HLS ASSIGNMENT5(kanekal kousar)

PART A:

(Q) Implement a DUT that accesses 8 elements from BRAM (the BRAM should be contained within the DUT, you can choose to populate the BRAM in any way you like) and gives out all 8 values as HLS streaming output in a single bundle in a single clock cycle. The starting index will be an input to the DUT and it will be a multiple of 8.

1) 8 consecutive elements are to be chosen and these elements will make up a bundle.

Header file

```
#include<hls_stream.h>
using namespace hls;
#include<ap_int.h>
typedef ap_uint<8> bit_width;
struct bundle{
        bit_width   data[8];
};
```

Design code:

```
#include "header.h"

void mem(int index,stream<bundle> &output){
       bit_width
bram[80]={0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,
28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49,50,51,52,53,54,55,
56,57,58,59,60,61,62,63,64,65,66,67,68,69,60,61,62,63,64,65,66,67,68,69};
       #pragma HLS RESOURCE variable=bram core=RAM_1P_BRAM
       #pragma HLS ARRAY_PARTITION variable=bram cyclic factor=8 dim=1
       //accessing 8 consecutive elements from bram
       bundle
out={bram[index],bram[index+1],bram[index+2],bram[index+3],bram[index+4],bram[index+
5],bram[index+6],bram[index+7]};
       output.write(out);
}
```

Test bench:

```
#include "header.h"
using namespace std;
#include <fstream>

void mem(int index,stream<bundle> &output);
int main(){
       stream<bundle> outdata;
       int index,a,b,c,d,e,f,g,h;
       ifstream in;
       fstream out_d;
       in.open("ref.dat");
       out_d.open("result.dat");
       int fail=0;
```

```
	while (in>>index>>a>>b>>c>>d>>e>>f>>g>>h){
		mem(index,outdata);
		bundle out=outdata.read();
		for (int i=0;i<8;i++){
			out_d<<out.data[i]<<"\t";
		}
		if (out.data[0]==a & out.data[1]==b & out.data[2]==c &
out.data[3]==d &out.data[4]==e & out.data[5]==f & out.data[6]==g &
out.data[7]==h ){
			out_d<<"passed"<<endl;
		}
		else{
			out_d<<"fails"<<endl;
			fail++;
		}
	}
		in.close();
		out_d.close();
	if (fail==0){
		cout<<"all test cases passed"<<endl;
	}
	else{
		cout<<fail<<"\t test case failed"<<endl;
	}}
```

Synthesis report:

Synthesis Report for 'mem'

General Information

Date:	Sun Apr 2 13:04:04 2023
Version:	2017.4 (Build 2086221 on Fri Dec 15 21:13:33 MST 2017)
Project:	assignment5_prac
Solution:	solution1
Product family:	kintexuplus
Target device:	xcku5p-ffva676-1-i

Performance Estimates

Timing (ns)

Summary

Clock	Target	Estimated	Uncertainty
ap_clk	10.00	5.17	1.25

Latency (clock cycles)

Summary

Latency		Interval		
min	max	min	max	Type
8	8	8	8	none

Detail

Instance

Loop

Utilization Estimates

Summary

Name	BRAM_18K	DSP48E	FF	LUT	URAM
DSP	-	-	-	-	-
Expression	-	-	0	769	-
FIFO	-	-	-	-	-
Instance	-	-	-	-	-
Memory	8	-	0	0	-
Multiplexer	-	-	-	471	-
Register	-	-	97	-	-
Total	8	0	97	1240	0
Available	960	1824	433920	216960	64
Utilization (%)	~0	0	~0	~0	0

Detail

Instance

DSP48

Memory

Memory	Module	BRAM_18K	FF	LUT	Words	Bits	Banks	W*Bits*Banks
bram_0_U	mem_bram_0	1	0	0	9	7	1	63
bram_1_U	mem_bram_1	1	0	0	9	7	1	63
bram_2_U	mem_bram_2	1	0	0	9	7	1	63
bram_3_U	mem_bram_3	1	0	0	9	7	1	63
bram_4_U	mem_bram_4	1	0	0	9	7	1	63
bram_5_U	mem_bram_5	1	0	0	9	7	1	63
bram_6_U	mem_bram_6	1	0	0	8	6	1	48
bram_7_U	mem_bram_7	1	0	0	8	6	1	48
Total		8	0	0	70	54	8	474

FIFO

Expression

Multiplexer

Register

Interface

Summary

RTL Ports	Dir	Bits	Protocol	Source Object	C Type
ap_clk	in	1	ap_ctrl_hs	mem	return value
ap_rst	in	1	ap_ctrl_hs	mem	return value
ap_start	in	1	ap_ctrl_hs	mem	return value
ap_done	out	1	ap_ctrl_hs	mem	return value
ap_idle	out	1	ap_ctrl_hs	mem	return value
ap_ready	out	1	ap_ctrl_hs	mem	return value
index	in	32	ap_none	index	scalar
output_V_data0_V_din	out	8	ap_fifo	output_V_data0_V	pointer
output_V_data0_V_full_n	in	1	ap_fifo	output_V_data0_V	pointer
output_V_data0_V_write	out	1	ap_fifo	output_V_data0_V	pointer
output_V_data1_V_din	out	8	ap_fifo	output_V_data1_V	pointer
output_V_data1_V_full_n	in	1	ap_fifo	output_V_data1_V	pointer
output_V_data1_V_write	out	1	ap_fifo	output_V_data1_V	pointer
output_V_data2_V_din	out	8	ap_fifo	output_V_data2_V	pointer
output_V_data2_V_full_n	in	1	ap_fifo	output_V_data2_V	pointer
output_V_data2_V_write	out	1	ap_fifo	output_V_data2_V	pointer
output_V_data3_V_din	out	8	ap_fifo	output_V_data3_V	pointer
output_V_data3_V_full_n	in	1	ap_fifo	output_V_data3_V	pointer
output_V_data3_V_write	out	1	ap_fifo	output_V_data3_V	pointer
output_V_data4_V_din	out	8	ap_fifo	output_V_data4_V	pointer
output_V_data4_V_full_n	in	1	ap_fifo	output_V_data4_V	pointer
output_V_data4_V_write	out	1	ap_fifo	output_V_data4_V	pointer
output_V_data5_V_din	out	8	ap_fifo	output_V_data5_V	pointer
output_V_data5_V_full_n	in	1	ap_fifo	output_V_data5_V	pointer
output_V_data5_V_write	out	1	ap_fifo	output_V_data5_V	pointer
output_V_data6_V_din	out	8	ap_fifo	output_V_data6_V	pointer
output_V_data6_V_full_n	in	1	ap_fifo	output_V_data6_V	pointer
output_V_data6_V_write	out	1	ap_fifo	output_V_data6_V	pointer
output_V_data7_V_din	out	8	ap_fifo	output_V_data7_V	pointer
output_V_data7_V_full_n	in	1	ap_fifo	output_V_data7_V	pointer
output_V_data7_V_write	out	1	ap_fifo	output_V_data7_V	pointer

Export the report(.html) using the Export Wizard

Open Analysis Perspective Analysis Perspective

Simulation:

```
INFO: [SIM 2] *************** CSIM start ***************
INFO: [SIM 4] CSIM will launch GCC as the compiler.
make: `csim.exe' is up to date.
all test cases passed
```

```
INFO: [SIM 1] CSim done with 0 errors.
INFO: [SIM 3] *************** CSIM finish ***************
```

Ref file

```
8 8 9 10 11 12 13 14 15
16 16 17 18 19 20 21 22
```

Result file

```
8     9     10    11    12    13    14    15    passed
16    17    18    19    20    21    22    23    passed
```

Cosimulatio:

```
INFO: [Common 17-206] Exiting xsim at Sun Apr  2 15:15:01 2023...
INFO: [COSIM 212-316] Starting C post checking ...
all test cases passed
INFO: [COSIM 212-1000] *** C/RTL co-simulation finished: PASS ***
INFO: [COSIM 212-211] II is measurable only when transaction number is greater than 1 in
RTL simulation. Otherwise, they will be marked as all NA. If user wants to calculate
them, please make sure there are at least 2 transactions in RTL simulation.
Finished C/RTL cosimulation.
```

Cosimulation Report for 'mem'

Result

		Latency			Interval		
RTL	Status	min	avg	max	min	avg	max
VHDL	NA	NA	NA	NA	NA	NA	NA
Verilog	Pass	8	8	8	NA	NA	NA

Export the report(.html) using the Export Wizard

I have not used Complete array_partition because,after sysnthesis its not partitioning the blocksRAMs into smaller blocks RAMs

2) Every 8th element is to be chosen and 8 such elements will make up a bundle

Header file:

```
#include<hls_stream.h>
using namespace hls;
#include<ap_int.h>
typedef ap_uint<8> bit_width;
struct bundle{
        bit_width   data[8];
};
```

Design code:

```
#include "hea.h"

void mem(int index,stream<bundle> &output){
	bit_width
bram[80]={0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,3
	#pragma HLS RESOURCE variable=bram core=RAM_1P_BRAM
	#pragma HLS ARRAY_PARTITION variable=bram block factor=10 dim=1
	//accessing 8 consecutive elements from bram
	bundle out={bram[index],bram[index+8],bram[index+16],bram[index+24],bram[index+32],bram[inc
	output.write(out);
}
```

Testbench code:

```
#include "hea.h"
using namespace std;
#include <fstream>

void mem(int index,stream<bundle> &output);
int main(){
	stream<bundle> outdata;
	int index,a,b,c,d,e,f,g,h;
	ifstream in;
	fstream out_d;
	in.open("in.dat");
	out_d.open("out.dat");
	int fail=0;
	while (in>>index>>a>>b>>c>>d>>e>>f>>g>>h){
		mem(index,outdata);
		bundle out=outdata.read();
		for (int i=0;i<8;i++){
			out_d<<out.data[i]<<"\t";
		}
		if (out.data[0]==a & out.data[1]==b & out.data[2]==c & out.data[3]==d
&out.data[4]==e & out.data[5]==f & out.data[6]==g & out.data[7]==h ){
			out_d<<"passed"<<endl;
		}
		else{
			out_d<<"fails"<<endl;
			fail++;
		}
	}
		in.close();
		out_d.close();
	if (fail==0){
		cout<<"all test cases passed"<<endl;
	}
	else{
		cout<<fail<<"\t test case failed"<<endl;
	}
}
```

Synthesis report:

Synthesis Report for 'mem'

General Information

Date:	Sun Apr 2 15:33:06 2023
Version:	2017.4 (Build 2086221 on Fri Dec 15 21:13:33 MST 2017
Project:	assignment5.2
Solution:	solution1
Product family:	zynq
Target device:	xc7z020clg484-1

Performance Estimates

Timing (ns)

Summary

Clock	Target	Estimated	Uncertainty
ap_clk	10.00	5.88	1.25

Latency (clock cycles)

Summary

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

Detail

- Instance
- Loop

Utilization Estimates

Summary

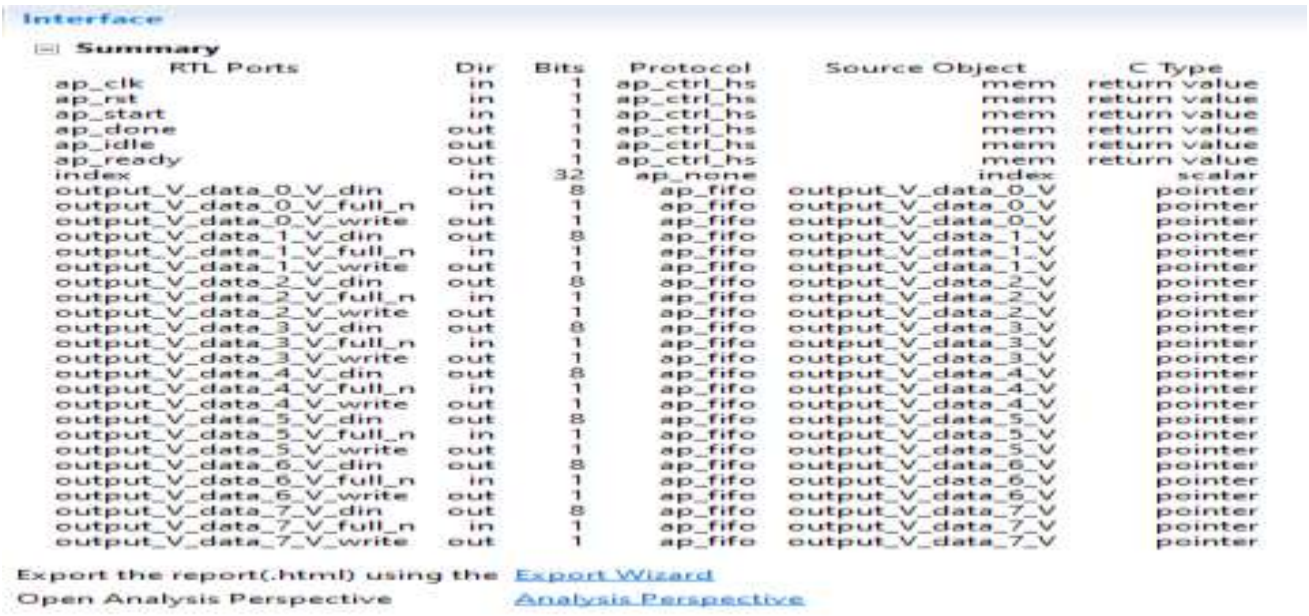
Name	BRAM_18K	DSP48E	FF	LUT
DSP	-	-	-	-
Expression	-	-	0	281
FIFO	-	-	-	-
Instance	-	-	0	440
Memory	10	-	0	0
Multiplexer	-	-	-	93
Register	-	-	96	-
Total	10	0	96	814
Available	280	220	106400	53200
Utilization (%)	3	0	~0	1

Detail

- Instance
- DSP48
- Memory

Memory	Module	BRAM_18K	FF	LUT	Words	Bits	Banks	W*Bits*Banks
bram_0_U	mem_bram_0	1	0	0	8	3	1	24
bram_1_U	mem_bram_1	1	0	0	8	4	1	32
bram_3_U	mem_bram_1	1	0	0	8	4	1	32
bram_7_U	mem_bram_1	1	0	0	8	4	1	32
bram_2_U	mem_bram_2	1	0	0	8	5	1	40
bram_6_U	mem_bram_2	1	0	0	8	5	1	40
bram_4_U	mem_bram_4	1	0	0	8	6	1	48
bram_5_U	mem_bram_5	1	0	0	8	6	1	48
bram_8_U	mem_bram_8	1	0	0	8	7	1	56
bram_9_U	mem_bram_9	1	0	0	8	7	1	56
Total		10	10	0	0	80	51	10	408

- FIFO
- Expression
- Multiplexer
- Register



Interface

Summary

RTL Ports	Dir	Bits	Protocol	Source Object	C Type
ap_clk	in	1	ap_ctrl_hs	mem	return value
ap_rst	in	1	ap_ctrl_hs	mem	return value
ap_start	in	1	ap_ctrl_hs	mem	return value
ap_done	out	1	ap_ctrl_hs	mem	return value
ap_idle	out	1	ap_ctrl_hs	mem	return value
ap_ready	out	1	ap_ctrl_hs	mem	return value
index	in	32	ap_none	index	scalar
output_V_data_0_V_din	out	8	ap_fifo	output_V_data_0_V	pointer
output_V_data_0_V_full_n	in	1	ap_fifo	output_V_data_0_V	pointer
output_V_data_0_V_write	out	1	ap_fifo	output_V_data_0_V	pointer
output_V_data_1_V_din	out	8	ap_fifo	output_V_data_1_V	pointer
output_V_data_1_V_full_n	in	1	ap_fifo	output_V_data_1_V	pointer
output_V_data_1_V_write	out	1	ap_fifo	output_V_data_1_V	pointer
output_V_data_2_V_din	out	8	ap_fifo	output_V_data_2_V	pointer
output_V_data_2_V_full_n	in	1	ap_fifo	output_V_data_2_V	pointer
output_V_data_2_V_write	out	1	ap_fifo	output_V_data_2_V	pointer
output_V_data_3_V_din	out	8	ap_fifo	output_V_data_3_V	pointer
output_V_data_3_V_full_n	in	1	ap_fifo	output_V_data_3_V	pointer
output_V_data_3_V_write	out	1	ap_fifo	output_V_data_3_V	pointer
output_V_data_4_V_din	out	8	ap_fifo	output_V_data_4_V	pointer
output_V_data_4_V_full_n	in	1	ap_fifo	output_V_data_4_V	pointer
output_V_data_4_V_write	out	1	ap_fifo	output_V_data_4_V	pointer
output_V_data_5_V_din	out	8	ap_fifo	output_V_data_5_V	pointer
output_V_data_5_V_full_n	in	1	ap_fifo	output_V_data_5_V	pointer
output_V_data_5_V_write	out	1	ap_fifo	output_V_data_5_V	pointer
output_V_data_6_V_din	out	8	ap_fifo	output_V_data_6_V	pointer
output_V_data_6_V_full_n	in	1	ap_fifo	output_V_data_6_V	pointer
output_V_data_6_V_write	out	1	ap_fifo	output_V_data_6_V	pointer
output_V_data_7_V_din	out	8	ap_fifo	output_V_data_7_V	pointer
output_V_data_7_V_full_n	in	1	ap_fifo	output_V_data_7_V	pointer
output_V_data_7_V_write	out	1	ap_fifo	output_V_data_7_V	pointer

Export the report(.html) using the Export Wizard

Open Analysis Perspective Analysis Perspective

Simulation:

```
INFO: [SIM 2] *************** CSIM start ***************
INFO: [SIM 4] CSIM will launch GCC as the compiler.
   Compiling ../../../mem_tb.cpp in debug mode
   Compiling ../../../mem.cpp in debug mode
   Generating csim.exe
all test cases passed
INFO: [SIM 1] CSim done with 0 errors.
INFO: [SIM 3] *************** CSIM finish ***************
```

Ref file

```
8 8 16 24 32 40 48 56 64
16 16 24 32 40 48 56 64 62
```

result file

```
8 16 24 32   40     48     56     64     passed
16 24  32 40  48     56     64     62     passed
```

Cosimulation:

```
INFO: [Common 17-206] Exiting xsim at Sun Apr  2 20:08:41 2023...

INFO: [COSIM 212-316] Starting C post checking ...

all test cases passed

INFO: [COSIM 212-1000] *** C/RTL co-simulation finished: PASS ***

Finished C/RTL cosimulation.
```

Cosimulation Report for 'mem'

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

PART-B)

(Q)implement a DUT that takes 4 inputs in 4 clock cycles saves all the inputs in BRAM at a single address.at the same time it also gives out all the 4 inputs as a single bundle output in the same clock cycle when it recieves the final input.the BRAM index will overflow after the bram is full and it should overwrite the old values as more inputs coming in.

Header file:

```
#include<hls_stream.h>
#include <ap_int.h>
#include<iostream>
#include <fstream>
using namespace std;
using namespace hls;


typedef ap_uint<8> WIDTH;
typedef ap_uint<3> addr_t;
const int DEPTH=8;
const int ELEMENTS=4;


struct bundle{
WIDTH data[ELEMENTS];
};
```

Design code:

```
#include "header.h"

void arr_re(stream<bundle> &in, stream<bundle> &out){
	bundle input=in.read();
	bundle output={0,0,0,0};
	WIDTH bram[DEPTH];
#pragma HLS RESOURCE variable=bram core=RAM_1P_BRAM
#pragma HLS ARRAY_RESHAPE variable=bram cyclic factor=4 dim=1
	addr_t count=0,add=0;
	int ind=0;
	//storing elements in bram
	for (int i=0;i<ELEMENTS+(ELEMENTS/4)-1;i++){
```

```
            if (count<DEPTH){
                  bram[add]=input.data[ind];
                  count++;
                  add++;
                  ind++;
                  //accessing elements from bram
                  if (count==4){
                        output.data[0]=bram[0];
                        output.data[1]=bram[1];
                        output.data[2]=bram[2];
                        output.data[3]=bram[3];
                  }
            }
            else{
                  //if number of elements >DEPTH ,storing from starting index
                  count=0;
                  add=0;
            }
      }
      out<<output;
}
```

Test bench:

```
#include "header.h"
void arr_re(stream<bundle> &in, stream<bundle> &out);
int main(){
      stream<bundle> indata;
      stream<bundle> outdata;
      bundle input;
      ifstream in;
      fstream out_d;
      in.open("ref.dat");
      out_d.open("result.dat");
      //access elements form ref.file and store in input
      int i,index=0,fail=0;
      while (in>>i && index<ELEMENTS){
      input.data[index]=i;
      index++;
      }
      indata<<input;
      arr_re(indata,outdata);
      bundle output=outdata.read();
      for (int j=0;j<4;j++){
            out_d<<output.data[j]<<"\t";
      }
      if (input.data[ELEMENTS-4]==output.data[0] & input.data[ELEMENTS-
3]==output.data[1] & input.data[ELEMENTS-2]==output.data[2] & input.data[ELEMENTS-
1]==output.data[3]  ){
            out_d<<"passed"<<endl;
      }
      else{
            out_d<<"fail"<<endl;
            fail++;
```

```
	}
	in.close();
	out_d.close();
	if (fail==0){
		cout<<"all passed"<<endl;
	}
	else{
		cout<<"failed"<<endl;
	}
	}
```

Synthesis report:

Synthesis Report for 'arr_re'

General Information

Date:	Sun Apr 2 20:52:18 2023
Version:	2017.4 (Build 2086221 on Fri Dec 15 21:13:33 MST 2017)
Project:	assignment5.3
Solution:	solution1
Product family:	zynq
Target device:	xc7z020clg484-1

Performance Estimates

Timing (ns)

Summary

Clock	Target	Estimated	Uncertainty
ap_clk	10.00	7.88	1.25

Latency (clock cycles)

Summary

Latency		Interval		
min	max	min	max	Type
9	9	9	9	none

Detail

Instance

Loop

Utilization Estimates

Summary

Name	BRAM_18K	DSP48E	FF	LUT
DSP	-	-	-	-
Expression	-	-	0	691
FIFO	-	-	-	-
Instance	-	-	0	21
Memory	1	-	0	0
Multiplexer	-	-	-	138
Register	-	-	87	-
Total	1	0	87	850
Available	280	220	106400	53200
Utilization (%)	~0	0	~0	1

Detail

Instance

DSP48

Memory

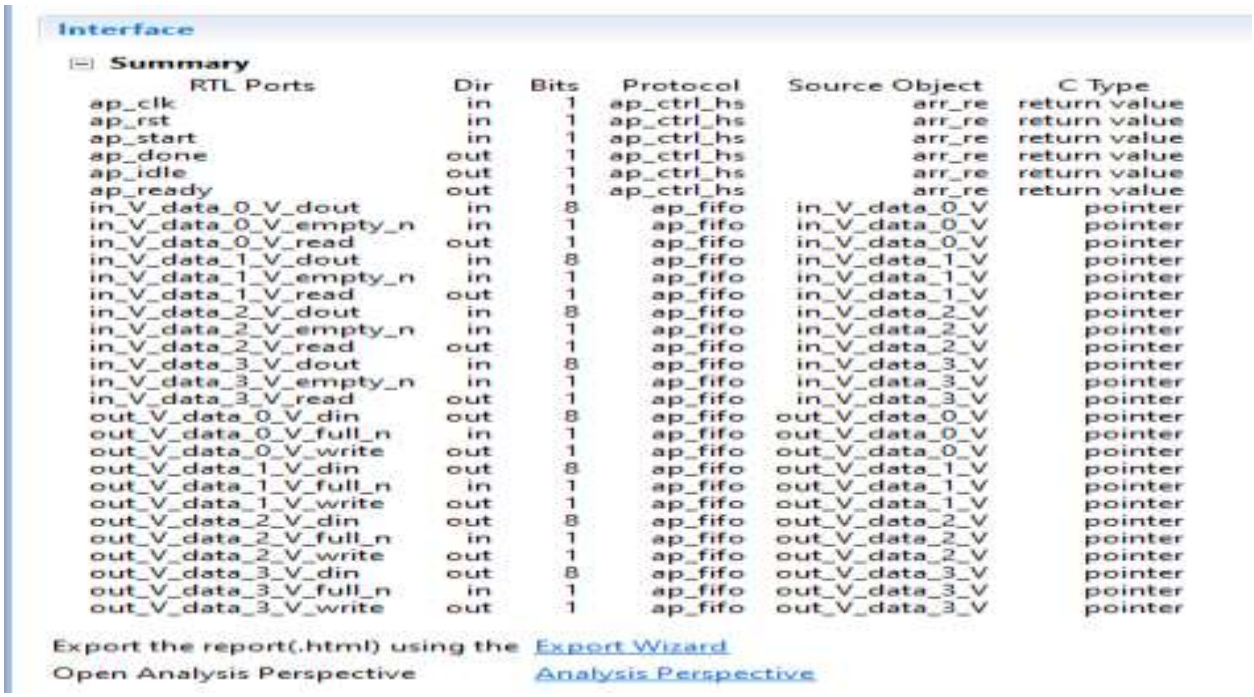
Memory	Module	BRAM_18K	FF	LUT	Words	Bits	Banks	W*Bits*Banks
bram_V_U	arr_re_bram_V	1	0	0	2	32	1	64
Total		1	0	0	2	32	1	64

FIFO

Expression

Multiplexer

Register



Interface

Summary

RTL Ports	Dir	Bits	Protocol	Source Object	C Type
ap_clk	in	1	ap_ctrl_hs	arr_re	return value
ap_rst	in	1	ap_ctrl_hs	arr_re	return value
ap_start	in	1	ap_ctrl_hs	arr_re	return value
ap_done	out	1	ap_ctrl_hs	arr_re	return value
ap_idle	out	1	ap_ctrl_hs	arr_re	return value
ap_ready	out	1	ap_ctrl_hs	arr_re	return value
in_V_data_0_V_dout	in	8	ap_fifo	in_V_data_0_V	pointer
in_V_data_0_V_empty_n	in	1	ap_fifo	in_V_data_0_V	pointer
in_V_data_0_V_read	out	1	ap_fifo	in_V_data_0_V	pointer
in_V_data_1_V_dout	in	8	ap_fifo	in_V_data_1_V	pointer
in_V_data_1_V_empty_n	in	1	ap_fifo	in_V_data_1_V	pointer
in_V_data_1_V_read	out	1	ap_fifo	in_V_data_1_V	pointer
in_V_data_2_V_dout	in	8	ap_fifo	in_V_data_2_V	pointer
in_V_data_2_V_empty_n	in	1	ap_fifo	in_V_data_2_V	pointer
in_V_data_2_V_read	out	1	ap_fifo	in_V_data_2_V	pointer
in_V_data_3_V_dout	in	8	ap_fifo	in_V_data_3_V	pointer
in_V_data_3_V_empty_n	in	1	ap_fifo	in_V_data_3_V	pointer
in_V_data_3_V_read	out	1	ap_fifo	in_V_data_3_V	pointer
out_V_data_0_V_din	out	8	ap_fifo	out_V_data_0_V	pointer
out_V_data_0_V_full_n	in	1	ap_fifo	out_V_data_0_V	pointer
out_V_data_0_V_write	out	1	ap_fifo	out_V_data_0_V	pointer
out_V_data_1_V_din	out	8	ap_fifo	out_V_data_1_V	pointer
out_V_data_1_V_full_n	in	1	ap_fifo	out_V_data_1_V	pointer
out_V_data_1_V_write	out	1	ap_fifo	out_V_data_1_V	pointer
out_V_data_2_V_din	out	8	ap_fifo	out_V_data_2_V	pointer
out_V_data_2_V_full_n	in	1	ap_fifo	out_V_data_2_V	pointer
out_V_data_2_V_write	out	1	ap_fifo	out_V_data_2_V	pointer
out_V_data_3_V_din	out	8	ap_fifo	out_V_data_3_V	pointer
out_V_data_3_V_full_n	in	1	ap_fifo	out_V_data_3_V	pointer
out_V_data_3_V_write	out	1	ap_fifo	out_V_data_3_V	pointer

Export the report(.html) using the Export Wizard

Open Analysis Perspective Analysis Perspective

Simulation :

```
INFO: [SIM 2] *************** CSIM start ***************
INFO: [SIM 4] CSIM will launch GCC as the compiler.
make: `csim.exe' is up to date.
all passed
INFO: [SIM 1] CSim done with 0 errors.
INFO: [SIM 3] *************** CSIM finish ***************
```

In file

```
1       2       3       5
```

out file

```
1       2       3       5       passed
```

Cosimulation:

```
INFO: [Common 17-206] Exiting xsim at Sun Apr  2 20:58:45 2023...
INFO: [COSIM 212-316] Starting C post checking ...
all passed
INFO: [COSIM 212-1000] *** C/RTL co-simulation finished: PASS ***
INFO: [COSIM 212-211] II is measurable only when transaction number is
greater than 1 in RTL simulation. Otherwise, they will be marked as all NA.
If user wants to calculate them, please make sure there are at least 2
transactions in RTL simulation.
Finished C/RTL cosimulation.
```

Cosimulation Report for 'arr_re'

Result

		Latency			Interval		
RTL	Status	min	avg	max	min	avg	max
VHDL	NA	NA	NA	NA	NA	NA	NA
Verilog	Pass	9	9	9	NA	NA	NA

Export the report(.html) using the Export Wizard