	32	ap_bus	d	pointer	
d_dataout	out	32	ap_bus	d	pointer	
d_size	out	32	ap_bus	d	pointer	

Интерфейс соответствуют заданному при настройке проекта ap\_bus.

C/RTL моделирование



На временной диаграмме отображена задержка и интервал инициализации.

### Решение 3

#### Создание исходного файла

```
void foo2 (int *d){
    static int acc = 0;
    for (int i = 0; i < 4; i++){
        acc += *(d+i);
        *(d+i) = acc;
    }
}
```

#### Создание тестового файла

```
#include <stdio.h>

int main()
{
    int d[4] = {0, 1, 2, 3};

    foo2(d);

    int res[4] = {0, 1, 3, 6};

    int pass;

    fprintf(stdout, "Expected    Actual\n");

    for (int i = 0; i < 4; i++) {
        fprintf(stdout, "res[%d]: %d == d[%d]: %d\n", i, res[i], i,
d[i]);

        if (res[i] == d[i]) {
            pass = 1;
        } else {
            pass = 0;
            break;
        }
    }
}
```

```

    }

    if (pass) {
        fprintf(stdout, "-----Pass!-----\n");
        return 0;
    } else {
        fprintf(stderr, "-----Fail!-----\n");
        return 1;
    }
}

```

## Настройка решения

**Solution Configuration**

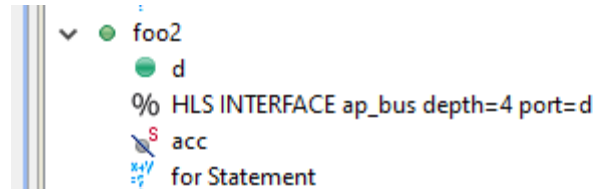
Create Vivado HLS solution for selected technology

Solution Name:

Clock  
 Period:  Uncertainty:

Part Selection  
 Part: **xa7a12tcsg325-1q** ...

## Настройка интерфейса



## Моделирование

```

Vivado HLS Console
INFO: [APCC 202-3] Tmp directory is apcc_db
INFO: [APCC 202-1] APCC is done.
Generating csim.exe
Expected Actual
res[0]: 0 == d[0]: 0
res[1]: 1 == d[1]: 1
res[2]: 3 == d[2]: 3
res[3]: 6 == d[3]: 6
-----Pass!-----
INFO: [SIM 211-1] CSim done with 0 errors.
INFO: [SIM 211-3] ***** CSIM finish *****
Finished C simulation.

```

Моделирование пройдено успешно.

## Синтез

## Производительность

## Performance Estimates

### Timing

#### Summary

Clock	Target	Estimated	Uncertainty
ap_clk	10.00 ns	9.900 ns	0.10 ns

### Latency

#### Summary

Latency (cycles)		Latency (absolute)		Interval (cycles)		
min	max	min	max	min	max	Type
14	14	0.140 us	0.140 us	14	14	none

#### Detail

##### + Instance

##### + Loop

Полученная величина задержки укладывается в заданное значение и соответствует предыдущим решениям.

## Использование ресурсов

## Utilization Estimates

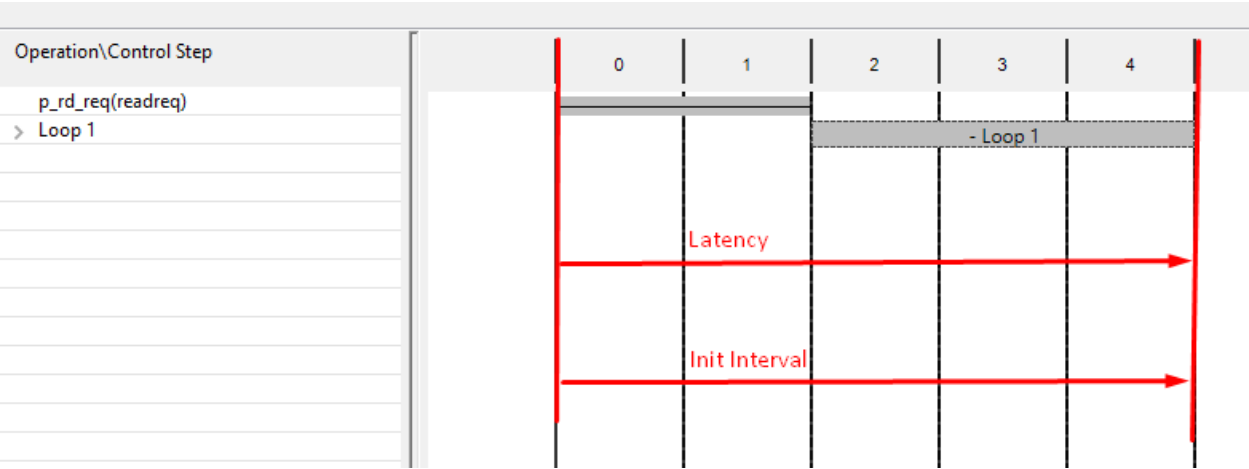
### Summary

Name	BRAM_18K	DSP48E	FF	LUT	URAM
DSP	-	-	-	-	-
Expression	-	-	0	71	-
FIFO	-	-	-	-	-
Instance	-	-	-	-	-
Memory	-	-	-	-	-
Multiplexer	-	-	-	42	-
Register	-	-	108	-	-
Total	0	0	108	113	0
Available	40	40	16000	8000	0
Utilization (%)	0	0	~0	1	0

Данное решение потребует на микросхеме 108 регистров и 113 элементов LUT. Полученные значения соответствуют значениям в решении 2 так как оба решения определяют множественную запись и чтение.

	Pipelined	Latency	Iteration Latency	Initiation Interval	Trip count
▼ ● foo2	-	14	-	15	-
● Loop 1	no	12	3	-	4

Величина задержки для одной итерации составляет 3 такта, количество итераций – 4, полная величина задержки равна 14 тактам, интервал инициализации составляет 15 тактов.

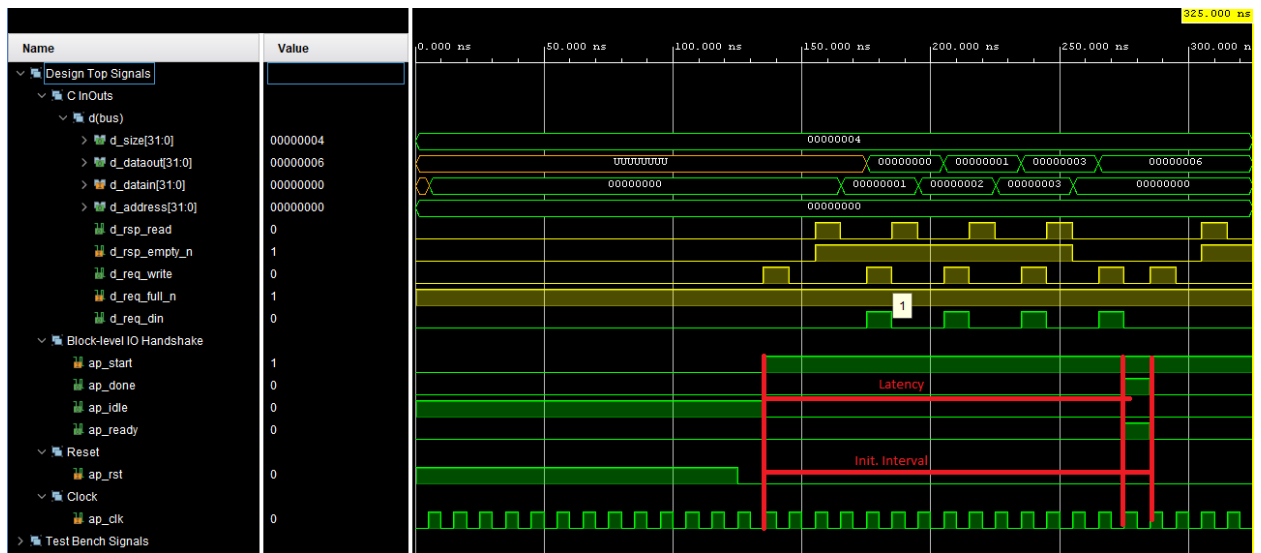


### Интерфейс

Interface					
Summary					
RTL Ports	Dir	Bits	Protocol	Source Object	C Type
ap_clk	in	1	ap_ctrl_hs	foo2	return value
ap_rst	in	1	ap_ctrl_hs	foo2	return value
ap_start	in	1	ap_ctrl_hs	foo2	return value
ap_done	out	1	ap_ctrl_hs	foo2	return value
ap_idle	out	1	ap_ctrl_hs	foo2	return value
ap_ready	out	1	ap_ctrl_hs	foo2	return value
d_req_din	out	1	ap_bus	d	pointer
d_req_full_n	in	1	ap_bus	d	pointer
d_req_write	out	1	ap_bus	d	pointer
d_rsp_empty_n	in	1	ap_bus	d	pointer
d_rsp_read	out	1	ap_bus	d	pointer
d_address	out	32	ap_bus	d	pointer
d_datain	in	32	ap_bus	d	pointer
d_dataout	out	32	ap_bus	d	pointer
d_size	out	32	ap_bus	d	pointer

Интерфейс соответствует заданному ap\_bus при конфигурировании решения.

C/RTL моделирование



На временной диаграмме отображены задержка и интервал инициализации.

## Решение 4

### Создание исходного файла

```
void foo3 (int *d){
    int buf1[4], buf2[4];
    int i;

    memcpy(buf1, d, 4*sizeof(int));
    for (i=0; i < 4; i++){
        buf2[i] = buf1[3-i];
    }
    memcpy(d, buf2, 4*sizeof(int));
}
```

```
void * memcpy( void * destptr, const void * srcptr, size_t num );
```

Функция `memcpy` копирует `num` байтов первого блока памяти, на который ссылается указатель `srcptr`, во второй блок памяти, на который ссылается указатель `destptr`.

Тип данных объектов, на которые указывают как `srcptr` так и `destptr` не имеют никакого значения. Так как эта функция работает с бинарными данными.

Функция не проверяет, есть ли символ завершения в `srcptr`, она всегда копирует количество байтов, указанное в `num`.

Чтобы избежать переполнения блока памяти `destptr`, размер `destptr` должен быть не менее `num` байтов. Однако, может возникнуть ситуация, когда `destptr` и `srcptr` пересекутся. Поэтому, для перекрытия блоков памяти, функция `memmove` является более безопасным подходом.

Параметры:

Destptr - Указатель на блок памяти назначения (куда будут копироваться байты данных), имеет тип данных void.

Srcptr - Указатель на блок памяти источник (т. е., откуда будут копироваться байты данных), имеет тип данных void.

Num - Количество копируемых байтов.

Возвращаемое значение - Указатель на блок памяти назначения.

Создание тестового файла

```
#include <stdio.h>

int main()
{
    int d[4] = {0, 1, 2, 3};

    foo3(d);

    int res[4] = {3, 2, 1, 0};

    int pass;

    fprintf(stdout, "Expected   Actual\n");

    for (int i = 0; i < 4; i++)    {
        fprintf(stdout, "res[%d]: %d == d[%d]: %d\n", i, res[i], i,
d[i]);

        if (res[i] == d[i]) {
            pass = 1;
        } else {
            pass = 0;
            break;
        }
    }


    if (pass) {
        fprintf(stdout, "-----Pass!-----\n");
        return 0;
    } else {
        fprintf(stderr, "-----Fail!-----\n");
        return 1;
    }
}
```

Настройка решения



## Solution Configuration

Create Vivado HLS solution for selected technology



Solution Name:

Clock

Period: 
Uncertainty:

Part Selection

Part: **xa7a12tcsg325-1q**

...

```

1
2
3 void foo3 (int *d){
4     #pragma HLS INTERFACE ap_bus depth=4 port=d
5     int buf1[4], buf2[4];
6     int i;
7
8     memcpy(buf1, d, 4*sizeof(int));
9     for (i=0; i < 4; i++){
10         buf2[i] = buf1[3-i];
11     }
12     memcpy(d, buf2, 4*sizeof(int));
13 }
14
15

```

В этот раз директива интерфейса вставлена прямо в исходный код, иначе синтез не будет проходить из за использования memcpy без инициализированного интерфейса шины.

## Моделирование

```

INFO: [APCC 202-3] Imp directory is apcc_db
INFO: [APCC 202-1] APCC is done.
    Generating csim.exe
Expected   Actual
res[0]: 3 == d[0]: 3
res[1]: 2 == d[1]: 2
res[2]: 1 == d[2]: 1
res[3]: 0 == d[3]: 0
-----Pass!-----
INFO: [SIM 211-1] CSim done with 0 errors.
INFO: [SIM 211-3] ***** CSIM finish *****
Finished C simulation.

```

Моделирование выполнено успешно

## Синтез

## Производительность

## Performance Estimates

### Timing

#### Summary

Clock	Target	Estimated	Uncertainty
ap_clk	10.00 ns	9.900 ns	0.10 ns

### Latency

#### Summary

Latency (cycles)		Latency (absolute)		Interval (cycles)		Type
min	max	min	max	min	max	
18	18	0.180 us	0.180 us	18	18	none

#### Detail

##### + Instance

##### + Loop

Полученная величина задержки укладывается в заданное значение и соответствует предыдущим решениям.

## Использование ресурсов

## Utilization Estimates

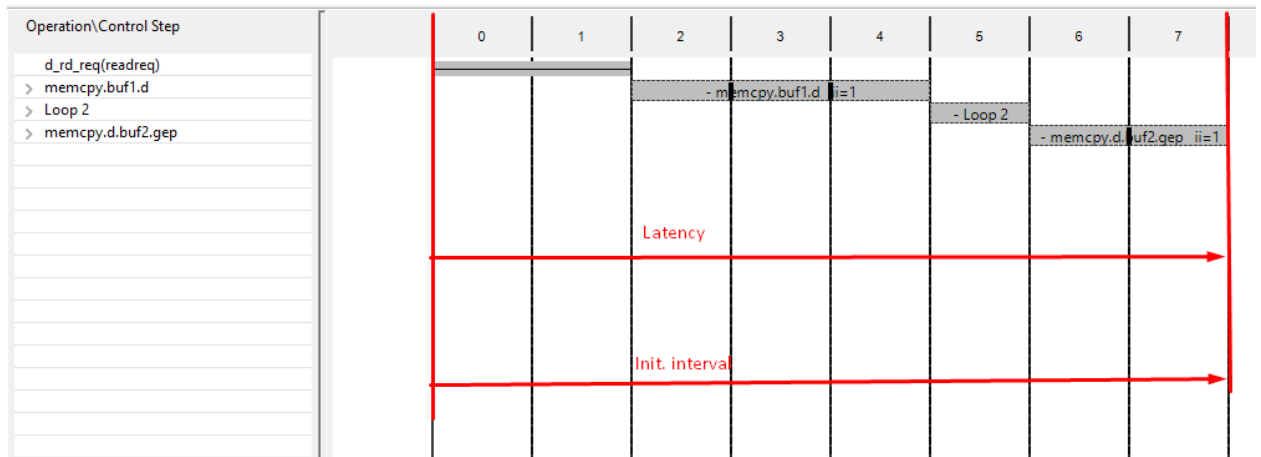
### Summary

Name	BRAM_18K	DSP48E	FF	LUT	URAM
DSP	-	-	-	-	-
Expression	-	-	0	88	-
FIFO	-	-	-	-	-
Instance	-	-	0	42	-
Memory	-	-	-	-	-
Multiplexer	-	-	-	98	-
Register	-	-	312	-	-
Total	0	0	312	228	0
Available	40	40	16000	8000	0
Utilization (%)	0	0	1	2	0

Данное решение требует 312 регистров и 228 элементов LUT, что значительно больше по сравнению с предыдущими решениями. Число требуемых регистров и элементов LUT возросло примерно в 3 раза по сравнению с предыдущими решениями.

Performance Profile		Resource Profile			
	Pipelined	Latency	Iteration Latency	Initiation Interval	Trip count
foo3	-	18	-	19	-
memcpy.buf1.d	yes	4	2	1	4
Loop 2	no	4	1	-	4
memcpy.d.buf2.gep	yes	4	2	1	4

Задержка одной итерации цикла составляет 1 такт, для всего цикла – 4 такта. Задержки для выполнения memcpy по 2 такта. Интервал инициализации составляет 19 тактов.

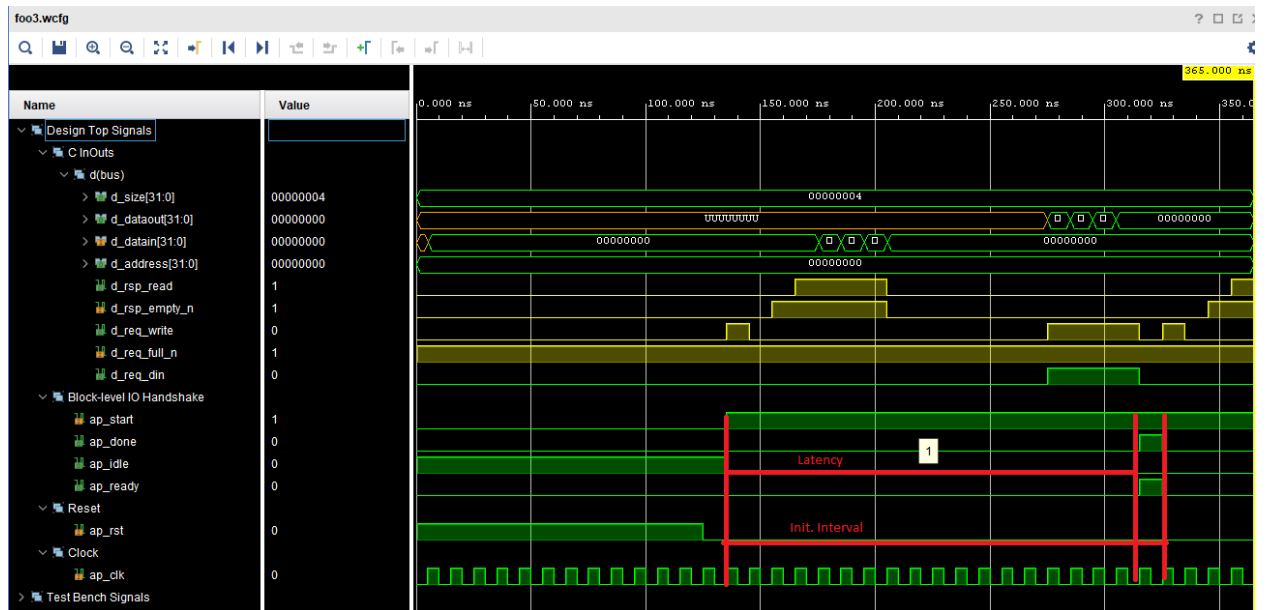


## Интерфейс

Interface					
Summary					
RTL Ports	Dir	Bits	Protocol	Source Object	C Type
ap_clk	in	1	ap_ctrl_hs	foo3	return value
ap_rst	in	1	ap_ctrl_hs	foo3	return value
ap_start	in	1	ap_ctrl_hs	foo3	return value
ap_done	out	1	ap_ctrl_hs	foo3	return value
ap_idle	out	1	ap_ctrl_hs	foo3	return value
ap_ready	out	1	ap_ctrl_hs	foo3	return value
d_req_din	out	1	ap_bus	d	pointer
d_req_full_n	in	1	ap_bus	d	pointer
d_req_write	out	1	ap_bus	d	pointer
d_rsp_empty_n	in	1	ap_bus	d	pointer
d_rsp_read	out	1	ap_bus	d	pointer
d_address	out	32	ap_bus	d	pointer
d_datain	in	32	ap_bus	d	pointer
d_dataout	out	32	ap_bus	d	pointer
d_size	out	32	ap_bus	d	pointer

Интерфейс соответствует заданному в начале решения ap\_bus.

## C/RTL моделирование



На временной диаграмме отображена задержка и интервал инициализации.

## **Выводы**

В ходе работы были построены четыре решения, одно с одиночным вариантом чтения и записи, два – с множественным и одно с режимом потокового обмена. Был использован интерфейс `ap_bus`. Применялись следующие порты: `D_datain` – входные данные, `D_req_full_n` - активный низкий уровень сигнала указывает что мост полный, `D-rsp_empty_n` – указывает на готовность принятия данных, `D_dataout` – выходные данные и другие, например, определяющие адрес и размер, готовность чтения и записи. Временные диаграммы и анализ решений приведены в отчете.