***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);  } |

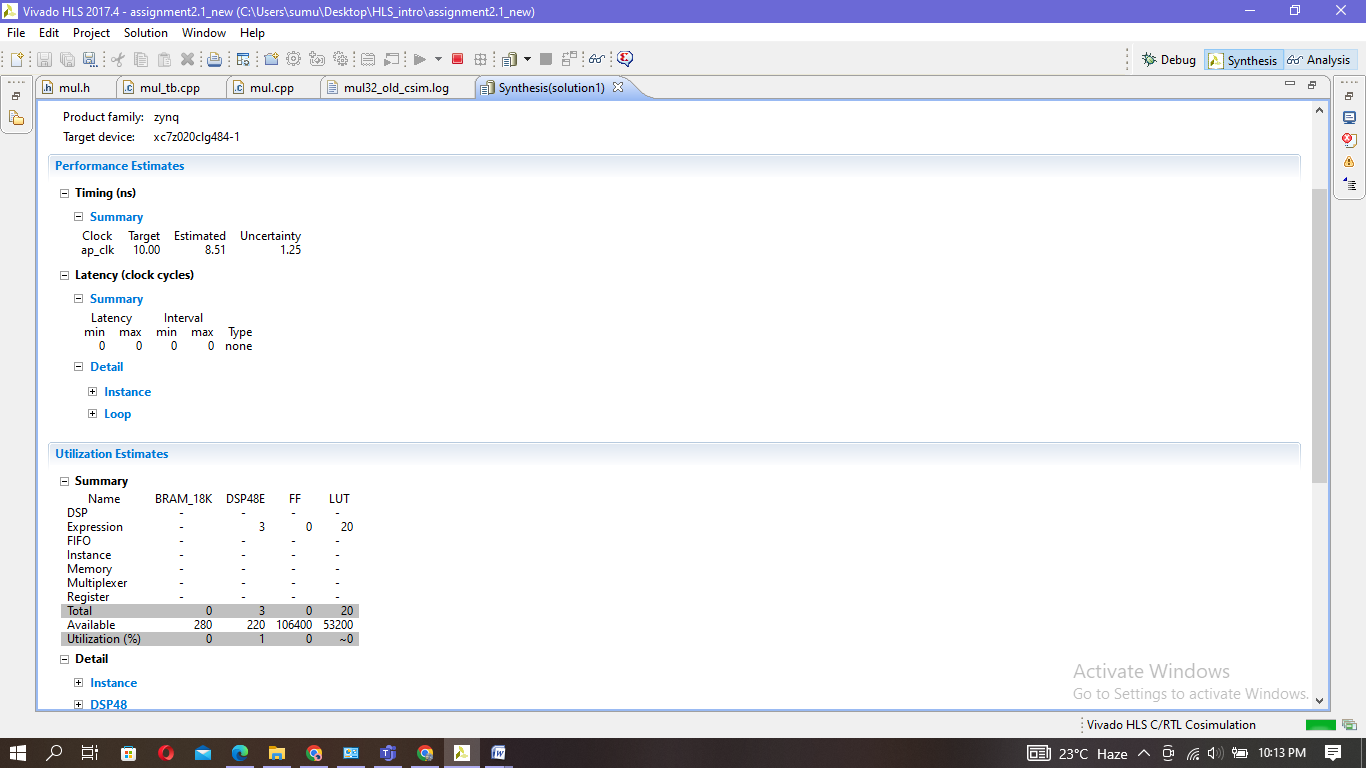
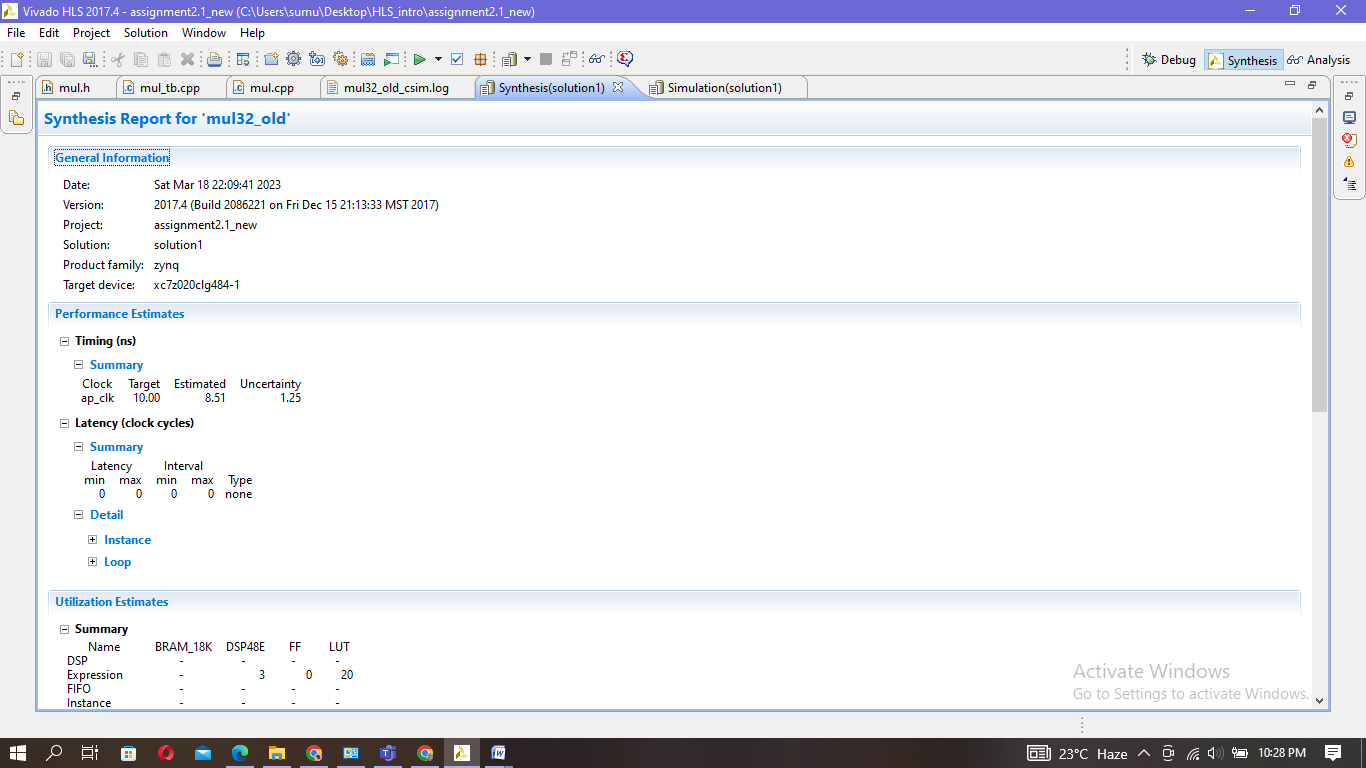
Testbench

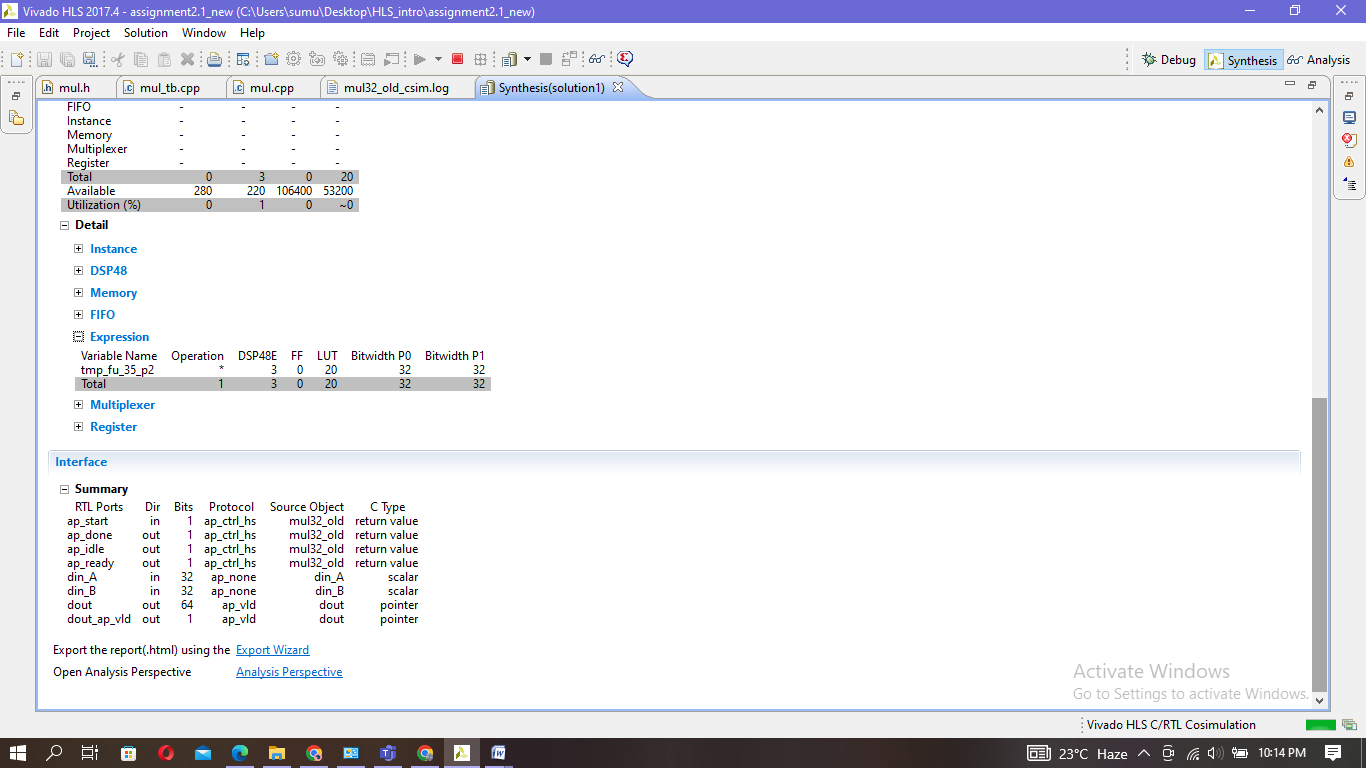
|  |
| --- |
| **#include** "mul.h"  **void** **mul32\_old**(inputs din,out &dout);  **int** **main**(){  inputs in;  out out;  **int** i;  **for** (i=0;i<10;i++){  in.A=i+2;  in.B=i;  mul32\_old(in,out);  cout<<in.A<<"X"<<in.B<<"="<< out <<**endl**;  }  **return** 0;  } |

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 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\* |

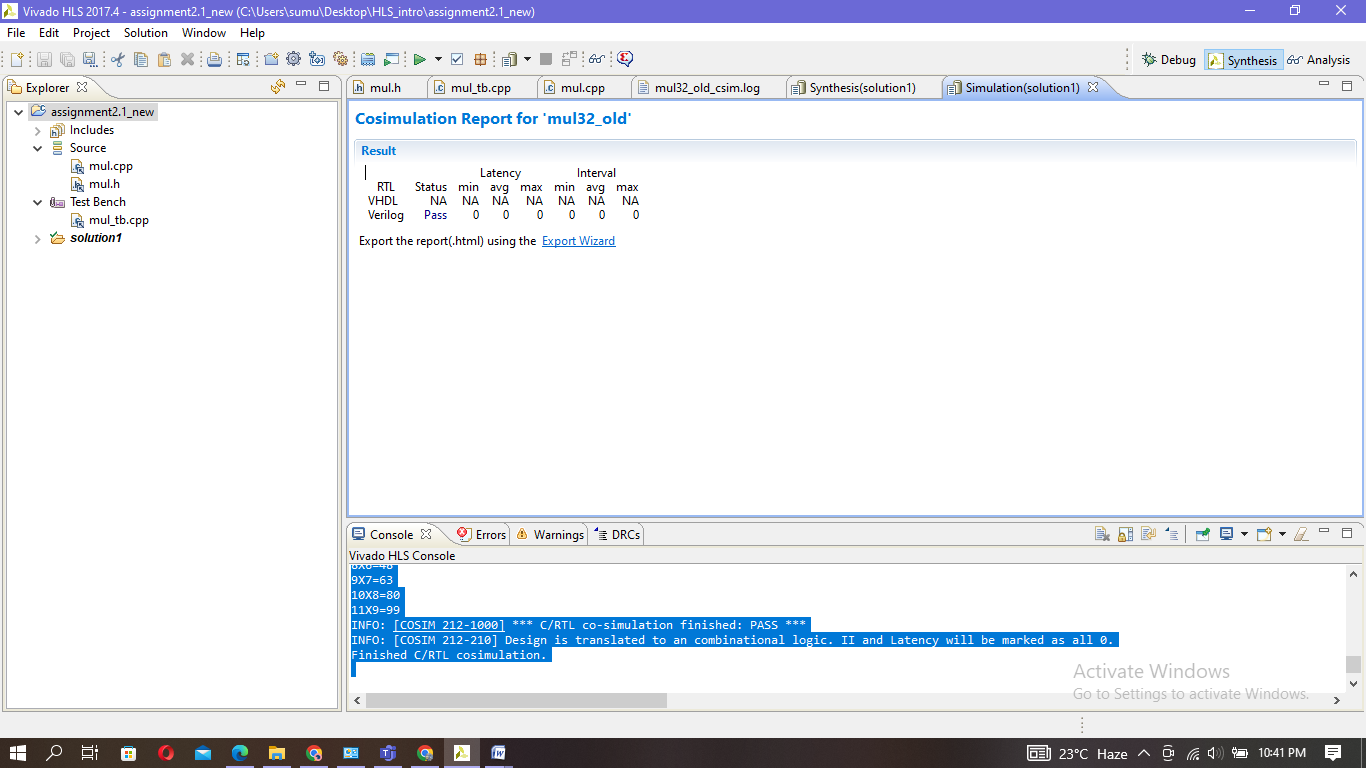
Synthesis report:





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. |



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.

|  |
| --- |
| **#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** |

Header file

C++ code

|  |
| --- |
| **#include** "mul.h"  **void** **mulf**(inputs din,fix56\_8 &dout){  dout=din.A\*din.B;  } |

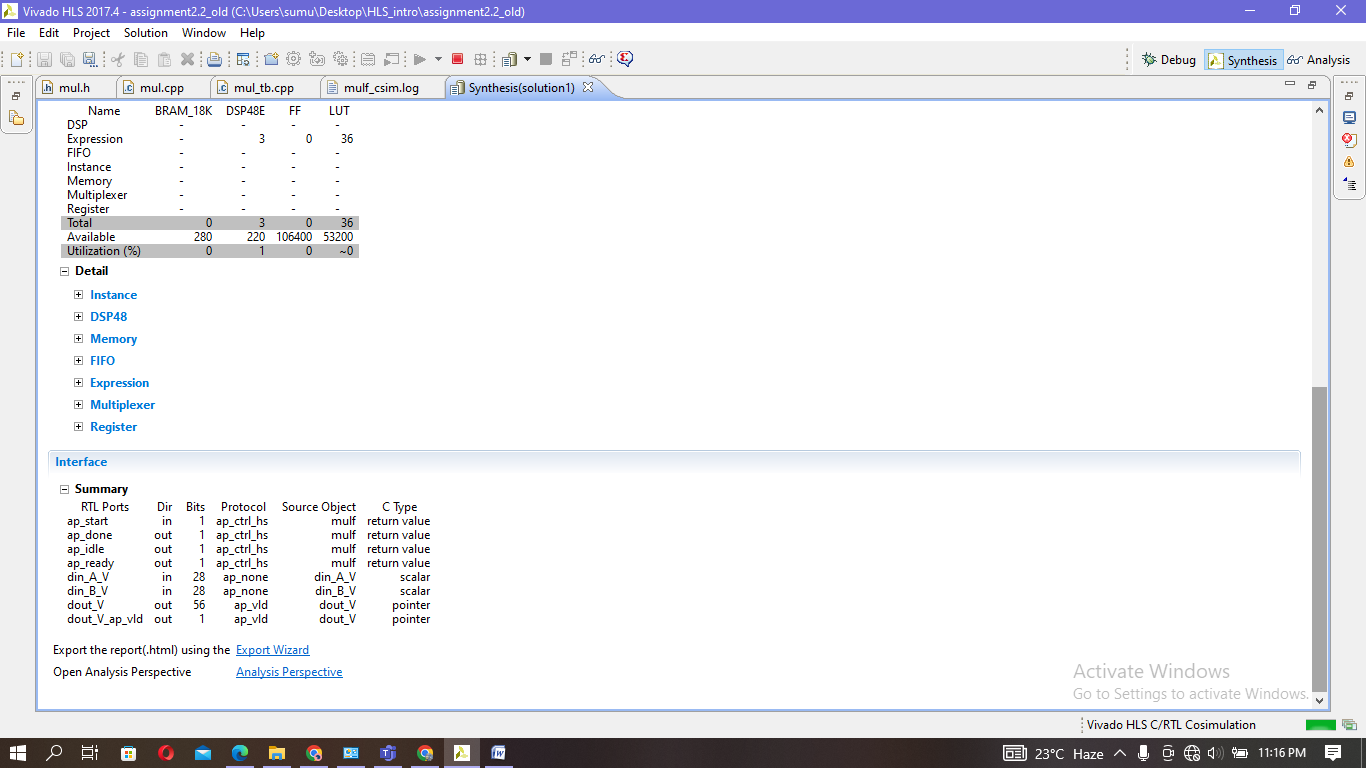
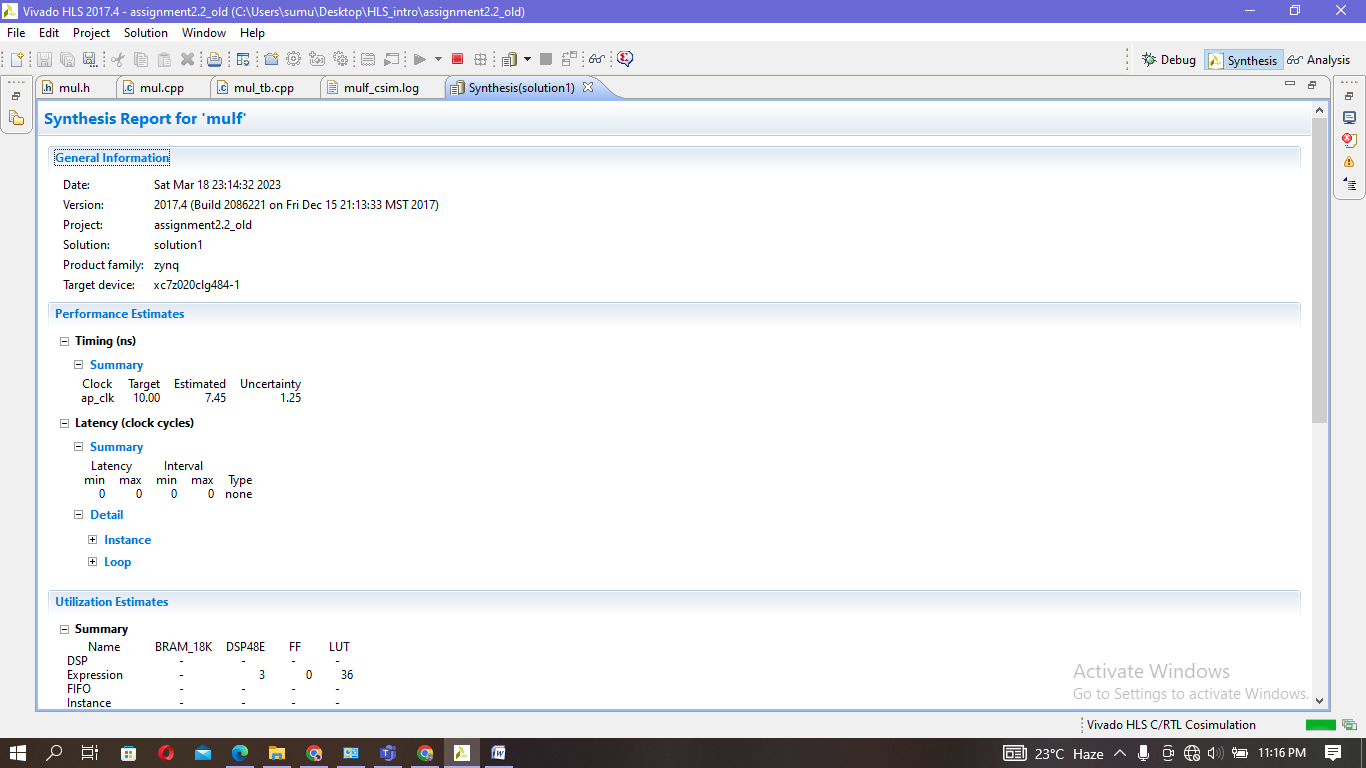
Test bench

|  |
| --- |
| **#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

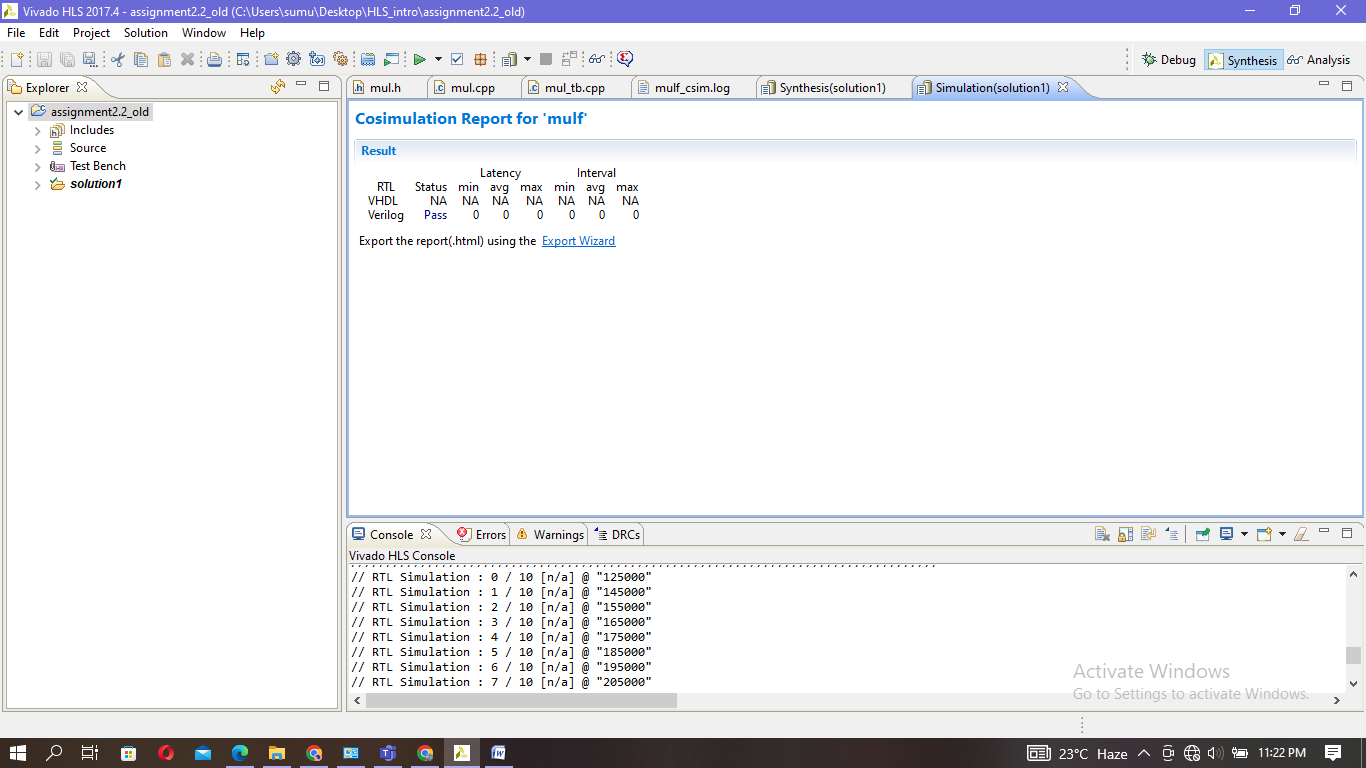
|  |
| --- |
| INFO: [SIM 2] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\* CSIM start \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  INFO: [SIM 4] CSIM will launch GCC as the compiler.  Compiling ../../../mul\_tb.cpp in debug mode  Compiling ../../../mul.cpp in debug mode  Generating csim.exe  0.64X0.52=0.3328  1.64X1.52=2.4928  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



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. |



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

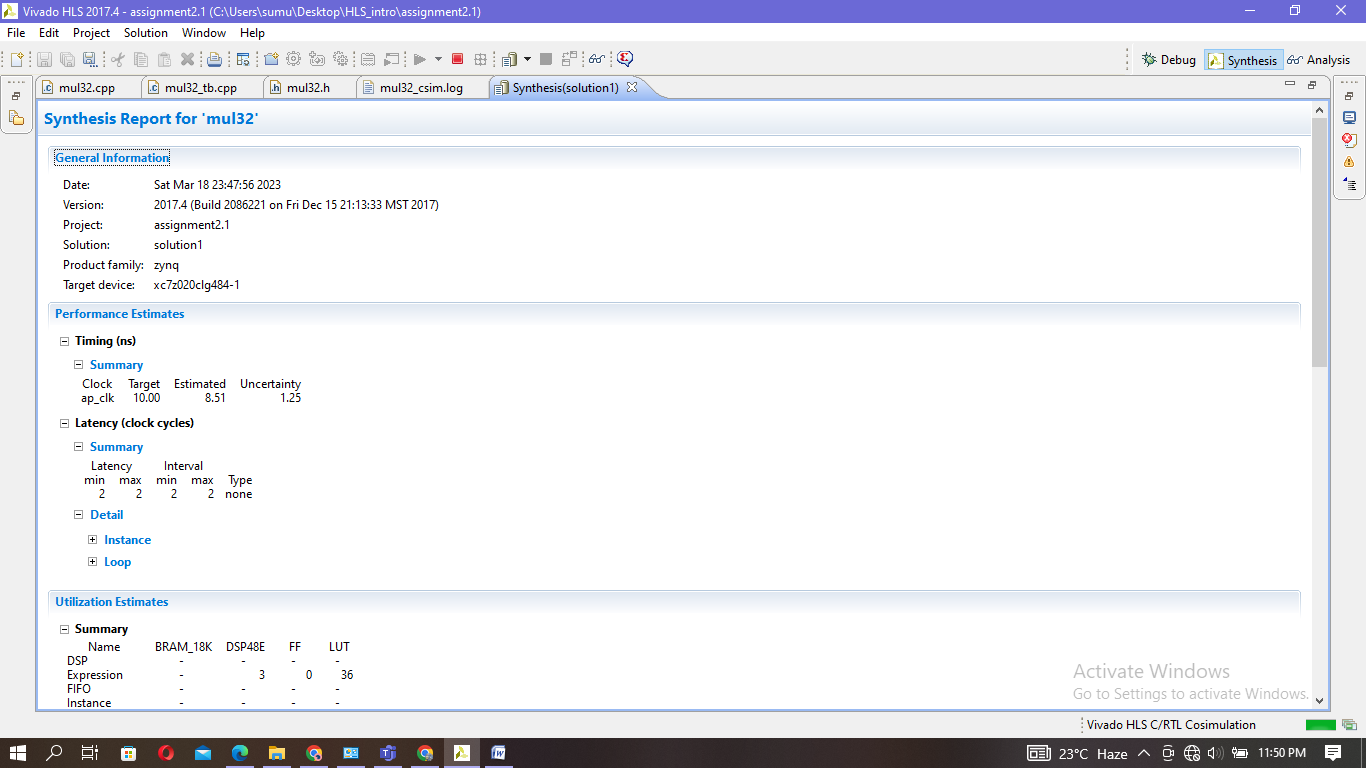
|  |
| --- |
| **#include** "mul32.h"  **void** **mul32**(stream<inputs> &din,stream<out> &dout){  inputs data=din.read();  dout.write(data.A \* data.B);  } |

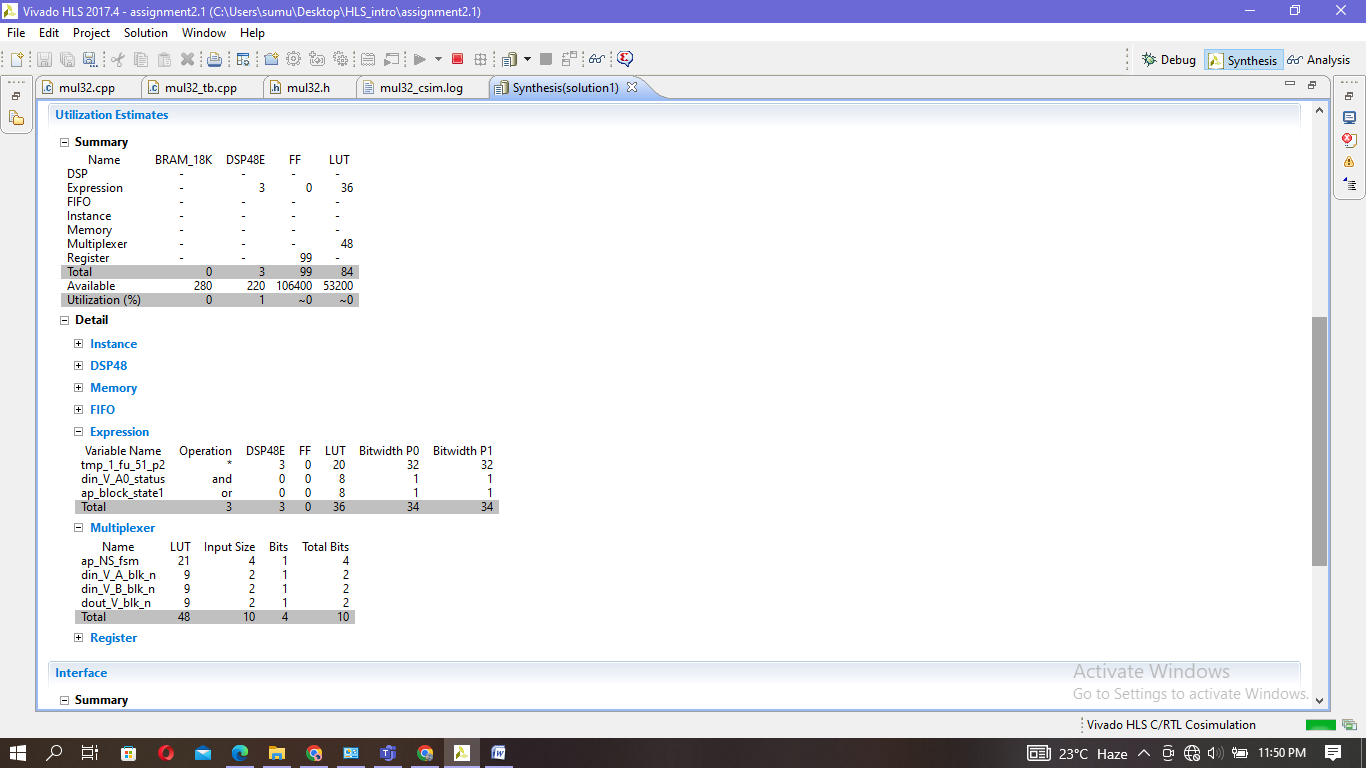
Test bench

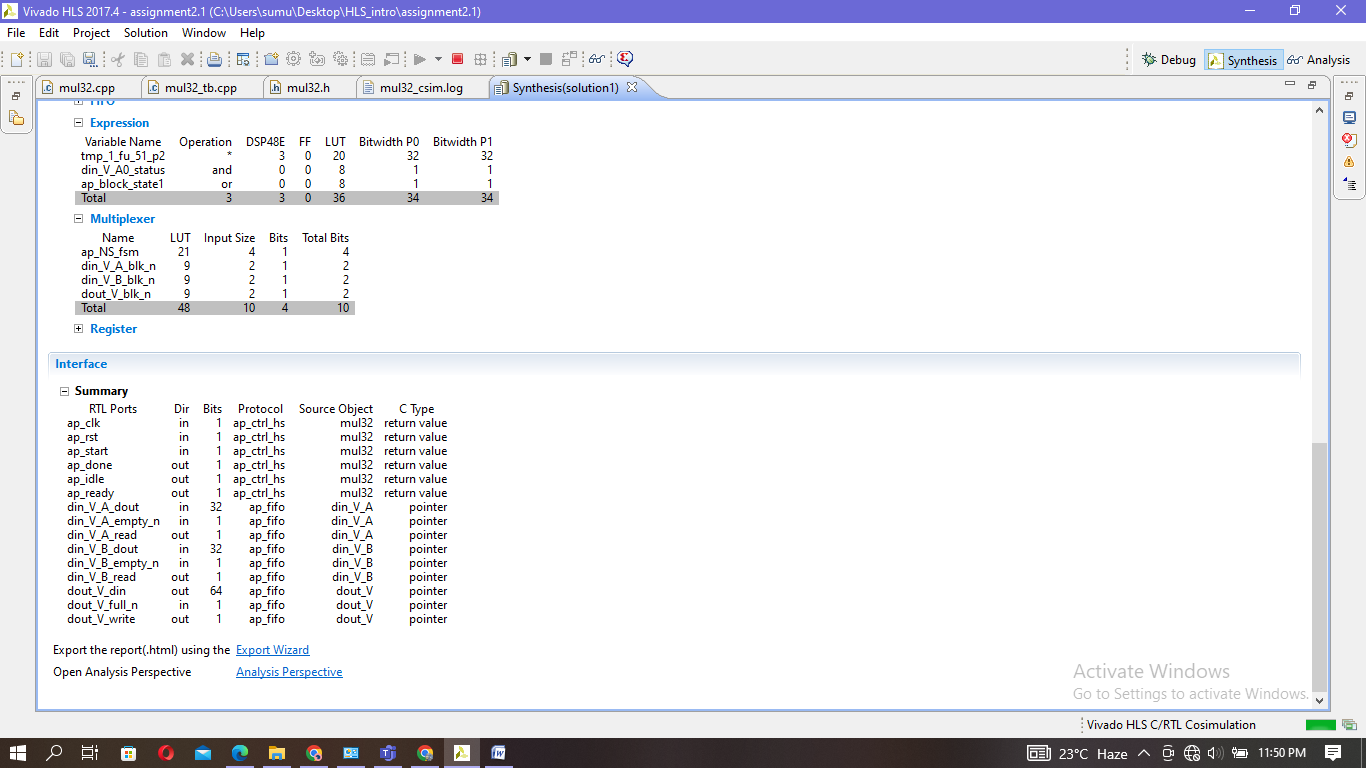
|  |
| --- |
| **#include** "mul32.h"  **void** **mul32**(stream<inputs> &din,stream<**long** **long**> &dout);  **int** **main**(){  stream<inputs> indata;  stream<**long** **long**> outdata;  inputs in;  **long** **long** out;  **int** i;  **for** (i=0;i<10;i++){  in.A=i+2;  in.B=i;  indata.write(in);  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 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\* |

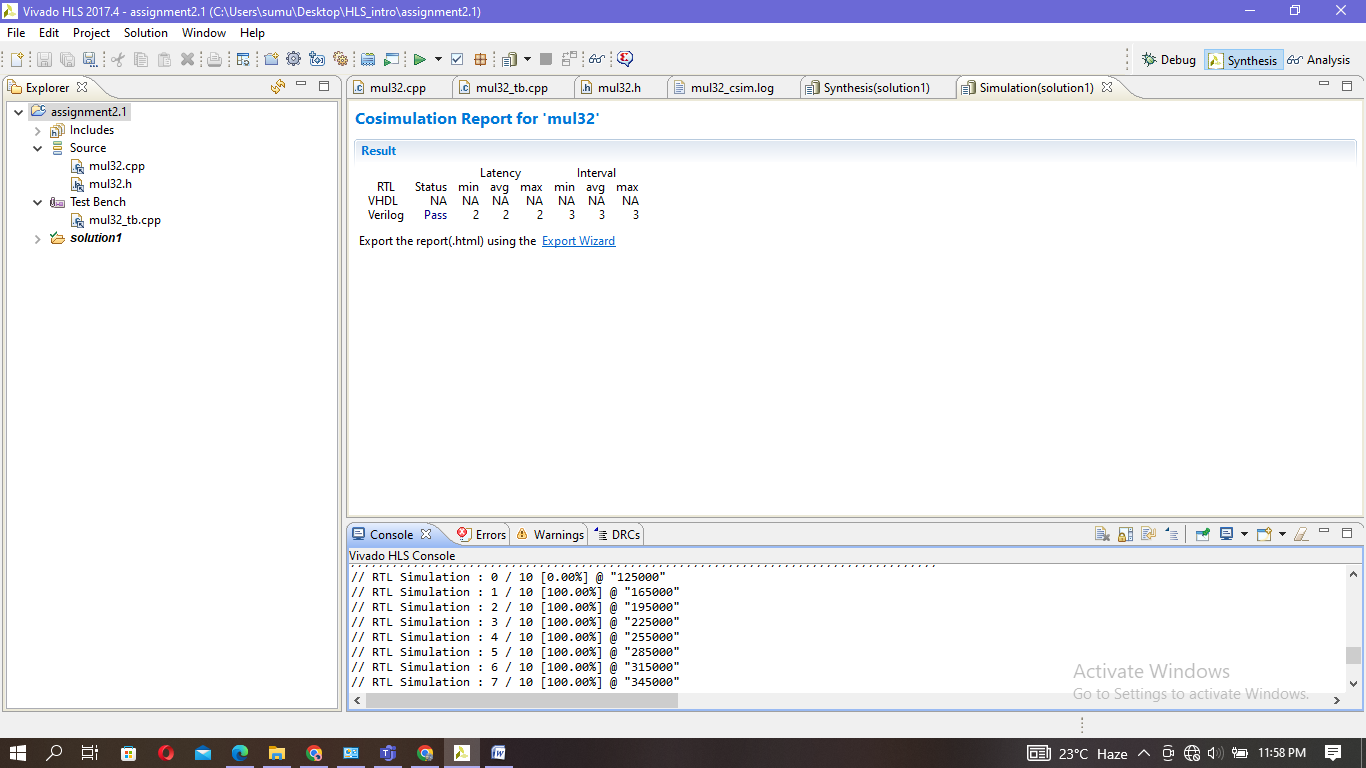
Synthesis:





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. |



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);  } |

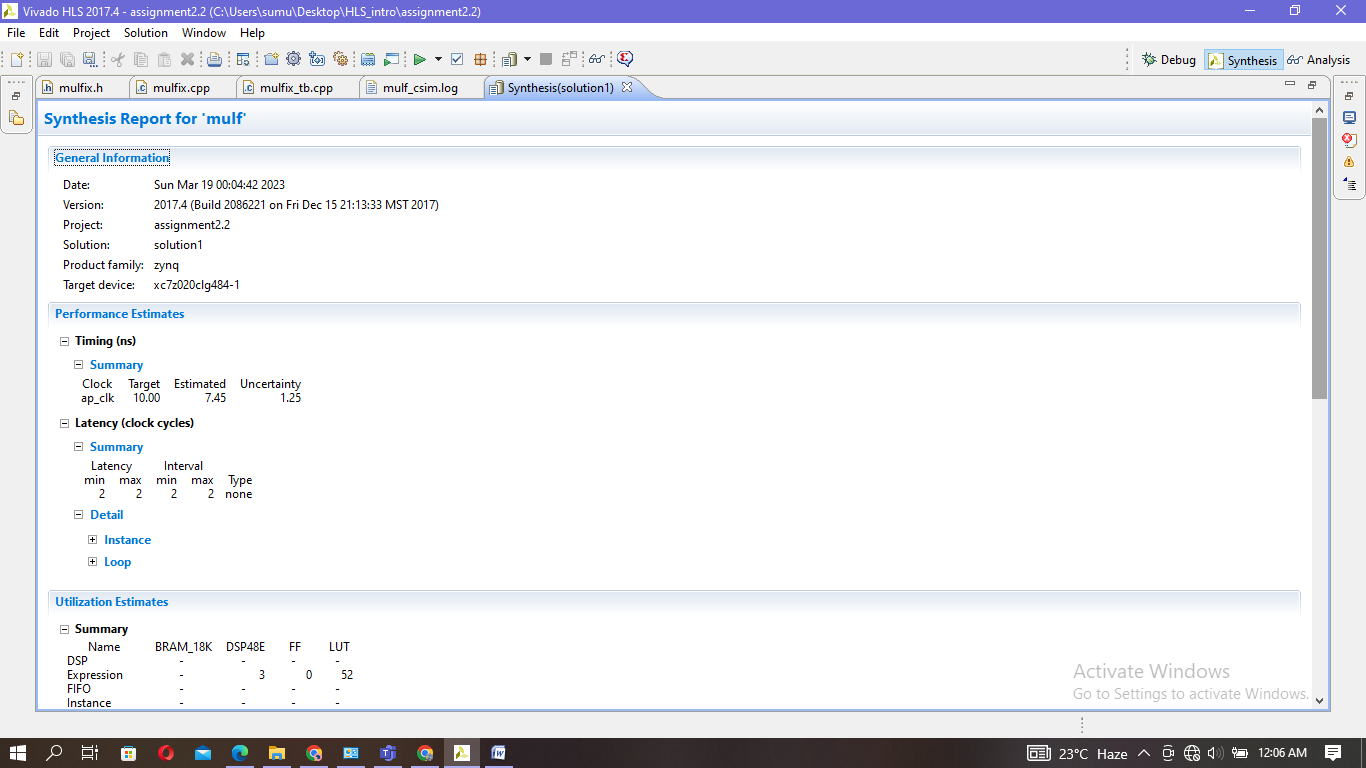
Test bench

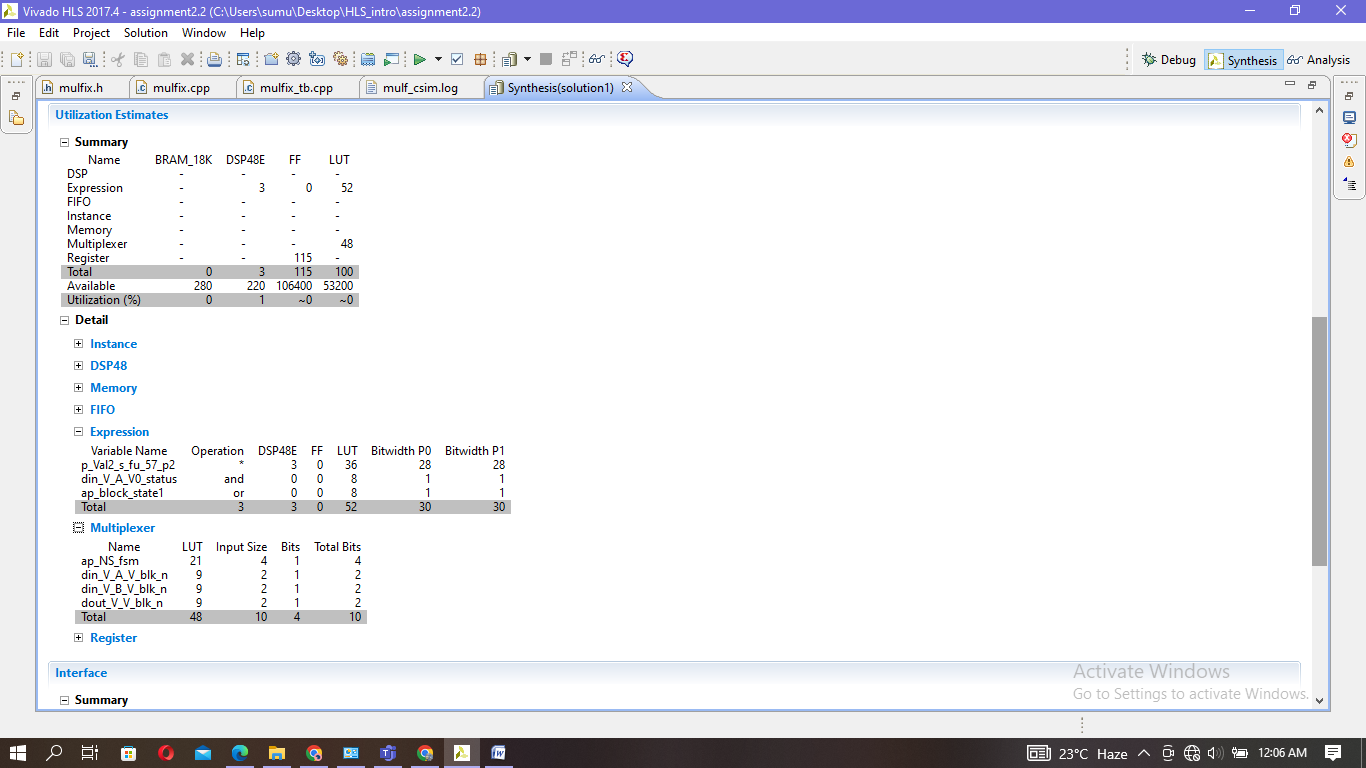
|  |
| --- |
| **#include** "mulfix.h"  **void** **mulf**(stream<inputs> &din,stream<fix56\_8> &dout);  **int** **main**(){  stream<inputs> indata;  stream<fix56\_8> outdata;  **int** i;  inputs in={0,0};  fix56\_8 out;  **for** (i=0;i<10;i++){  in.A=i+0.6;  in.B=i+0.5;  indata.write(in);  mulf(indata,outdata);  outdata>>out;  cout <<in.A<<"X"<<in.B<<"="<<out<< **endl**;}} |

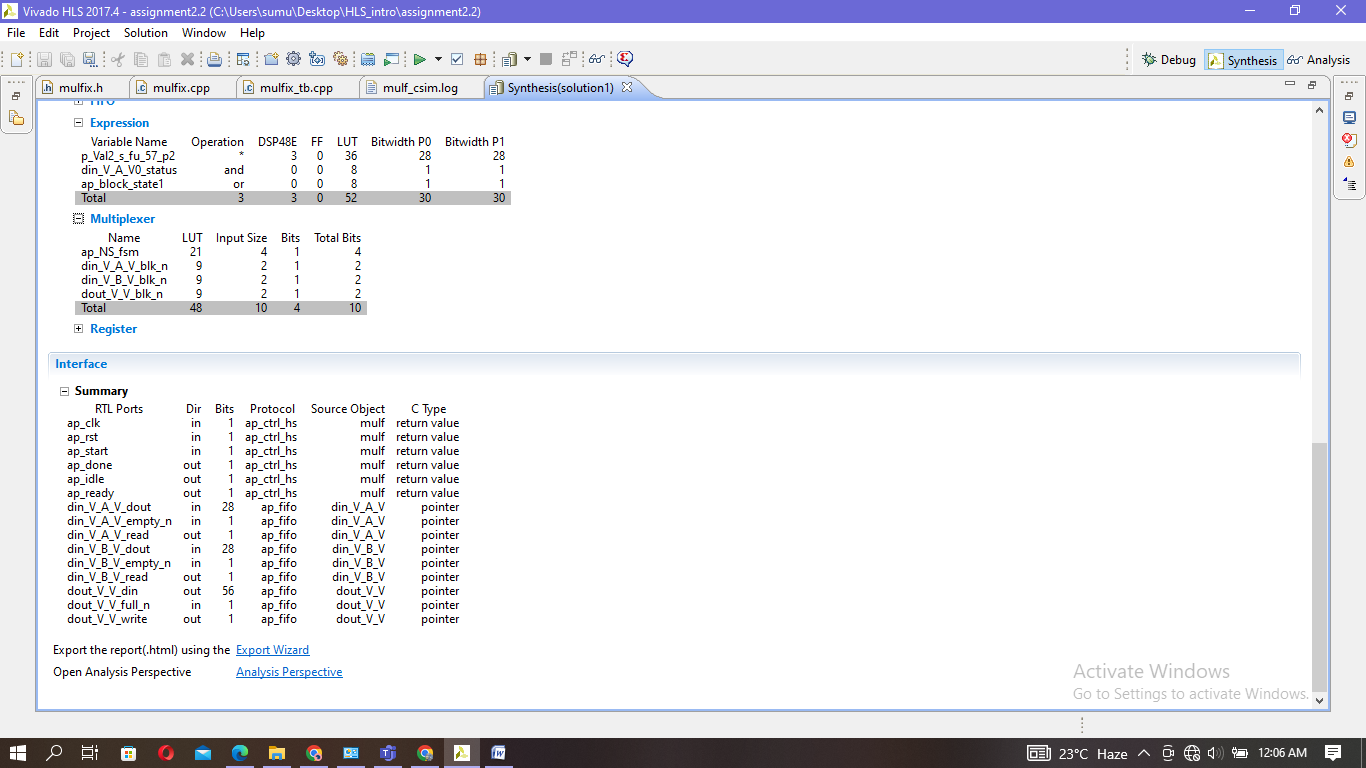
Simulation reports

|  |
| --- |
| 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:







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 \*\*\* | |
| Screenshot (355).png | |

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