HLS Assignment 2

(KANEKAL KOUSAR[FWC2022063])

2.1) Use 32bit inputs and figure out the what the bitwidth should be for your design using your understanding of digital design and arithmetic operations

Header file

```
#ifndef MUL
#define MUL

#include <iostream>
using namespace std;

typedef long long out;

struct inputs{
	int A;
	int B;
};

#endif
```

C++ code

```
#include "mul.h"

void mul32_old(inputs din,out &dout){
	dout=(din.A)*(din.B);
}
```

Testbench

```
#include "mul.h"
void mul32_old(inputs din,out &dout);
int main(){
	inputs in;
	out out;
	int i;
	for (i=0;i<10;i++){
		in.A=i+2;
		in.B=i;
		mul32_old(in,out);
		cout<<in.A<<"X"<<in.B<<"="<< out <<endl;
	}
	return 0;
}
```

Simulation report:

```
INFO: [SIM 2] *************** CSIM start ***************
INFO: [SIM 4] CSIM will launch GCC as the compiler.
   Compiling ../../../mul_tb.cpp in debug mode
   Compiling ../../../mul.cpp in debug mode
   Generating csim.exe
```

```
2X0=0
3X1=3
4X2=8
5X3=15
6X4=24
7X5=35
8X6=48
9X7=63
10X8=80
11X9=99
INFO: [SIM 1] CSim done with 0 errors.
INFO: [SIM 3] *************** CSIM finish ***************
```

Synthesis report:

Synthesis Report for 'mul32_old'

General Information

Date:	Sat Mar 18 22:09:41 2023
Version:	2017.4 (Build 2086221 on Fri Dec 15 21:13:33 MST 2017)
Project:	assignment2.1_new
Solution:	solution1
Product family:	zynq
Target device:	xc7z020clg484-1

Performance Estimates

Timing (ns)

Summary

Clock	Target	Estimated	Uncertainty
ap_clk	10.00	8.51	1.25

Latency (clock cycles)

Summary

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

Detail

Instance

Loop

Utilization Estimates

Summary

Name	BRAM_18K	DSP48E	FF	LUT
DSP	-	-	-	-
Expression	-	3	0	20
FIFO	-	-	-	-
Instance	-	-	-	-
Memory	-	-	-	-
Multiplexer	-	-	-	-
Register	-	-	-	-
Total	0	3	0	20
Available	280	220	106400	53200
Utilization (%)	0	1	0	~0

Detail

Instance

DSP48

Memory

FIFO

Expression

Variable Name	Operation	DSP48E	FF	LUT	Bitwidth P0	Bitwidth P1
tmp_fu_35_p2	*	3	0	20	32	32
Total	1	3	0	20	32	32

Multiplexer

Register

Interface

Summary

RTL Ports	Dir	Bits	Protocol	Source Object	C Type
ap_start	in	1	ap_ctrl_hs	mul32_old	return value
ap_done	out	1	ap_ctrl_hs	mul32_old	return value
ap_idle	out	1	ap_ctrl_hs	mul32_old	return value
ap_ready	out	1	ap_ctrl_hs	mul32_old	return value
din_A	in	32	ap_none	din_A	scalar
din_B	in	32	ap_none	din_B	scalar
dout	out	64	ap_vld	dout	pointer
dout_ap_vld	out	1	ap_vld	dout	pointer

Cosimulation report

```
INFO: [Common 17-206] Exiting xsim at Sat Mar 18 22:17:02 2023...
INFO: [COSIM 212-316] Starting C post checking ...
2X0=0
3X1=3
4X2=8
5X3=15
6X4=24
7X5=35
8X6=48
9X7=63
10X8=80
11X9=99
INFO: [COSIM 212-1000] *** C/RTL co-simulation finished: PASS ***
INFO: [COSIM 212-210] Design is translated to an combinational logic. II and
Latency will be marked as all 0.
Finished C/RTL cosimulation.
```

Cosimulation Report for 'mul32_old'

Result

		Latency			Interval		
RTL	Status	min	avg	max	min	avg	max
VHDL	NA	NA	NA	NA	NA	NA	NA
Verilog	Pass	0	0	0	0	0	0

Export the report(.html) using the Export Wizard

2.2) Use arbitrary precision data type with bitwidth of 4 for integer and 24 for fractional part of the inputs and figure out what the bitwidth of the output should be for your design.

Header file

```
#ifndef MUL
#define MUL

#include <iostream>
#include "ap_fixed.h"
using namespace std;

typedef ap_ufixed<28,4> fix28_4;
typedef ap_ufixed<56,8> fix56_8;

struct inputs{
	fix28_4 A;
	fix28_4 B;
};
#endif
```

C++ code

```
#include "mul.h"
void mulf(inputs din,fix56_8 &dout){
	dout=din.A*din.B;
}
```

Test bench

```
#include "mul.h"
void mulf(inputs din,fix56_8 &dout);
int main(){
	inputs in;
	fix56_8 out;
	int i;
	for (i=0;i<10;i++){
		in.A=i+0.64;
		in.B=i+0.52;
		mulf(in,out);
		cout <<in.A<<"X"<<in.B<<"="<<out<< endl;
	}
}
```

Simulation report

```
INFO: [SIM 2] *************** CSIM start ***************
INFO: [SIM 4] CSIM will launch GCC as the compiler.
   Compiling ../../../mul_tb.cpp in debug mode
   Compiling ../../../mul.cpp in debug mode
   Generating csim.exe
0.64X0.52=0.3328
1.64X1.52=2.4928
2.64X2.52=6.6528
3.64X3.52=12.8128
```

```
4.64X4.52=20.9728
5.64X5.52=31.1328
6.64X6.52=43.2928
7.64X7.52=57.4528
8.64X8.52=73.6128
9.64X9.52=91.7728
INFO: [SIM 1] CSim done with 0 errors.
INFO: [SIM 3] *************** CSIM finish ***************
```

Synthesis report

Synthesis Report for 'mulf'

General Information

Date:	Sat Mar 18 23:14:32 2023
Version:	2017.4 (Build 2086221 on Fri Dec 15 21:13:33 MST 2017)
Project:	assignment2.2_old
Solution:	solution1
Product family:	zynq
Target device:	xc7z020clg484-1

Performance Estimates

Timing (ns)

Summary

Clock	Target	Estimated	Uncertainty
ap_clk	10.00	7.45	1.25

Latency (clock cycles)

Summary

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

Detail

- Instance
- Loop

Utilization Estimates

Summary

Name	BRAM_18K	DSP48E	FF	LUT
DSP	-	-	-	-
Expression	-	3	0	36
FIFO	-	-	-	-
Instance	-	-	-	-
Memory	-	-	-	-
Multiplexer	-	-	-	-
Register	-	-	-	-
Total	0	3	0	36
Available	280	220	106400	53200
Utilization (%)	0	1	0	~0

Detail

- Instance
- DSP48
- Memory
- FIFO
- Expression
- Multiplexer
- Register

Interface

Summary

RTL Ports	Dir	Bits	Protocol	Source Object	C Type
ap_start	in	1	ap_ctrl_hs	mulf	return value
ap_done	out	1	ap_ctrl_hs	mulf	return value
ap_idle	out	1	ap_ctrl_hs	mulf	return value
ap_ready	out	1	ap_ctrl_hs	mulf	return value
din_A_V	in	28	ap_none	din_A_V	scalar
din_B_V	in	28	ap_none	din_B_V	scalar
dout_V	out	56	ap_vld	dout_V	pointer
dout_V_ap_vld	out	1	ap_vld	dout_V	pointer

Co-simulation report

```
INFO: [Common 17-206] Exiting xsim at Sat Mar 18 23:19:27 2023...
INFO: [COSIM 212-316] Starting C post checking ...
0.64X0.52=0.3328
1.64X1.52=2.4928
2.64X2.52=6.6528
3.64X3.52=12.8128
4.64X4.52=20.9728
5.64X5.52=31.1328
6.64X6.52=43.2928
7.64X7.52=57.4528
8.64X8.52=73.6128
9.64X9.52=91.7728
INFO: [COSIM 212-1000] *** C/RTL co-simulation finished: PASS ***
INFO: [COSIM 212-210] Design is translated to an combinational logic. II and Latency
will be marked as all 0.
Finished C/RTL cosimulation.
```

Result

		Latency			Interval		
RTL	Status	min	avg	max	min	avg	max
VHDL	NA	NA	NA	NA	NA	NA	NA
Verilog	Pass	0	0	0	0	0	0

Export the report(.html) using the Export Wizard

2.3.1) *Use HLS stream blocking interface for the ports in your design*

Header file

```
#ifndef MUL32
#define MUL32

#include <iostream>
#include "hls_stream.h"

using namespace std;
using namespace hls;

typedef long long out;
struct inputs{
	int A;
	int B;
};

#endif
```

C++ code

```
#include "mul32.h"

void mul32(stream<inputs> &din,stream<out> &dout){
```

```
        inputs data=din.read();
        dout.write(data.A * data.B);
}
```

Test bench

```
#include "mul32.h"

void mul32(stream<inputs> &din,stream<long long> &dout);
int main(){
        stream<inputs> indata;
        stream<long long> outdata;
        inputs in;
        long long out;
        int i;
        for (i=0;i<10;i++){
                in.A=i+2;
                in.B=i;
                indata.write(in);
                mul32(indata,outdata);
                outdata>>out;
                cout<<in.A<<"X"<<in.B<<"="<< out <<endl;
        }
        return 0;
}
```

Simulation

```
INFO: [SIM 2] *************** CSIM start ***************
INFO: [SIM 4] CSIM will launch GCC as the compiler.
   Compiling ../../../mul32_tb.cpp in debug mode
   Compiling ../../../mul32.cpp in debug mode
   Generating csim.exe
2X0=0
3X1=3
4X2=8
5X3=15
6X4=24
7X5=35
8X6=48
9X7=63
10X8=80
11X9=99
INFO: [SIM 1] CSim done with 0 errors.
INFO: [SIM 3] *************** CSIM finish ***************
```

Synthesis:

Synthesis Report for 'mul32'

General Information

Date:	Sat Mar 18 23:47:56 2023
Version:	2017.4 (Build 2086221 on Fri Dec 15 21:13:33 MST 2017)
Project:	assignment2.1
Solution:	solution1
Product family:	zynq
Target device:	xc7z020clg484-1

Performance Estimates

Timing (ns)

Summary

Clock	Target	Estimated	Uncertainty
ap_clk	10.00	8.51	1.25

Latency (clock cycles)

Summary

Latency		Interval		
min	max	min	max	Type
2	2	2	2	none

Detail

Instance

Loop

Utilization Estimates

Summary

Name	BRAM_18K	DSP48E	FF	LUT
DSP	-	-	-	-
Expression	-	3	0	36
FIFO	-	-	-	-
Instance	-	-	-	-
Memory	-	-	-	-
Multiplexer	-	-	-	48
Register	-	-	99	-
Total	0	3	99	84
Available	280	220	106400	53200
Utilization (%)	0	1	~0	~0

Detail

Instance

DSP48

Memory

FIFO

Expression

Variable Name	Operation	DSP48E	FF	LUT	Bitwidth P0	Bitwidth P1
tmp_1_fu_51_p2	*	3	0	20	32	32
din_V_A0_status	and	0	0	8	1	1
ap_block_state1	or	0	0	8	1	1
Total		3	3	0	36	34	34

Multiplexer

Name	LUT	Input Size	Bits	Total Bits
ap_NS_fsm	21	4	1	4
din_V_A_blk_n	9	2	1	2
din_V_B_blk_n	9	2	1	2
dout_V_blk_n	9	2	1	2
Total	48	10	4	10

Register

Interface

Summary

RTL Ports	Dir	Bits	Protocol	Source Object	C Type
ap_clk	in	1	ap_ctrl_hs	mul32	return value
ap_rst	in	1	ap_ctrl_hs	mul32	return value
ap_start	in	1	ap_ctrl_hs	mul32	return value
ap_done	out	1	ap_ctrl_hs	mul32	return value
ap_idle	out	1	ap_ctrl_hs	mul32	return value
ap_ready	out	1	ap_ctrl_hs	mul32	return value
din_V_A_dout	in	32	ap_fifo	din_V_A	pointer
din_V_A_empty_n	in	1	ap_fifo	din_V_A	pointer
din_V_A_read	out	1	ap_fifo	din_V_A	pointer
din_V_B_dout	in	32	ap_fifo	din_V_B	pointer
din_V_B_empty_n	in	1	ap_fifo	din_V_B	pointer
din_V_B_read	out	1	ap_fifo	din_V_B	pointer
dout_V_din	out	64	ap_fifo	dout_V	pointer
dout_V_full_n	in	1	ap_fifo	dout_V	pointer
dout_V_write	out	1	ap_fifo	dout_V	pointer

Co-simualtion report:

```
INFO: [Common 17-206] Exiting xsim at Sat Mar 18 23:50:44 2023...
INFO: [COSIM 212-316] Starting C post checking ...
2X0=0
3X1=3
4X2=8
5X3=15
6X4=24
7X5=35
8X6=48
9X7=63
10X8=80
11X9=99
INFO: [COSIM 212-1000] *** C/RTL co-simulation finished: PASS ***
Finished C/RTL cosimulation.
```

Cosimulation Report for 'mul32'

Result

		Latency			Interval		
RTL	Status	min	avg	max	min	avg	max
VHDL	NA	NA	NA	NA	NA	NA	NA
Verilog	Pass	2	2	2	3	3	3

Export the report(.html) using the Export Wizard

2.3.1)

Header file

```
#ifndef MULFIX
#define MULFIX
```

```
#include <iostream>
#include "ap_fixed.h"
#include "hls_stream.h"

using namespace std;
using namespace hls;

typedef ap_ufixed<28,4, AP_RND, AP_WRAP> fix28_4;
typedef ap_ufixed<56,8, AP_RND, AP_WRAP> fix56_8;

struct inputs{
	fix28_4 A;
	fix28_4 B;
};
#endif
```

C++ code

```
#include "mulfix.h"
void mulf(stream<inputs> &din,stream<fix56_8> &dout){
	inputs data=din.read();
	dout.write(data.A * data.B);
}
```

Test bench

```
#include "mulfix.h"
void mulf(stream<inputs> &din,stream<fix56_8> &dout);
int main(){
	stream<inputs> indata;
	stream<fix56_8> outdata;
	int i;
	inputs in={0,0};
	fix56_8 out;
	for (i=0;i<10;i++){
		in.A=i+0.6;
		in.B=i+0.5;
		indata.write(in);
		mulf(indata,outdata);
		outdata>>out;
		cout <<in.A<<"X"<<in.B<<"="<<out<< endl;}}
```

Simulation reports

```
INFO: [SIM 2] *************** CSIM start ***************
INFO: [SIM 4] CSIM will launch GCC as the compiler.
   Compiling ../../../mulfix_tb.cpp in debug mode
   Generating csim.exe
0.6X0.5=0.3
1.6X1.5=2.4
2.6X2.5=6.5
3.6X3.5=12.6
4.6X4.5=20.7
5.6X5.5=30.8
6.6X6.5=42.9
7.6X7.5=57
8.6X8.5=73.1
```

```
9.6X9.5=91.2
INFO: [SIM 1] CSim done with 0 errors.
INFO: [SIM 3] *************** CSIM finish ***************
```

Synthesis:

Synthesis Report for 'mulf'

General Information

Date:	Sun Mar 19 00:04:42 2023
Version:	2017.4 (Build 2086221 on Fri Dec 15 21:13:33 MST 2017)
Project:	assignment2.2
Solution:	solution1
Product family:	zynq
Target device:	xc7z020clg484-1

Performance Estimates

Timing (ns)

Summary

Clock	Target	Estimated	Uncertainty
ap_clk	10.00	7.45	1.25

Latency (clock cycles)

Summary

Latency		Interval		
min	max	min	max	Type
2	2	2	2	none

Detail

Instance

Loop

Utilization Estimates

Summary

Name	BRAM_18K	DSP48E	FF	LUT
DSP	-	-	-	-
Expression	-	3	0	52
FIFO	-	-	-	-
Instance	-	-	-	-
Memory	-	-	-	-
Multiplexer	-	-	-	48
Register	-	-	115	-
Total	0	3	115	100
Available	280	220	106400	53200
Utilization (%)	0	1	~0	~0

Detail

Instance

DSP48

Memory

FIFO

Expression

Variable Name	Operation	DSP48E	FF	LUT	Bitwidth P0	Bitwidth P1
p_Val2_s_fu_57_p2	*	3	0	36	28	28
din_V_A_V0_status	and	0	0	8	1	1
ap_block_state1	or	0	0	8	1	1
Total		3	3	0	52	30	30

Multiplexer

Name	LUT	Input Size	Bits	Total Bits
ap_NS_fsm	21	4	1	4
din_V_A_V_blk_n	9	2	1	2
din_V_B_V_blk_n	9	2	1	2
dout_V_V_blk_n	9	2	1	2
Total	48	10	4	10

Register

Interface

Summary

RTL Ports	Dir	Bits	Protocol	Source Object	C Type
ap_clk	in	1	ap_ctrl_hs	mulf	return value
ap_rst	in	1	ap_ctrl_hs	mulf	return value
ap_start	in	1	ap_ctrl_hs	mulf	return value
ap_done	out	1	ap_ctrl_hs	mulf	return value
ap_idle	out	1	ap_ctrl_hs	mulf	return value
ap_ready	out	1	ap_ctrl_hs	mulf	return value
din_V_A_V_dout	in	28	ap_fifo	din_V_A_V	pointer
din_V_A_V_empty_n	in	1	ap_fifo	din_V_A_V	pointer
din_V_A_V_read	out	1	ap_fifo	din_V_A_V	pointer
din_V_B_V_dout	in	28	ap_fifo	din_V_B_V	pointer
din_V_B_V_empty_n	in	1	ap_fifo	din_V_B_V	pointer
din_V_B_V_read	out	1	ap_fifo	din_V_B_V	pointer
dout_V_V_din	out	56	ap_fifo	dout_V_V	pointer
dout_V_V_full_n	in	1	ap_fifo	dout_V_V	pointer
dout_V_V_write	out	1	ap_fifo	dout_V_V	pointer

Cosimulation report:

```
INFO: [Common 17-206] Exiting xsim at Sun Mar 19 00:17:15 2023...
INFO: [COSIM 212-316] Starting C post checking ...
0.6X0.5=0.3
1.6X1.5=2.4
2.6X2.5=6.5
3.6X3.5=12.6
4.6X4.5=20.7
5.6X5.5=30.8
6.6X6.5=42.9
7.6X7.5=57
8.6X8.5=73.1
9.6X9.5=91.2
INFO: [COSIM 212-1000] *** C/RTL co-simulation finished: PASS ***
```

Cosimulation Report for 'mulf'

Result

		Latency			Interval		
RTL	Status	min	avg	max	min	avg	max
VHDL	NA	NA	NA	NA	NA	NA	NA
Verilog	Pass	2	2	2	3	3	3

Export the report(.html) using the Export Wizard

OBSERVATIONS 1:

assignment	Time (ns)	*Resources*	
		LUT	DSP48E
1 (8-bit)	4.1	41	-
2.1 (32-bits)	8.51	20	3

- *From assignment 1 to 2.1 the data-width increased ,so timing increased*
- *Multiplications which are >=10-bits are implemented on DSP48,else LUT's are used*

OBSERVATION2:

Assignment	Time (ns)	Resources	
		LUT	DSP48E
2.1 (32-bits)	8.51	20	3
2.2 (28-bits)	7.45	36	3

- *From assignment 2.1 to 2.2 the data-width decreased ,so timing decreased*
- In assignment 2.2 arbitrary precision data type is used,so number of LUT's increased

OBSERVATION3:

Assignment		Time (ns)	Resources		
			LUT	DSP48	FF
2.1 (32-bits)	Without HLS stream interface	8.51	20	3	-
	With HLS stream interface	8.51	84	3	99
2.2 (28-bits)	Without HLS stream interface	7.45	36	3	-
	With HLS stream interface	7.45	100	3	115

- By using HLS stream interface there is no effect on timing
- By using HLS stream interface the number of resources consumed increased