National Cheng Kung University

Department of Electrical Engineering

Introduction to VLSI CAD (Spring 2022)

Lab Session 3

Design of ALU and Fixed point Multiplication

Name	Student ID	
陳慕丞	E94096097	
Practical Sections:	Points	Marks
Prob A	20	
Prob B	30	
Prob C	30	
Report	15	
File hierarchy, naming…etc.	5	
Notes		

Due Date: 15:00, March 08, 2023 @ moodle

Deliverables

1) All Verilog codes including testbenches for each problem should be uploaded.
   NOTE: Please **DO NOT** include source code in the paper report!
2) All homework requirements should be uploaded in this file hierarchy or you will not get the full credit.
   NOTE: Please **DO NOT** upload waveforms!
3) Important! TA will use the command in Appendix A to check your design under SoC Lab environment, if your code can not be recompiled by TA successfully using the commands, you will not get the full credit.
4) If you upload a dead body which we can't even compile you will get **NO** credit!
5) All Verilog file should get at least 90% superLint Coverage.
6) File hierarchy should not be changed; it may cause your code can not be recompiled by TA successfully using the autograding commands

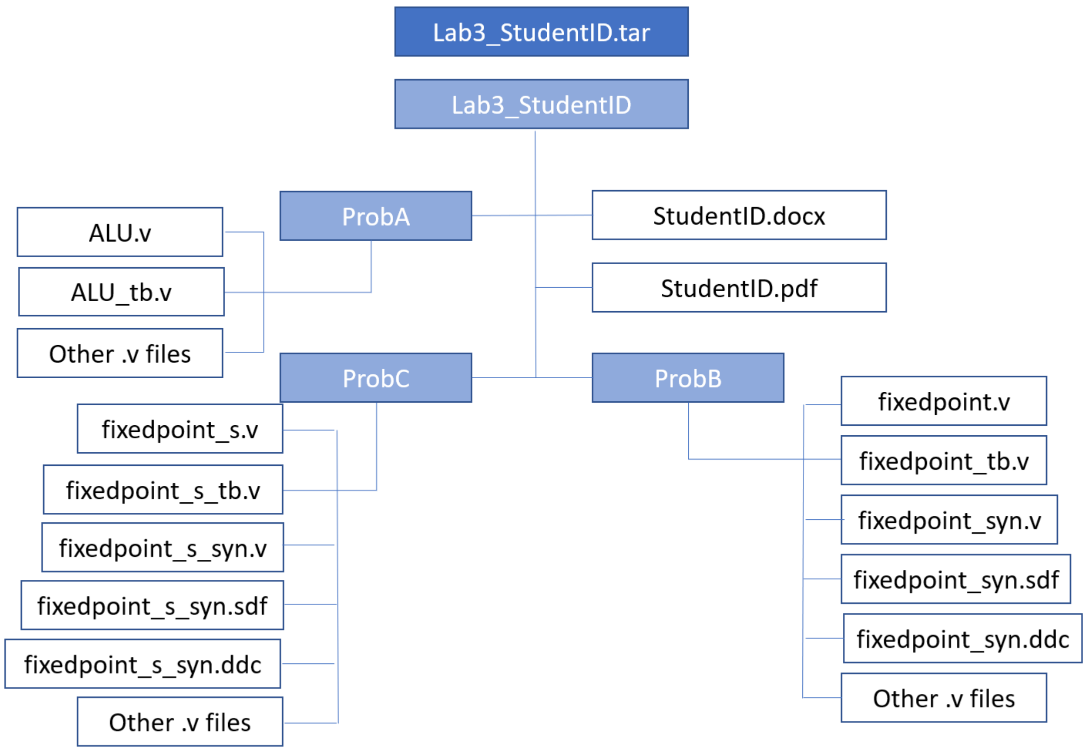
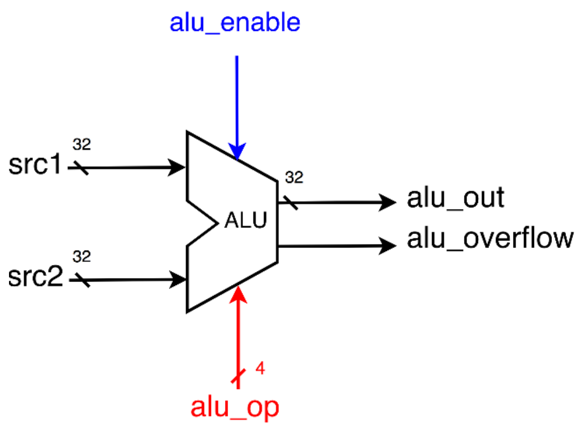


Fig.1 File hierarchy for Homework submission

Objectives:

Learn how to design an ALU.

Prob A: Arithmetic Logic Unit



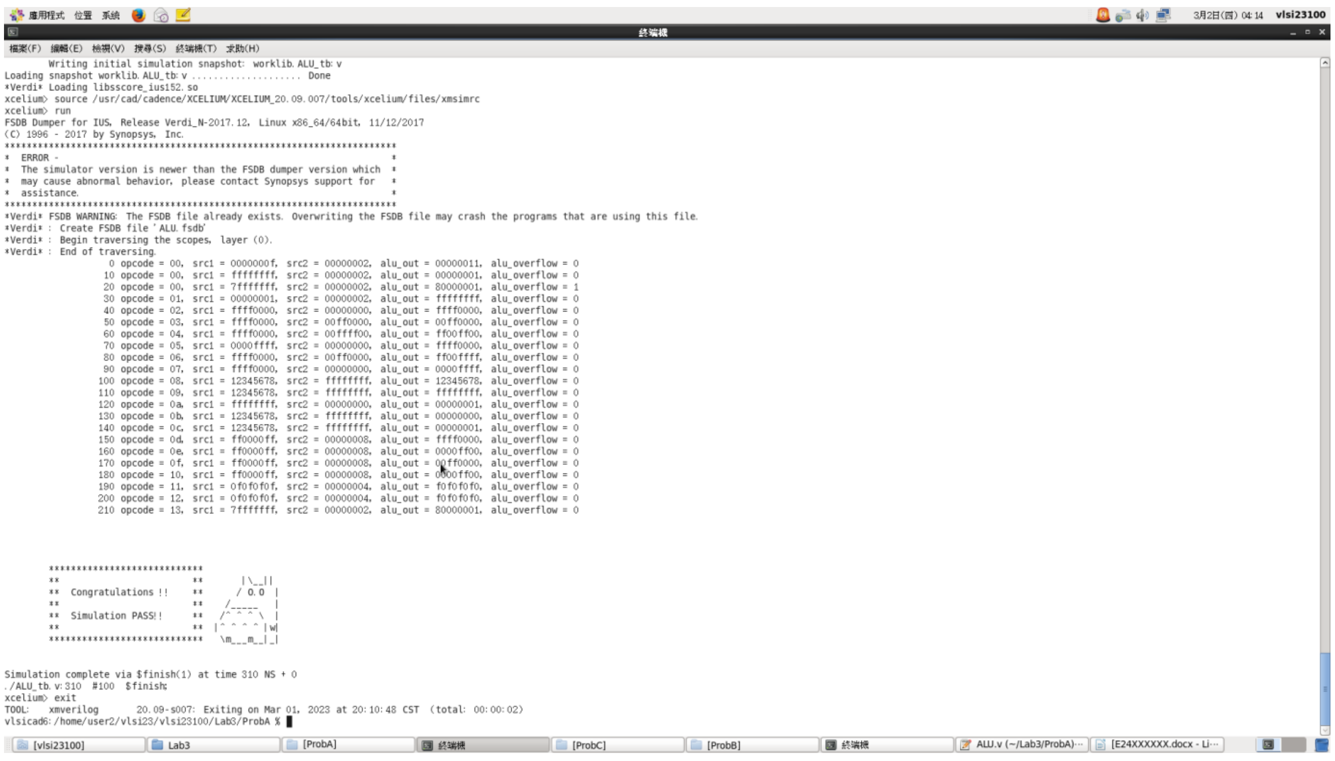
1. Based on the reference code, please implement the following operations.

alu_op	Operation	Description
00000	ADD	alu_out = src1 $_{\text{signed}}$ + src2 $_{\text{signed}}$
00001	SUB	alu_out = src1 $_{\text{signed}}$ - src2 $_{\text{signed}}$
00010	OR	alu_out = src1 | src2
00011	AND	alu_out = src1 & src2
00100	XOR	alu_out = src1 ^ src2
00101	NOT	alu_out = ~src1
00110	NAND	alu_out = ~(src1 & src2)
00111	NOR	alu_out = ~(src1 | src2)

alu_op	Operation	Description
01000	MAX	alu_out = max{$src1_{signed}$, $src2_{signed}$ }
01001	MIN	alu_out = min{$src1_{signed}$, $src2_{signed}$ }
01010	ABS	alu_out = $\vert src1_{signed} \vert$
01011	SLT	alu_out = ($src1_{signed} < src2_{signed}$) ? 32’d1 : 32’d0
01100	SLTU	alu_out = ($src1_{unsigned} < src2_{unsigned}$) ? 32’d1 : 32’d0
01101	SRA	alu_out = $src1_{signed}$ >>> $src2_{unsigned}$
01110	SLA	alu_out = $src1_{signed}$ <<< $src2_{unsigned}$
01111	SRL	alu_out = $src1_{unsigned}$ >> $src2_{unsigned}$
10000	SLL	alu_out = $src1_{unsigned}$ << $src2_{unsigned}$
10001	ROTR	alu_out = src1 rotate right by “src2 bits”
10010	ROTL	alu_out = src1 rotate left by “src2 bits”
10011	ADDU	alu_out = $src1_{unsigned} + src2_{unsigned}$

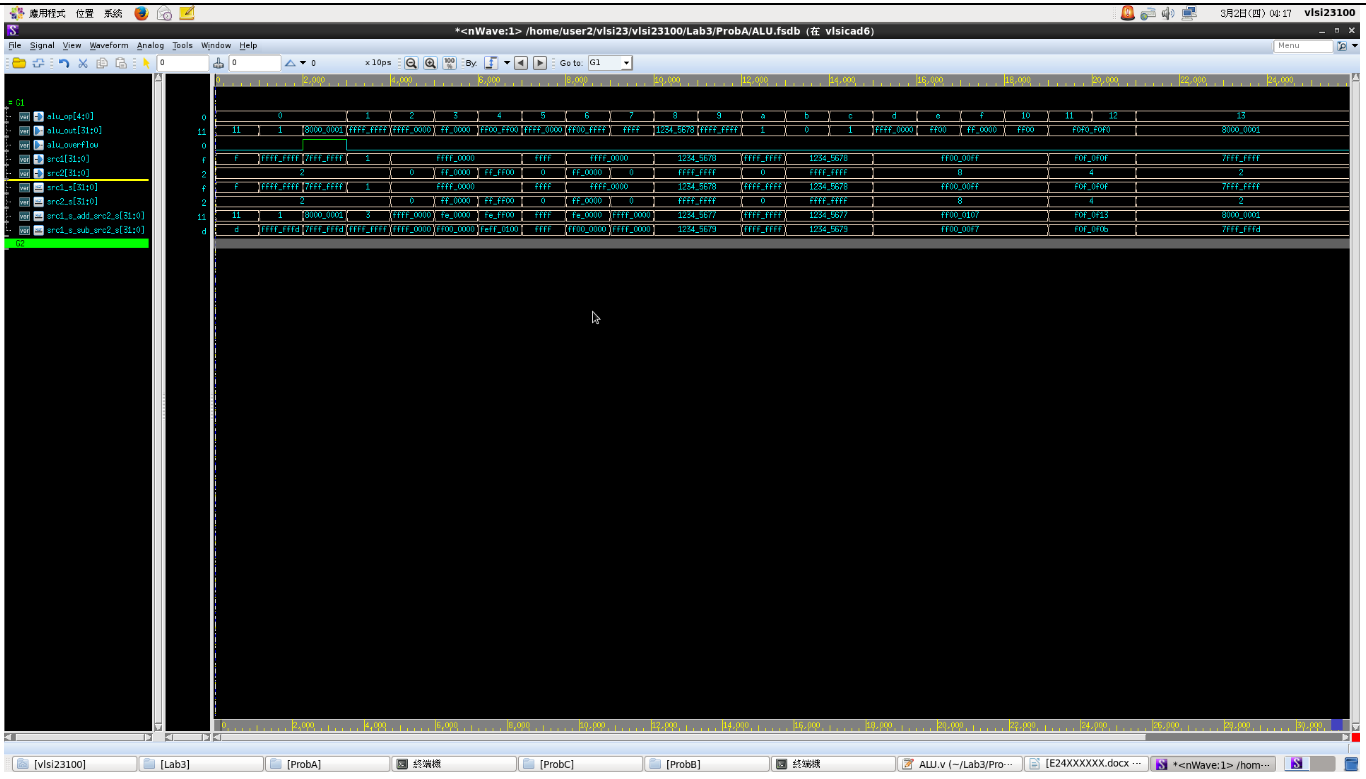
■ **Please attach your design waveforms.**

Your simulation result on the terminal.



```
        Writing initial simulation snapshot: worklib.ALU_tb:v
Loading snapshot worklib.ALU_tb:v .................... Done
*Verdi* Loading libsscore_ius152.so
xcelium> source /usr/cad/cadence/XCELIUM/XCELIUM_20.09.007/tools/xcelium/files/xmsimrc
xcelium> run
FSDB Dumper for IUS, Release Verdi_N-2017.12, Linux x86_64/64bit, 11/12/2017
(C) 1996 - 2017 by Synopsys, Inc.
*********************************************************************
*  ERROR -                                                          *
*  The simulator version is newer than the FSDB dumper version which *
*  may cause abnormal behavior, please contact Synopsys support for  *
*  assistance.                                                      *
*********************************************************************
*Verdi* FSDB WARNING: The FSDB file already exists. Overwriting the FSDB file may crash the programs that are using this file.
*Verdi* : Create FSDB file 'ALU.fsdb'
*Verdi* : Begin traversing the scopes, layer (0).
*Verdi* : End of traversing.
                  0 opcode = 00, src1 = 0000000f, src2 = 00000002, alu_out = 00000011, alu_overflow = 0
                 10 opcode = 00, src1 = ffffffff, src2 = 00000002, alu_out = 00000001, alu_overflow = 0
                 20 opcode = 00, src1 = 7fffffff, src2 = 00000002, alu_out = 80000001, alu_overflow = 1
                 30 opcode = 01, src1 = 00000001, src2 = 00000002, alu_out = ffffffff, alu_overflow = 0
                 40 opcode = 02, src1 = ffff0000, src2 = 00000000, alu_out = ffff0000, alu_overflow = 0
                 50 opcode = 03, src1 = ffff0000, src2 = 00ff0000, alu_out = 00ff0000, alu_overflow = 0
                 60 opcode = 04, src1 = ffff0000, src2 = 00ffff00, alu_out = ff00ff00, alu_overflow = 0
                 70 opcode = 05, src1 = 0000ffff, src2 = 00000000, alu_out = ffff0000, alu_overflow = 0
                 80 opcode = 06, src1 = ffff0000, src2 = 00ff0000, alu_out = ff00ffff, alu_overflow = 0
                 90 opcode = 07, src1 = ffff0000, src2 = 00000000, alu_out = 0000ffff, alu_overflow = 0
                100 opcode = 08, src1 = 12345678, src2 = ffffffff, alu_out = 12345678, alu_overflow = 0
                110 opcode = 09, src1 = 12345678, src2 = ffffffff, alu_out = ffffffff, alu_overflow = 0
                120 opcode = 0a, src1 = ffffffff, src2 = 00000000, alu_out = 00000001, alu_overflow = 0
                130 opcode = 0b, src1 = 12345678, src2 = ffffffff, alu_out = 00000000, alu_overflow = 0
                140 opcode = 0c, src1 = 12345678, src2 = ffffffff, alu_out = 00000001, alu_overflow = 0
                150 opcode = 0d, src1 = ff0000ff, src2 = 00000008, alu_out = ffff0000, alu_overflow = 0
                160 opcode = 0e, src1 = ff0000ff, src2 = 00000008, alu_out = 0000ff00, alu_overflow = 0
                170 opcode = 0f, src1 = ff0000ff, src2 = 00000008, alu_out = 00ff0000, alu_overflow = 0
                180 opcode = 10, src1 = ff0000ff, src2 = 00000008, alu_out = 0000ff00, alu_overflow = 0
                190 opcode = 11, src1 = 0f0f0f0f, src2 = 00000004, alu_out = f0f0f0f0, alu_overflow = 0
                200 opcode = 12, src1 = 0f0f0f0f, src2 = 00000004, alu_out = f0f0f0f0, alu_overflow = 0
                210 opcode = 13, src1 = 7fffffff, src2 = 00000002, alu_out = 80000001, alu_overflow = 0



        ****************************
        **                        **      |\__||
        **  Congratulations !!    **     / O.O  |
        **                        **   /_____   |
        **  Simulation PASS!!     **  /^ ^ ^ \  |
        **                        ** |^ ^ ^ ^ |w|
        ****************************  \m___m__|_|


Simulation complete via $finish(1) at time 310 NS + 0
./ALU_tb.v:310  #100  $finish;
xcelium> exit
TOOL:   xmverilog       20.09-s007: Exiting on Mar 01, 2023 at 20:10:48 CST  (total: 00:00:02)
vlsicad6:/home/user2/vlsi23/vlsi23100/Lab3/ProbA % 
```

Your waveform :



Input:alu_op,src1,src2

Output:alu_overflow,alu_out

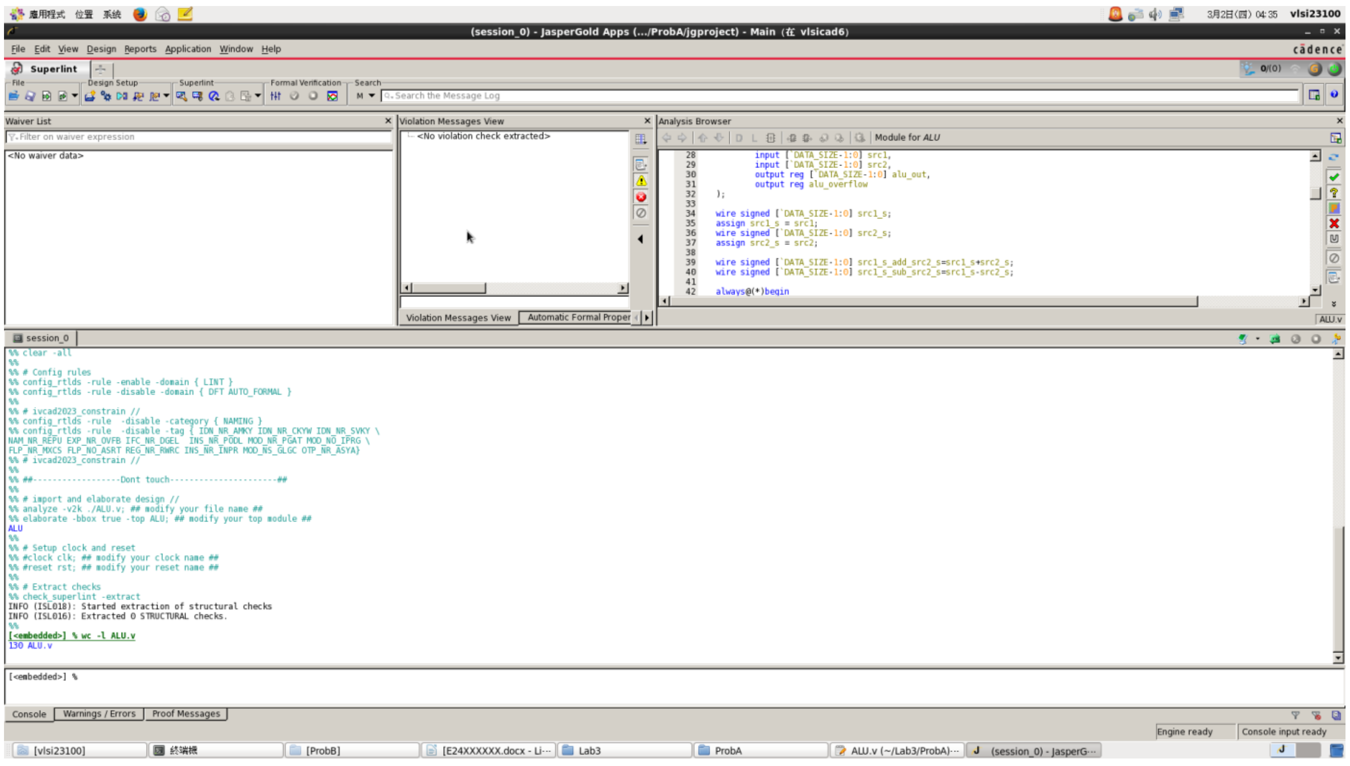
src1_s 為 src1 轉成 signed 的 wire; src2_s 為 src2 轉成 signed 的 wire，當 src1 和 src2 訊號進來時 src1_s

和 src2_s 會馬上把值轉成 signed 的形式。

src1_s_add_src2_s 為 src1_s 加 src2_s 的結果，也就是 src1 的 signed 形式加 src2 的 signed 形式; src1_s_sub_src2_s 為 src1_s 減 src2_s 的結果，也就是 src1 的 signed 形式減 src2 的 signed 形式。

當 alu_op 的值進來時，case 判別要執行怎麼樣的運算，之後再利用 src1,src2, src1_s, src2_s 進行運算，alu_out 輸出結果，alu_overflow 則是在 alu_op=5'b00000(ADD), alu_op=5'b00001(SUB), alu_op=5'b10011(ADDU)時利用 src1,src2, src1_s, src2_s, src1_s_add_src2_s, src1_s_sub_src2_s 進行判別，最後輸出結果。

SuperLint Coverage



```
            input [`DATA_SIZE-1:0] src1,
            input [`DATA_SIZE-1:0] src2,
            output reg [`DATA_SIZE-1:0] alu_out,
            output reg alu_overflow
    );

    wire signed [`DATA_SIZE-1:0] src1_s;
    assign src1_s = src1;
    wire signed [`DATA_SIZE-1:0] src2_s;
    assign src2_s = src2;

    wire signed [`DATA_SIZE-1:0] src1_s_add_src2_s=src1_s+src2_s;
    wire signed [`DATA_SIZE-1:0] src1_s_sub_src2_s=src1_s-src2_s;

    always@(*)begin
```

```
%% clear -all
%%
%% # Config rules
%% config_rtlds -rule -enable -domain { LINT }
%% config_rtlds -rule -disable -domain { DFT AUTO_FORMAL }
%%
%% # ivcad2023_constrain //
%% config_rtlds -rule  -disable -category { NAMING }
%% config_rtlds -rule  -disable -tag { IDN_NR_AMKY IDN_NR_CKYW IDN_NR_SVKY \
NAM_NR_REPU EXP_NR_OVFB IFC_NR_DGEL  INS_NR_PODL MOD_NR_PGAT MOD_NO_IPRG \
FLP_NR_MXCS FLP_NO_ASRT REG_NR_RWRC INS_NR_INPR MOD_NS_GLGC OTP_NR_ASYA}
%% # ivcad2023_constrain //
%%
%% ##-----------------Dont touch---------------------##
%%
%% # import and elaborate design //
%% analyze -v2k ./ALU.v; ## modify your file name ##
%% elaborate -bbox true -top ALU; ## modify your top module ##
ALU
%%
%% # Setup clock and reset
%% #clock clk; ## modify your clock name ##
%% #reset rst; ## modify your reset name ##
%%
%% # Extract checks
%% check_superlint -extract
INFO (ISL018): Started extraction of structural checks
INFO (ISL016): Extracted 0 STRUCTURAL checks.
%%
[<embedded>] % wc -l ALU.v
130 ALU.v
```

```
[<embedded>] %
```

Coverage:100%

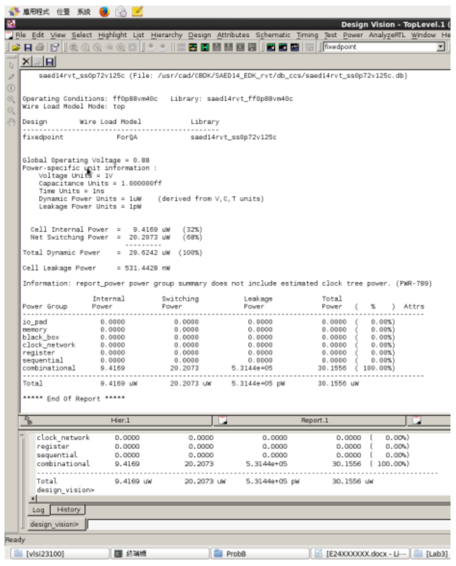
Prob B: Practice fixed point

Design your Verilog code with the following specifications: Number format: unsigned numbers.

a. The frame code and testbench are given. Follow the frame code to finish this homework. The decimal part should be rounded.

b. Follow the PPT file to synthesize your code.

After you synthesize your design, you may have some information about the circuit. Fill in the following form.

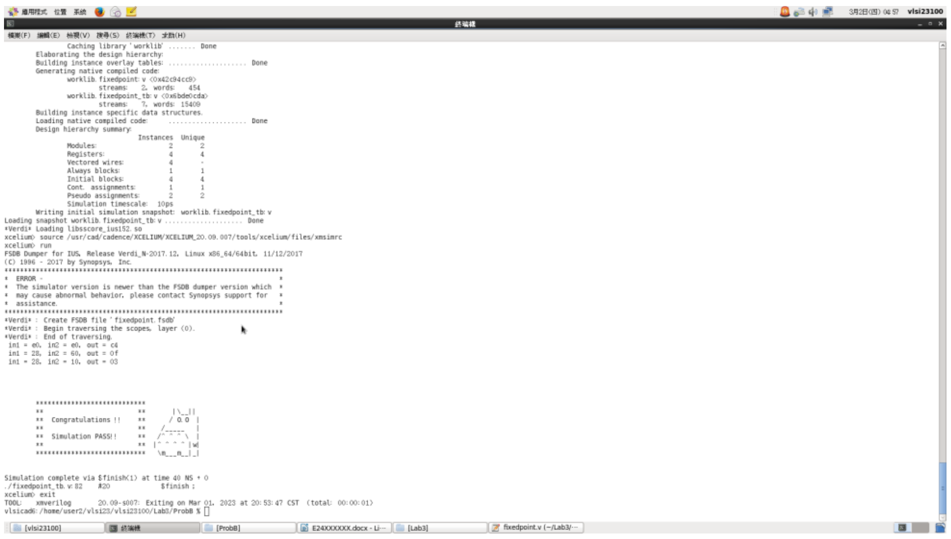
Timing (slack)	**Area (total cell area)**	**Power (total)**
19.16	**79.031998**	

```
    saed14rvt_ss0p72v125c (File: /usr/cad/CBDK/SAED14_EDK_rvt/db_ccs/saed14rvt_ss0p72v125c.db)


Operating Conditions: ff0p88vm40c   Library: saed14rvt_ff0p88vm40c
Wire Load Model Mode: top

Design        Wire Load Model            Library
------------------------------------------------
fixedpoint             ForQA             saed14rvt_ss0p72v125c


Global Operating Voltage = 0.88
Power-specific unit information :
    Voltage Units = 1V
    Capacitance Units = 1.000000ff
    Time Units = 1ns
    Dynamic Power Units = 1uW    (derived from V,C,T units)
    Leakage Power Units = 1pW


  Cell Internal Power  =   9.4169 uW   (32%)
  Net Switching Power  =  20.2073 uW   (68%)
                         ---------
Total Dynamic Power    =  29.6242 uW  (100%)

Cell Leakage Power     = 531.4428 nW

Information: report_power power group summary does not include estimated clock tree power. (PWR-789)

                 Internal         Switching           Leakage            Total
Power Group      Power            Power               Power              Power   (   %    )  Attrs
--------------------------------------------------------------------------------------------------
io_pad             0.0000           0.0000            0.0000             0.0000  (   0.00%)
memory             0.0000           0.0000            0.0000             0.0000  (   0.00%)
black_box          0.0000           0.0000            0.0000             0.0000  (   0.00%)
clock_network      0.0000           0.0000            0.0000             0.0000  (   0.00%)
register           0.0000           0.0000            0.0000             0.0000  (   0.00%)
sequential         0.0000           0.0000            0.0000             0.0000  (   0.00%)
combinational      9.4169          20.2073        5.3144e+05            30.1556  ( 100.00%)
--------------------------------------------------------------------------------------------------
Total              9.4169 uW       20.2073 uW     5.3144e+05 pW         30.1556 uW

***** End Of Report *****

    clock_network      0.0000          0.0000          0.0000          0.0000  (   0.00%)
    register           0.0000          0.0000          0.0000          0.0000  (   0.00%)
    sequential         0.0000          0.0000          0.0000          0.0000  (   0.00%)
    combinational      9.4169         20.2073      5.3144e+05         30.1556  ( 100.00%)
    ------------------------------------------------------------------------------------
    Total              9.4169 uW      20.2073 uW   5.3144e+05 pW      30.1556 uW
    design_vision>
```

Please attach your design waveforms.

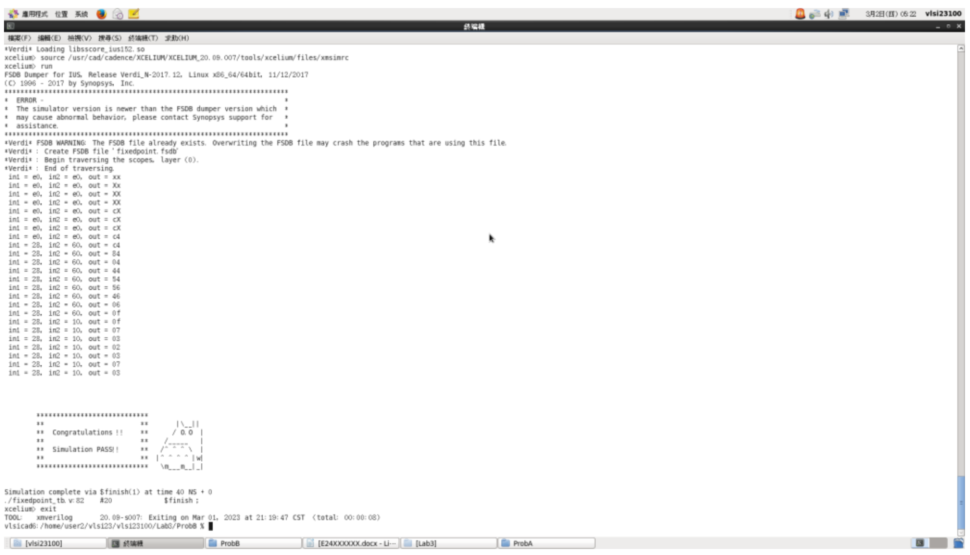
Your simulation result on the terminal.

RTL:



```
                Caching library 'worklib' ....... Done
        Elaborating the design hierarchy:
        Building instance overlay tables: .................... Done
        Generating native compiled code:
                worklib.fixedpoint:v <0x42c94cc9>
                        streams:   2, words:   454
                worklib.fixedpoint_tb:v <0x6bde0cda>
                        streams:   7, words: 15409
        Building instance specific data structures.
        Loading native compiled code:     .................... Done
        Design hierarchy summary:
                                 Instances  Unique
                Modules:                 2       2
                Registers:               4       4
                Vectored wires:          4       -
                Always blocks:           1       1
                Initial blocks:          4       4
                Cont. assignments:       1       1
                Pseudo assignments:      2       2
                Simulation timescale:  10ps
        Writing initial simulation snapshot: worklib.fixedpoint_tb:v
Loading snapshot worklib.fixedpoint_tb:v .................... Done
*Verdi* Loading libsscore_ius152.so
xcelium> source /usr/cad/cadence/XCELIUM/XCELIUM_20.09.007/tools/xcelium/files/xmsimrc
xcelium> run
FSDB Dumper for IUS, Release Verdi_N-2017.12, Linux x86_64/64bit, 11/12/2017
(C) 1996 - 2017 by Synopsys, Inc.
*********************************************************************
*  ERROR -                                                          *
*  The simulator version is newer than the FSDB dumper version which *
*  may cause abnormal behavior, please contact Synopsys support for  *
*  assistance.                                                      *
*********************************************************************
*Verdi* : Create FSDB file 'fixedpoint.fsdb'
*Verdi* : Begin traversing the scopes, layer (0).
*Verdi* : End of traversing.
 in1 = e0, in2 = e0, out = c4
 in1 = 28, in2 = 60, out = 0f
 in1 = 28, in2 = 10, out = 03




        ***************************
        **                       **      |\__||
        **  Congratulations !!   **     / O.O  |
        **                       **   /_____   |
        **  Simulation PASS!!    **  /^ ^ ^ \  |
        **                       ** |^ ^ ^ ^ |w|
        ***************************  \m___m__|_|


Simulation complete via $finish(1) at time 40 NS + 0
./fixedpoint_tb.v:82    #20            $finish;
xcelium> exit
TOOL:   xmverilog      20.09-s007: Exiting on Mar 01, 2023 at 20:53:47 CST  (total: 00:00:01)
vlsicad6:/home/user2/vlsi23/vlsi23100/Lab3/ProbB %
```

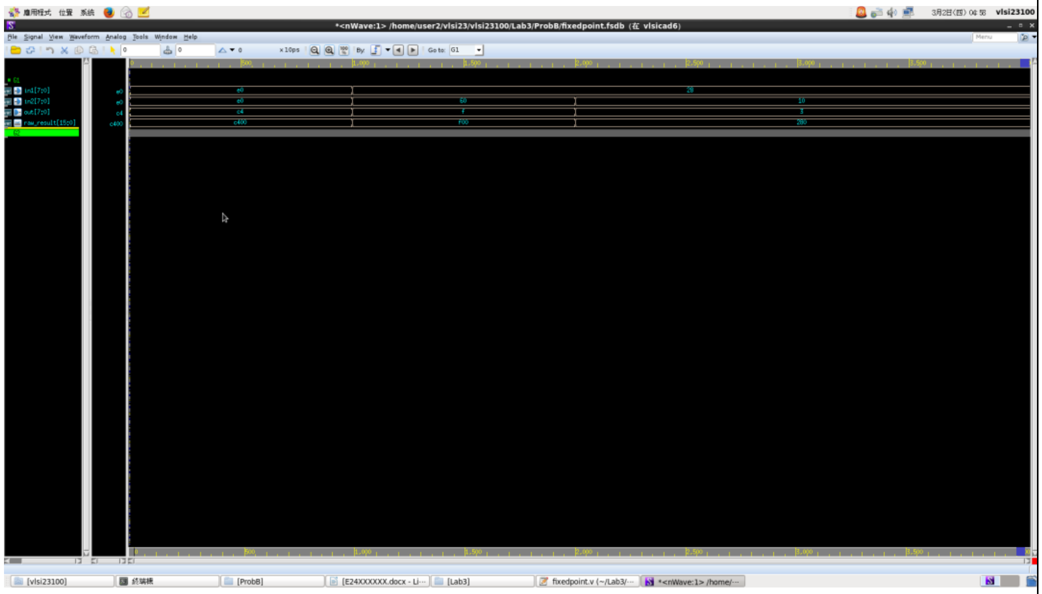
Synthesis:



```
*Verdi* Loading libsscore_ius152.so
xcelium> source /usr/cad/cadence/XCELIUM/XCELIUM_20.09.007/tools/xcelium/files/xmsimrc
xcelium> run
FSDB Dumper for IUS, Release Verdi_N-2017.12, Linux x86_64/64bit, 11/12/2017
(C) 1996 - 2017 by Synopsys, Inc.
*********************************************************************
*  ERROR -                                                          *
*  The simulator version is newer than the FSDB dumper version which *
*  may cause abnormal behavior, please contact Synopsys support for  *
*  assistance.                                                      *
*********************************************************************
*Verdi* FSDB WARNING: The FSDB file already exists. Overwriting the FSDB file may crash the programs that are using this file.
*Verdi* : Create FSDB file 'fixedpoint.fsdb'
*Verdi* : Begin traversing the scopes, layer (0).
*Verdi* : End of traversing.
 in1 = e0, in2 = e0, out = xx
 in1 = e0, in2 = e0, out = Xx
 in1 = e0, in2 = e0, out = XX
 in1 = e0, in2 = e0, out = XX
 in1 = e0, in2 = e0, out = cX
 in1 = e0, in2 = e0, out = cX
 in1 = e0, in2 = e0, out = cX
 in1 = e0, in2 = e0, out = c4
 in1 = 28, in2 = 60, out = c4
 in1 = 28, in2 = 60, out = 84
 in1 = 28, in2 = 60, out = 04
 in1 = 28, in2 = 60, out = 44
 in1 = 28, in2 = 60, out = 54
 in1 = 28, in2 = 60, out = 56
 in1 = 28, in2 = 60, out = 46
 in1 = 28, in2 = 60, out = 06
 in1 = 28, in2 = 60, out = 0f
 in1 = 28, in2 = 10, out = 0f
 in1 = 28, in2 = 10, out = 07
 in1 = 28, in2 = 10, out = 03
 in1 = 28, in2 = 10, out = 02
 in1 = 28, in2 = 10, out = 03
 in1 = 28, in2 = 10, out = 07
 in1 = 28, in2 = 10, out = 03



        ***************************
        **                       **      |\__||
        **  Congratulations !!   **     / 0.0  |
        **                       **   /_____   |
        **  Simulation PASS!!    **  /^ ^ ^ \  |
        **                       ** |^ ^ ^ ^ |w|
        ***************************  \m___m__|_|

Simulation complete via $finish(1) at time 40 NS + 0
./fixedpoint_tb.v:82     #20              $finish;
xcelium> exit
TOOL:    xmverilog       20.09-s007: Exiting on Mar 01, 2023 at 21:19:47 CST  (total: 00:00:08)
vlsicad6:/home/user2/vlsi23/vlsi23100/Lab3/ProbB % 
```

Your waveform (RTL & Synthesis) :

RTL:

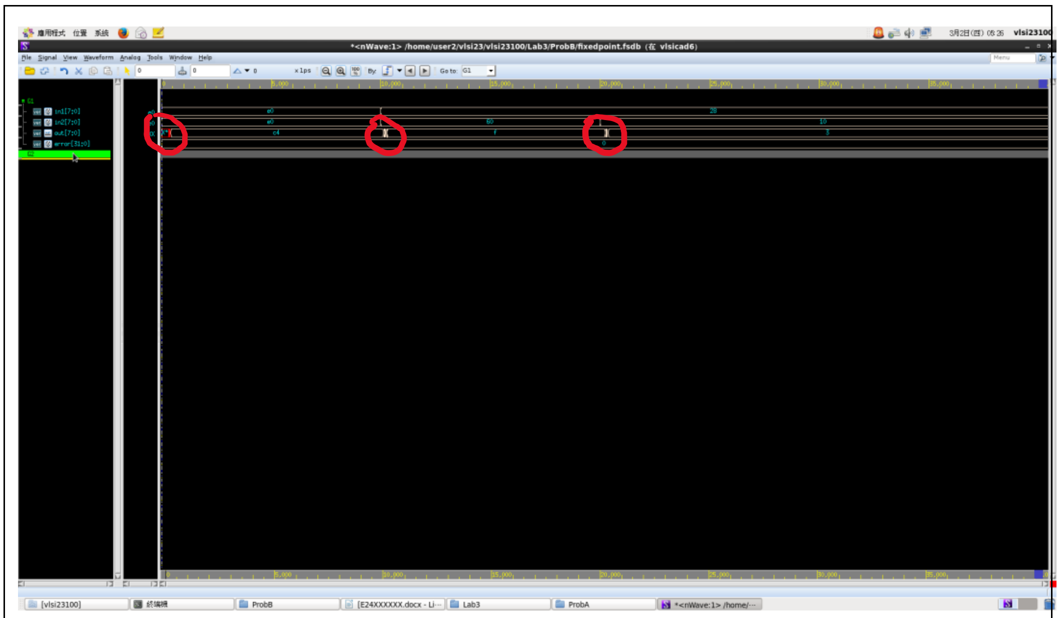

*<nWave:1> /home/user2/vlsi23/vlsi23100/Lab3/ProbB/fixedpoint.fsdb (在 vlsicad6)
in1[7:0]
in2[7:0]
out[7:0]
raw_result[15:0]
e0
28
60
10
c4
f
3
c400
f00
280

Input:in1,in2

Output:out

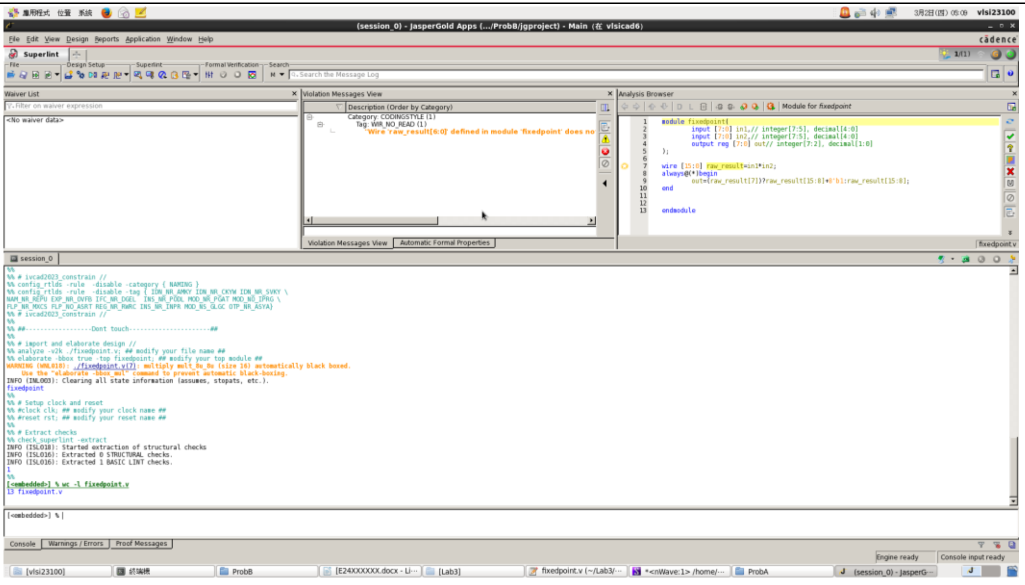
raw_result 為 in1*in2 的 wire，而 out 利用 raw_result[7]判別是否輸出的 raw_result[15:8]需加 1，最後輸出。

Synthesis:


*<nWave:1> /home/user2/vlsi23/vlsi23100/Lab3/ProbB/fixedpoint.fsdb (在 vlsicad6)
in1[7:0]
in2[7:0]
out[7:0]
error[31:0]

在合成過後模擬時會出現 out 在值變換時波形不穩定的情形，如圖中圈起來所示，且變換時機明顯落後於 input 值的變化，此原因為電路在 output 前需要時間運算，且在運算時 out 的值會變換且不穩定。

SuperLint Coverage



Description (Order by Category)
Category: CODINGSTYLE (1)
Tag: WIR_NO_READ (1)
'Wire 'raw_result[6:0]' defined in module 'fixedpoint' does no

```
module fixedpoint(
        input [7:0] in1,// integer[7:5], decimal[4:0]
        input [7:0] in2,// integer[7:5], decimal[4:0]
        output reg [7:0] out// integer[7:2], decimal[1:0]
);

wire [15:0] raw_result=in1*in2;
always@(*)begin
        out=(raw_result[7])?raw_result[15:8]+8'b1:raw_result[15:8];
end


endmodule
```

```
%%
%% # ivcad2023_constrain //
%% config_rtlds -rule  -disable -category { NAMING }
%% config_rtlds -rule  -disable -tag { IDN_NR_AMKY IDN_NR_CKYW IDN_NR_SVKY \
NAM_NR_REPU EXP_NR_OVFB IFC_NR_DGEL  INS_NR_PODL MOD_NR_PGAT MOD_NO_IPRG \
FLP_NR_MXCS FLP_NO_ASRT REG_NR_RWRC INS_NR_INPR MOD_NS_GLGC OTP_NR_ASYA}
%% # ivcad2023_constrain //
%%
%% ##-----------------Dont touch--------------------##
%%
%% # import and elaborate design //
%% analyze -v2k ./fixedpoint.v; ## modify your file name ##
%% elaborate -bbox true -top fixedpoint; ## modify your top module ##
WARNING (WNL018): ./fixedpoint.v(7): multiply mult_8u_8u (size 16) automatically black boxed.
    Use the "elaborate -bbox_mul" command to prevent automatic black-boxing.
INFO (INL003): Clearing all state information (assumes, stopats, etc.).
fixedpoint
%%
%% # Setup clock and reset
%% #clock clk; ## modify your clock name ##
%% #reset rst; ## modify your reset name ##
%%
%% # Extract checks
%% check_superlint -extract
INFO (ISL018): Started extraction of structural checks
INFO (ISL016): Extracted 0 STRUCTURAL checks.
INFO (ISL016): Extracted 1 BASIC LINT checks.
1
%%
[<embedded>] % wc -l fixedpoint.v
13 fixedpoint.v
```

```
[<embedded>] % |
```

Coverage:92.31%

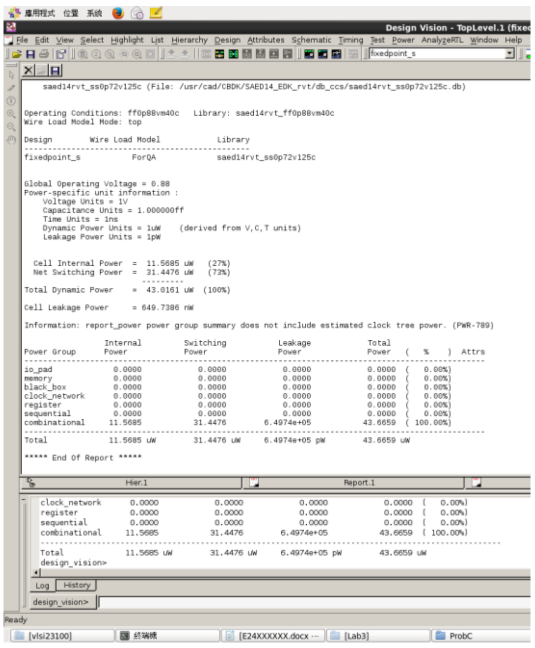
Prob C: Practice fixed point (signed)

Design your Verilog code with the following specifications: Number format: signed numbers.

a. The frame code and testbench are given. Follow the frame code to finish this homework. The decimal part should be rounded.

b. Follow the PPT file to synthesize your code.

After you synthesize your design, you may have some information about the circuit. Fill in the following form

Timing (slack)	**Area (total cell area)**	**Power (total)**
18.94	**98.701199**	

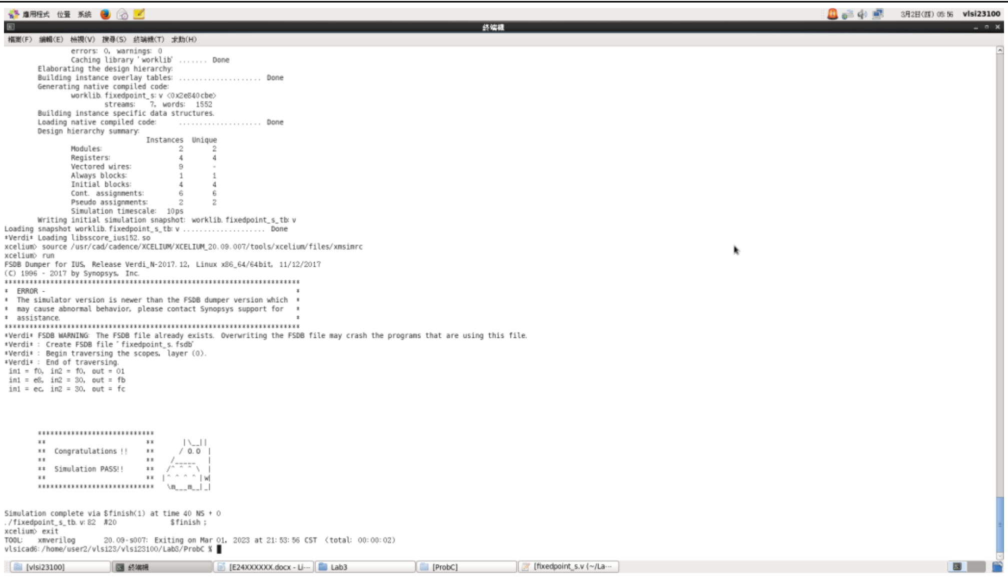
應用程式 位置 系統
Design Vision - TopLevel.1 (fixe
File Edit View Select Highlight List Hierarchy Design Attributes Schematic Timing Test Power AnalyzeRTL Window Help
fixedpoint_s
Hier.1
Report.1
Log
History
design_vision>
Ready
[vlsi23100]
終端機
[E24XXXXXX.docx ···
[Lab3]
ProbC

```
    saed14rvt_ss0p72v125c (File: /usr/cad/CBDK/SAED14_EDK_rvt/db_ccs/saed14rvt_ss0p72v125c.db)


Operating Conditions: ff0p88vm40c   Library: saed14rvt_ff0p88vm40c
Wire Load Model Mode: top

Design        Wire Load Model            Library
------------------------------------------------
fixedpoint_s           ForQA             saed14rvt_ss0p72v125c


Global Operating Voltage = 0.88
Power-specific unit information :
    Voltage Units = 1V
    Capacitance Units = 1.000000ff
    Time Units = 1ns
    Dynamic Power Units = 1uW    (derived from V,C,T units)
    Leakage Power Units = 1pW


  Cell Internal Power  =  11.5685 uW   (27%)
  Net Switching Power  =  31.4476 uW   (73%)
                         ---------
Total Dynamic Power    =  43.0161 uW  (100%)

Cell Leakage Power     = 649.7386 nW

Information: report_power power group summary does not include estimated clock tree power. (PWR-789)

                 Internal         Switching           Leakage            Total
Power Group      Power            Power               Power              Power   (   %    )  Attrs
--------------------------------------------------------------------------------------------------
io_pad             0.0000           0.0000           0.0000             0.0000  (   0.00%)
memory             0.0000           0.0000           0.0000             0.0000  (   0.00%)
black_box          0.0000           0.0000           0.0000             0.0000  (   0.00%)
clock_network      0.0000           0.0000           0.0000             0.0000  (   0.00%)
register           0.0000           0.0000           0.0000             0.0000  (   0.00%)
sequential         0.0000           0.0000           0.0000             0.0000  (   0.00%)
combinational     11.5685          31.4476       6.4974e+05            43.6659  ( 100.00%)
--------------------------------------------------------------------------------------------------
Total             11.5685 uW       31.4476 uW    6.4974e+05 pW         43.6659 uW

***** End Of Report *****
```

```
clock_network      0.0000           0.0000           0.0000             0.0000  (   0.00%)
register           0.0000           0.0000           0.0000             0.0000  (   0.00%)
sequential         0.0000           0.0000           0.0000             0.0000  (   0.00%)
combinational     11.5685          31.4476       6.4974e+05            43.6659  ( 100.00%)
--------------------------------------------------------------------------------------------------
Total             11.5685 uW       31.4476 uW    6.4974e+05 pW         43.6659 uW
design_vision>
```

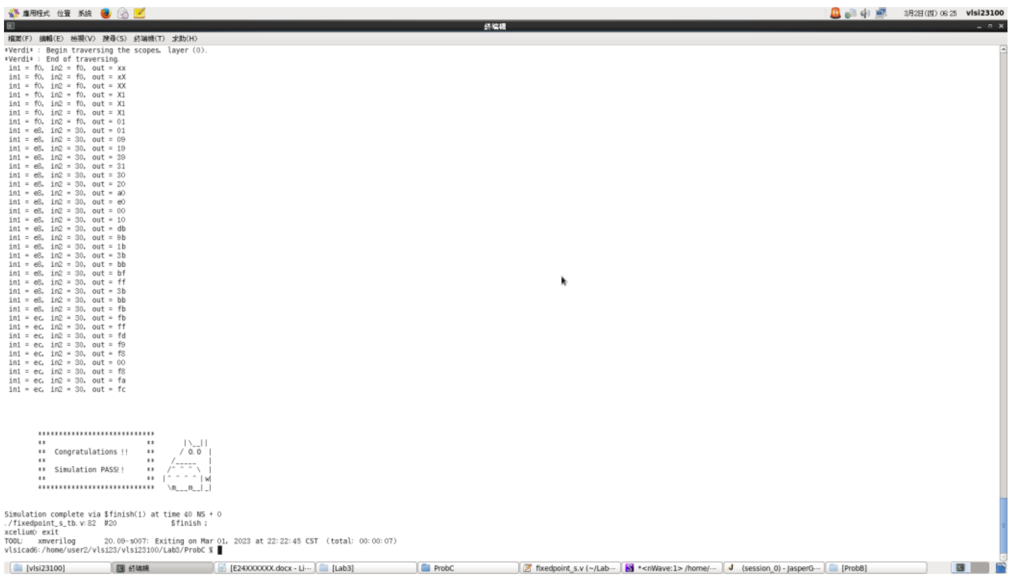
Please attach your design waveforms.

Your simulation result on the terminal.
RTL:



```
        errors: 0, warnings: 0
        Caching library 'worklib' ....... Done
    Elaborating the design hierarchy:
    Building instance overlay tables: .................... Done
    Generating native compiled code:
        worklib.fixedpoint_s:v <0x2e840cbe>
            streams:   7, words:  1552
    Building instance specific data structures.
    Loading native compiled code:     .................... Done
    Design hierarchy summary:
                          Instances  Unique
        Modules:                  2       2
        Registers:                4       4
        Vectored wires:           9       -
        Always blocks:            1       1
        Initial blocks:           4       4
        Cont. assignments:        6       6
        Pseudo assignments:       2       2
        Simulation timescale:  10ps
    Writing initial simulation snapshot:  worklib.fixedpoint_s_tb:v
Loading snapshot worklib.fixedpoint_s_tb:v .................... Done
*Verdi* Loading libsscore_ius152.so
xcelium> source /usr/cad/cadence/XCELIUM/XCELIUM_20.09.007/tools/xcelium/files/xmsimrc
xcelium> run
FSDB Dumper for IUS, Release Verdi_N-2017.12, Linux x86_64/64bit, 11/12/2017
(C) 1996 - 2017 by Synopsys, Inc.
*********************************************************************
*  ERROR -                                                          *
*  The simulator version is newer than the FSDB dumper version which *
*  may cause abnormal behavior, please contact Synopsys support for  *
*  assistance.                                                      *
*********************************************************************
*Verdi* FSDB WARNING: The FSDB file already exists. Overwriting the FSDB file may crash the programs that are using this file.
*Verdi* : Create FSDB file 'fixedpoint_s.fsdb'
*Verdi* : Begin traversing the scopes, layer (0).
*Verdi* : End of traversing.
 in1 = f0, in2 = f0, out = 01
 in1 = e8, in2 = 30, out = fb
 in1 = ec, in2 = 30, out = fc



    ***************************
    **                       **      |\__||
    **  Congratulations !!   **     / O.O  |
    **                       **   /_____   |
    **  Simulation PASS!!    **  /^ ^ ^ \  |
    **                       ** |^ ^ ^ ^ |w|
    ***************************  \m___m__|_|


Simulation complete via $finish(1) at time 40 NS + 0
./fixedpoint_s_tb.v:82   #20            $finish;
xcelium> exit
TOOL:   xmverilog       20.09-s007: Exiting on Mar 01, 2023 at 21:53:56 CST  (total: 00:00:02)
vlsicad6:/home/user2/vlsi23/vlsi23100/Lab3/ProbC %
```

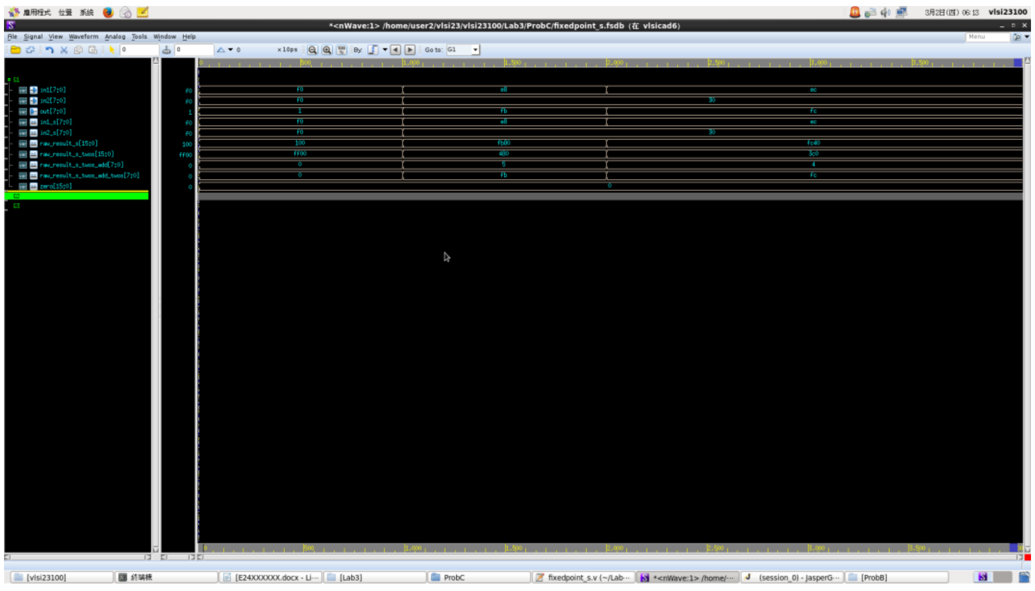
Synthesis:



```
*Verdi* : Begin traversing the scopes, layer (0).
*Verdi* : End of traversing.
 in1 = f0, in2 = f0, out = xx
 in1 = f0, in2 = f0, out = xX
 in1 = f0, in2 = f0, out = XX
 in1 = f0, in2 = f0, out = X1
 in1 = f0, in2 = f0, out = X1
 in1 = f0, in2 = f0, out = X1
 in1 = f0, in2 = f0, out = 01
 in1 = e8, in2 = 30, out = 01
 in1 = e8, in2 = 30, out = 09
 in1 = e8, in2 = 30, out = 19
 in1 = e8, in2 = 30, out = 39
 in1 = e8, in2 = 30, out = 31
 in1 = e8, in2 = 30, out = 30
 in1 = e8, in2 = 30, out = 20
 in1 = e8, in2 = 30, out = a0
 in1 = e8, in2 = 30, out = e0
 in1 = e8, in2 = 30, out = 00
 in1 = e8, in2 = 30, out = 10
 in1 = e8, in2 = 30, out = db
 in1 = e8, in2 = 30, out = 9b
 in1 = e8, in2 = 30, out = 1b
 in1 = e8, in2 = 30, out = 3b
 in1 = e8, in2 = 30, out = bb
 in1 = e8, in2 = 30, out = bf
 in1 = e8, in2 = 30, out = ff
 in1 = e8, in2 = 30, out = 3b
 in1 = e8, in2 = 30, out = bb
 in1 = e8, in2 = 30, out = fb
 in1 = ec, in2 = 30, out = fb
 in1 = ec, in2 = 30, out = ff
 in1 = ec, in2 = 30, out = fd
 in1 = ec, in2 = 30, out = f9
 in1 = ec, in2 = 30, out = f8
 in1 = ec, in2 = 30, out = 00
 in1 = ec, in2 = 30, out = f8
 in1 = ec, in2 = 30, out = fa
 in1 = ec, in2 = 30, out = fc



    ***************************
    **                       **      |\__||
    **  Congratulations !!   **     / O.O  |
    **                       **   /_____   |
    **  Simulation PASS!!    **  /^ ^ ^ \  |
    **                       ** |^ ^ ^ ^ |w|
    ***************************  \m___m__|_|


Simulation complete via $finish(1) at time 40 NS + 0
./fixedpoint_s_tb.v:82   #20            $finish;
xcelium> exit
TOOL:   xmverilog       20.09-s007: Exiting on Mar 01, 2023 at 22:22:45 CST  (total: 00:00:07)
vlsicad6:/home/user2/vlsi23/vlsi23100/Lab3/ProbC %
```

Your waveform (RTL & Synthesis) :

RTL:


*<nWave:1> /home/user2/vlsi23/vlsi23100/Lab3/ProbC/fixedpoint_s.fsdb (在 vlsicad6)
in1[7:0]
in2[7:0]
out[7:0]
in1_s[7:0]
in2_s[7:0]
raw_result_s[15:0]
raw_result_s_twos[15:0]
raw_result_s_twos_add[7:0]
raw_result_s_twos_add_twos[7:0]
zero[15:0]

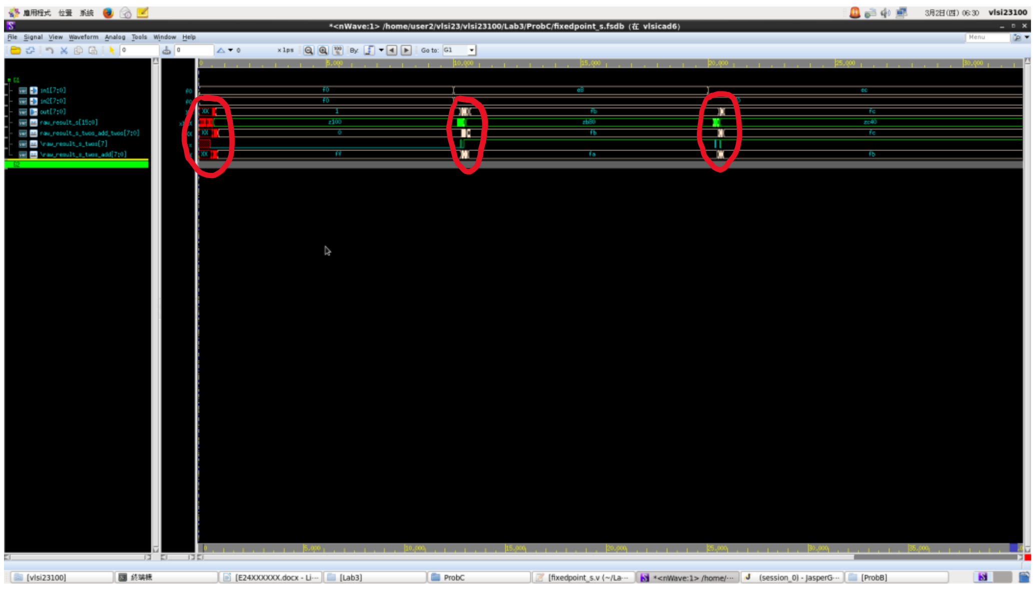
Input:in1,in2

Output:out

In1_s 為 in1 轉換成 signed 形式的 wire；In2_s 為 in2 轉換成 signed 形式的 wire；　raw_result_s 為 in1_s*in2_s 的 wire；raw_result_s_twos 為 (~raw_result_s)+1 的 wire；raw_result_s_twos_add 為 raw_result_s_twos[15:8]+1 的 wire；raw_result_s_twos_add_twos 為 (~raw_result_s_twos_add)+1；zero 為值為 0 的 wire。

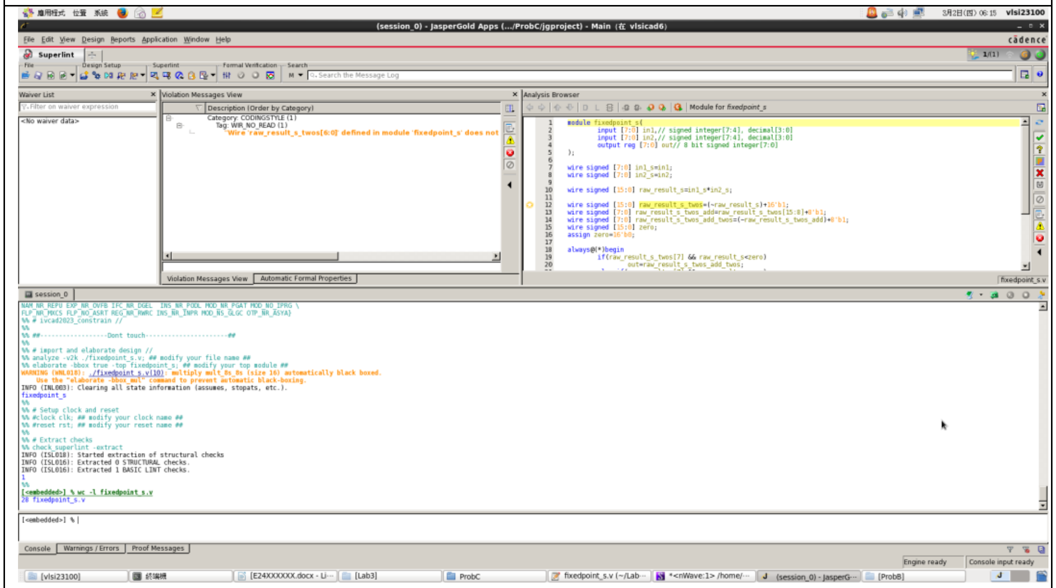
out 利用 raw_result_s_twos[7], raw_result_s[7], raw_result_s 和 zero 判別，若 raw_result_s 為負數且需進位，out 為 raw_result_s_twos_add_twos；若 raw_result_s 為正數且需進位，out 為 raw_result_s[15:8]+1；其餘的 out 則為 raw_result_s[15:8]。

Synthesis:


*<nWave:1> /home/user2/vlsi23/vlsi23100/Lab3/ProbC/fixedpoint_s.fsdb (在 vlsicad6)

在合成過後模擬時會出現圖中圈起來的部分在值變換時波形不穩定的情形，且變換時機明顯落後於 input 值的變化，此原因為電路在值到各個 wire 和 output 前需要時間運算，且在運算時各個 wire 和 output 的值會變換且不穩定。

SuperLint Coverage



(session_0) - JasperGold Apps (.../ProbC/jgproject) - Main (在 vlsicad6)

Violation Messages View

```
Category: CODINGSTYLE (1)
  Tag: WIR_NO_READ (1)
    "Wire 'raw_result_s_twos[6:0]' defined in module 'fixedpoint_s' does not
```

Analysis Browser — Module for fixedpoint_s

```
module fixedpoint_s(
        input [7:0] in1,// signed integer[7:4], decimal[3:0]
        input [7:0] in2,// signed integer[7:4], decimal[3:0]
        output reg [7:0] out// 8 bit signed integer[7:0]
);

wire signed [7:0] in1_s=in1;
wire signed [7:0] in2_s=in2;

wire signed [15:0] raw_result_s=in1_s*in2_s;

wire signed [15:0] raw_result_s_twos=(~raw_result_s)+16'b1;
wire signed [7:0] raw_result_s_twos_add=raw_result_s_twos[15:8]+8'b1;
wire signed [7:0] raw_result_s_twos_add_twos=(~raw_result_s_twos_add)+8'b1;
wire signed [15:0] zero;
assign zero=16'b0;

always@(*)begin
        if(raw_result_s_twos[7] && raw_result_s<zero)
                out=raw_result_s_twos_add_twos;
```

```
NAM_NR_REPU EXP_NR_OVFB IFC_NR_DGEL  INS_NR_PODL MOD_NR_PGAT MOD_NO_IPRG \
FLP_NR_MXCS FLP_NO_ASRT REG_NR_RWRC INS_NR_INPR MOD_NS_GLGC OTP_NR_ASYA}
%% # ivcad2023_constrain //
%%
%% ##-----------------Dont touch--------------------##
%%
%% # import and elaborate design //
%% analyze -v2k ./fixedpoint_s.v; ## modify your file name ##
%% elaborate -bbox true -top fixedpoint_s; ## modify your top module ##
WARNING (WNL018): ./fixedpoint_s.v(10): multiply mult_8s_8s (size 16) automatically black boxed.
    Use the "elaborate -bbox_mul" command to prevent automatic black-boxing.
INFO (INL003): Clearing all state information (assumes, stopats, etc.).
fixedpoint_s
%%
%% # Setup clock and reset
%% #clock clk; ## modify your clock name ##
%% #reset rst; ## modify your reset name ##
%%
%% # Extract checks
%% check_superlint -extract
INFO (ISL018): Started extraction of structural checks
INFO (ISL016): Extracted 0 STRUCTURAL checks.
INFO (ISL016): Extracted 1 BASIC LINT checks.
1
%%
[<embedded>] % wc -l fixedpoint_s.v
28 fixedpoint_s.v
```

```
[<embedded>] % |
```

Coverage:96.43%

At last, please write the lessons learned from this lab session, or some suggestions for this lab session. Thank you.

在 **Lab3** 學到如何利用 **behavior level** 寫出 **combinational** 的電路，也學到 **ALU** 和 **fixed point** 的概念與實現方法，還有如何將寫完的電路正確的合成。

Problem		Command
ProbA	Compile	% ncverilog ALU.v
	Simulate	% ncverilog ALU_tb.v +define+FSDB +access+r
ProbB	Compile	% ncverilog fixedpoint.v
	Simulate	% ncverilog fixedpoint_tb.v +define+FSDB +access+r
	Synthesis	% ncverilog fixedpoint_tb.v +define+FSDB+syn +access+r
ProbC	Compile	% ncverilog fixedpoint_s.v
	Simulate	% ncverilog fixedpoint_s_tb.v +define+FSDB +access+r
	Synthesis	% ncverilog fixedpoint_s_tb.v +define+FSDB+syn +access+r

Appendix A : Commands we will use to check your homework