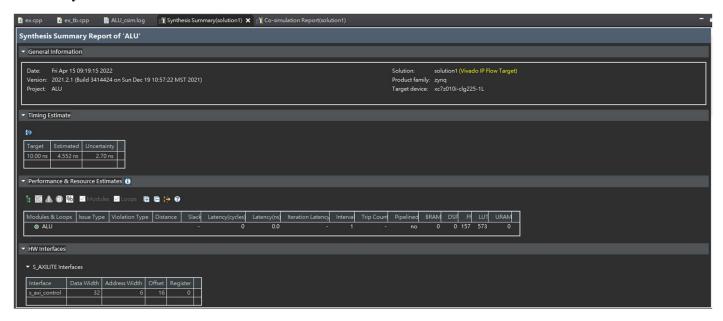
EE 3220 – System-on-Chip Design Assignment 3

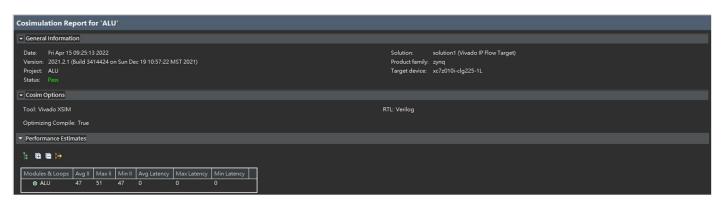
Run C Simulation

```
Co-simulation Report(solution1)
.c ex.cpp
        c ex_tb.cpp
                 2 INFO: [SIM 4] CSIM will launch GCC as the compiler.
   Compiling ../../src/ex_tb.cpp in debug mode
   Compiling ../../src/ex.cpp in debug mode
   Generating csim.exe
 6 mode: 0
 7 mode: 1
 8 mode: 2
 9 mode: 3
10 mode: 4
11 mode: 5
12 mode: 6
13 mode: 7
14 mode: 8
```

Run C Synthesis



Run Co-simulation



ALU description:

Mode 000: Add, Y = A + B

Mode 010: Subtract, Y = A - B

Mode 010: Decrement, Y = A - 1

Mode 011: Increment, Y = A + 1

Mode 100: 1's complement, Y = not A

Mode 101: Bitwise AND, Y = A and B

Mode 110: Bitwise OR, Y = A or B

Mode 111: Bitwise XOR, Y = A xor B

Mode default: Return 0, Y = 0

Question 4:

The model design was written in software with C++ and the software will test, verify the design of the IP or the FPGA and translate the code to Verilog or VHDL.

Advantage:

Productivity gain, the design period can be shortened with using HLS as the code are generated by the software instead writing the code from scratch. In the tutorials, it is much faster to develop a complicated design with HLS then using VHDL.

Optimization, the code will be more optimized by the software. From the tutorial, it is shown that the software will optimize the code instead of optimize by the programmer which requires a lot of experience on FPGA designs.

Appendix:

ex.cpp

```
#include "ex.h"
void ALU(int a, int b, int &y, short mode)
#pragma HLS INTERFACE s_axilite port = return
#pragma HLS INTERFACE s_axilite port = a
#pragma HLS INTERFACE s_axilite port = b
#pragma HLS INTERFACE s_axilite port = y
    switch (mode)
    case 0:
        y = a + b;
       break;
    case 1:
       y = a - b;
        break;
    case 2:
       y = a - 1;
        break;
    case 3:
        y = a + 1;
        break;
    case 4:
       y = \sim a;
        break;
    case 5:
        y = a \& b;
        break;
    case 6:
        y = a \mid b;
        break;
    case 7:
        y = a ^ b;
        break;
    default:
        y = 0;
        break;
```

ex.h

```
#ifndef __EX__
#define __EX__

void ALU(int a, int b, int &y, short mode);
#endif
```

```
#include <iostream>
#include "ex.h"
using namespace std;
int main()
    int a;
    int b;
    int y;
    int gy;
    for (short mode = 0; mode <= 8; mode++)</pre>
        cout << "mode: " << mode << endl;</pre>
        for (a = -100; a < 100; a++)
             for (b = -100; b \le 100; b++)
                 switch (mode)
                 case 0:
                     gy = a + b;
                     break;
                 case 1:
                     gy = a - b;
                     break;
                 case 2:
                     gy = a - 1;
                     break;
                 case 3:
                     gy = a + 1;
                     break;
                 case 4:
                     gy = \sim a;
                     break;
                 case 5:
                     gy = a \& b;
                     break;
                 case 6:
                     gy = a \mid b;
                     break;
                 case 7:
                     gy = a ^ b;
                     break;
                 default:
                     gy = 0;
                     break;
                 ALU(a, b, y, mode);
                 if (gy != y)
```

```
{
        cout << "Error when a=" << a << ", b=" << b;
        return 1;
      }
    }
}
return 0;
}</pre>
```