

# National Cheng Kung University

## Department of Electrical Engineering

### *Introduction to VLSI CAD (Spring 2024)*

#### Lab Session 3

## Design of ALU and Multiplication Using Verilog Coding

Name	Student ID	
簡笠恩	E14102305	
Practical Sections:	Points	Marks
Prob A	30	
Prob B	30	
Prob C	20	
Report	15	
File hierarchy, naming...etc.	5	
Notes		

**Due Date: 15:00, March 13, 2024 @ 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

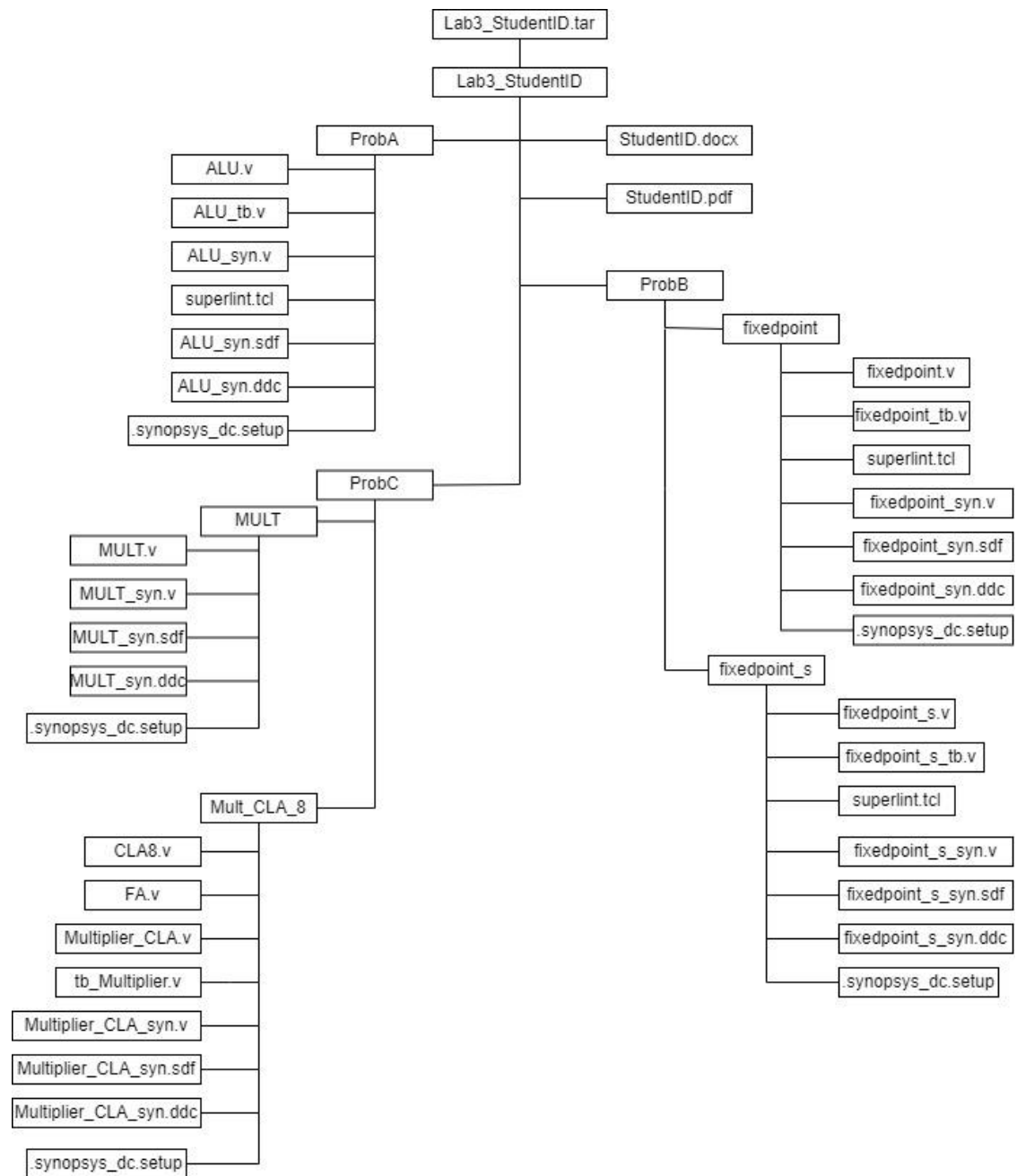


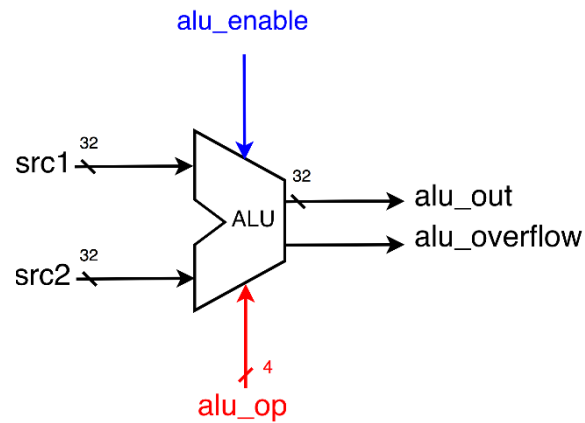
Fig.1 File hierarchy for Homework submission

---

*Prob A: Arithmetic Logic Unit*

---

**Design your Verilog code with the following specifications:**



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

alu_op	Operation	Description
00000	ADD	$\text{src1}_{\text{signed}} + \text{src2}_{\text{signed}}$
00001	SUB	$\text{src1}_{\text{signed}} - \text{src2}_{\text{signed}}$
00010	OR	<code>src1 or src2</code>
00011	AND	<code>src1 and src2</code>
00100	XOR	<code>src1 xor src2</code>
00101	NOT	Inversion of <code>src1</code>
00110	NAND	<code>src1 nand src2</code>
00111	NOR	<code>src1 nor src2</code>

alu_op	Operation	Description
01011	SLT	$\text{alu\_out} = (\text{src1}_{\text{signed}} < \text{src2}_{\text{signed}}) ? 32'd1 : 32'd0$
01100	SLTU	$\text{alu\_out} = (\text{src1}_{\text{unsigned}} < \text{src2}_{\text{unsigned}}) ? 32'd1 : 32'd0$
01101	SRA	$\text{alu\_out} = \text{src1}_{\text{signed}} \ggg \text{src2}_{\text{unsigned}}$
01110	SLA	$\text{alu\_out} = \text{src1}_{\text{signed}} \lll \text{src2}_{\text{unsigned}}$
01111	SRL	$\text{alu\_out} = \text{src1}_{\text{unsigned}} \gg \text{src2}_{\text{unsigned}}$
10000	SLL	$\text{alu\_out} = \text{src1}_{\text{unsigned}} \ll \text{src2}_{\text{unsigned}}$
10001	ROTR	$\text{alu\_out} = \text{src1}$ rotate right by "src2 bits"
10010	ROTL	$\text{alu\_out} = \text{src1}$ rotate left by "src2 bits"
10011	MUL	$\text{alu\_out} = \text{lower 32 bits of } (\text{src1} * \text{src2})$
10100	MULH	$\text{alu\_out} = \text{upper 32 bits of } (\text{src1}_{\text{signed}} * \text{src2}_{\text{signed}})$
10101	MULHSU	$\text{alu\_out} = \text{upper 32 bits of } (\text{src1}_{\text{signed}} * \text{src2}_{\text{unsigned}})$
10110	MULHU	$\text{alu\_out} = \text{upper 32 bits of } (\text{src1}_{\text{unsigned}} * \text{src2}_{\text{unsigned}})$

- The frame code and testbench are given. Follow the frame code to finish this homework. The decimal part should be rounded.
- 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)
18.04	4127.967	4.2602(mW)

**Please attach your design waveforms.**

Your simulation result on the terminal.

RTL

```

                210 opcode = 13, src1 = 12345678, src2 = f0f0f0f0, alu_op
223343, alu_overflow = 0

*****
**                                     **
**      Congratulations !!           **
**      Simulation PASS!!            **
**                                     **
*****

                | \_ | |
                / 0.0 |
                /_____ |
                / ^ ^ ^ \ |
                | ^ ^ ^ ^ |w|
                \m   m  |_ |

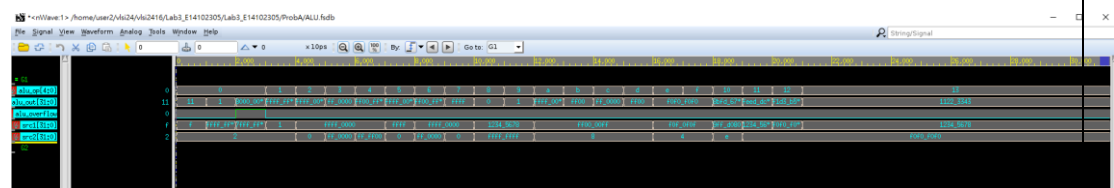
$finish called from file "ALU_tb.v", line 315.
$finish at simulation time          31000
      V C S   S i m u l a t i o n   R e p o r t
Time: 310000 ps
CPU Time:      0.530 seconds;      Data structure size:  0.0Mb
Mon Mar 18 18:25:32 2024
CPU time: .382 seconds to compile + .340 seconds to elab + .382 seconds to
+ .565 seconds in simulation
vlsicad6:/home/user2/vlsi24/vlsi2416/Lab3_E14102305/Lab3_E14102305/ProbA %

```

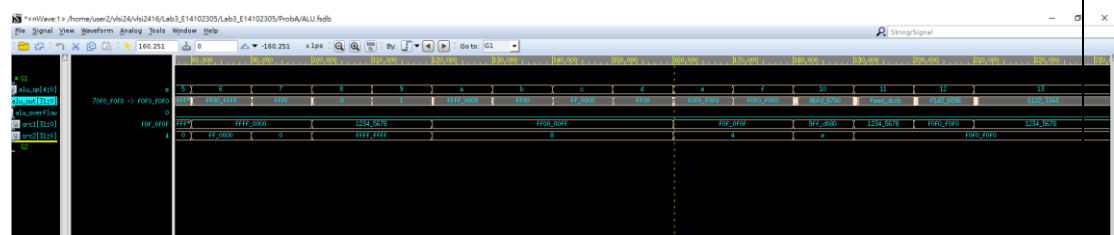
SYN

```
$finish called from file "ALU_tb.v", line 315.
$finish at simulation time      310000
      V C S   S i m u l a t i o n   R e p o r t
Time: 310000 ps
CPU Time:      1.100 seconds;      Data structure size:      3.3Mb
Mon Mar 18 18:49:28 2024
CPU time: 13.550 seconds to compile + 1.100 seconds to elab + .899 seconds to li
nk + 1.141 seconds in simulation
vlsicad6:/home/user2/vlsi24/vlsi2416/Lab3_El4102305/Lab3_El4102305/ProbA %
```

RTL



SYn



Page 7 of 15

The screenshot displays the Synopsys Design Compiler (DC) interface. The top window is titled 'Violation Messages View' and is split into two panes. The left pane, 'Waiver List', shows a filter on the waiver expression and a list of waivers. The right pane, 'Violation Messages View', shows a tree view of violations. The selected violation is 'In module/design-unit ALU, latch is assigned by blocking assignments'. Below the panes, the command line shows the configuration and execution of the 'elaborate' command, including the extraction of structural checks and the generation of the '135 ALU.v' file.

**Waiver List:**

- <No waiver data>

**Violation Messages View:**

- Description (Order by Category)
  - Category: SYNTHESIS (2)
    - Tag: LAT\_NR\_MXCB (1)
      - "The latches 'alu\_overflow' in the process/always block are mixed with combinational logic"
    - Tag: LAT\_NR\_BLAS (1)
      - "In module/design-unit ALU, latch is assigned by blocking assignments"
  - Category: CODINGSTYLE (3)
    - Tag: ASG\_MS\_READ (3)
      - "Unequal length operands in assignment in module/design-unit 'ALU'. Length of RHS is less than LHS"
      - "Unequal length operands in assignment in module/design-unit 'ALU'. Length of RHS is less than LHS"
      - "Unequal length operands in assignment in module/design-unit 'ALU'. Length of RHS is less than LHS"

**Command Line:**

```
% config rtlds -rule -enable -domain { LINT }
% config rtlds -rule -disable -domain { DFT AUTO FORMAL }
%
% # iucad2021 constrain //
% config rtlds -rule -disable -category { NAMING }
% config rtlds -rule -disable -tag { IDN NR APKY IDN NR OKYV IDN NR SVKY \
NAN NR REPU EXP NR OVFB JFC NR DGEL JNS NR POOL MOD NR PGAT MOD NO JPRG \
FLP NR MXCS FLP NO ASRT REG NR RMRC JNS NR INPR MOD NS GLGC OTP NR ASTA }
% # iucad2021 constrain //
%
% ##-----Dont touch-----##
%
% # import and elaborate design //
% analyze -v2k ./ALU.v; ## modify your file name ##
% elaborate -bbox true -top ALU; ## modify your top module ##
WARNING (MNL018): ./ALU.v(113): multiply mult_32s_32s (size 64) automatically black boxed.
Use the "elaborate -bbox mul" command to prevent automatic black-boxing.
WARNING (MNL018): ./ALU.v(117): multiply mult_32s_32s (size 64) automatically black boxed.
Use the "elaborate -bbox mul" command to prevent automatic black-boxing.
WARNING (MNL018): ./ALU.v(121): multiply mult_32s_32s (size 65) automatically black boxed.
Use the "elaborate -bbox mul" command to prevent automatic black-boxing.
WARNING (MNL018): ./ALU.v(126): multiply mult_32u_32u (size 64) automatically black boxed.
Use the "elaborate -bbox mul" command to prevent automatic black-boxing.
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 (ISL018): Extracted 0 STRUCTURAL checks.
INFO (ISL018): Extracted 3 SYNTH checks.
INFO (ISL018): Extracted 3 BASIC LINT checks.
6
%
% #  
% #wc -l ALU.v
135 ALU.v
```



## RTL

```

140.116.156.6 - PuTTY

in1 = 28, in2 = 10, out = 03

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

$finish called from file "fixedpoint_tb.v", line 82.
$finish at simulation time          4000
      V C S   S i m u l a t i o n   R e p o r t
Time: 40000 ps
CPU Time:      0.530 seconds;      Data structure size:   0.0Mb
Mon Mar 18 19:24:33 2024
CPU time: .352 seconds to compile + .342 seconds to elab + .384 seconds to link
+ .559 seconds in simulation
vlsicad6:/home/user2/vlsi24/vlsi2416/Lab3_E14102305/Lab3_E14102305/ProbB/fixedpo
int %

```

## SYN

```

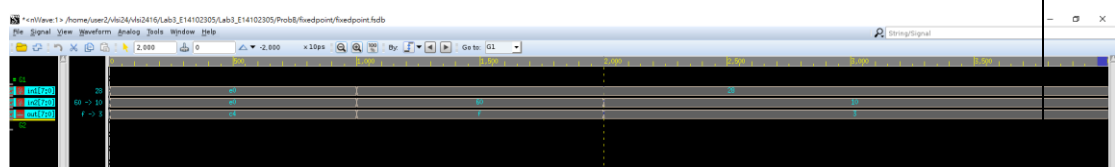
*****
**                                     I\__||
** Congratulations !!                / O.O |
**                                     /_____|
** Simulation PASS!!                / ^ ^ ^ \ |
**                                     | ^ ^ ^ ^ |w|
**                                     \m__m__|_|
*****

$finish called from file "fixedpoint_tb.v", line 82.
$finish at simulation time          40000
      V C S   S i m u l a t i o n   R e p o r t
Time: 40000 ps
CPU Time:      0.720 seconds;      Data structure size:   0.5Mb
Mon Mar 18 19:53:58 2024
CPU time: 11.072 seconds to compile + .537 seconds to elab + .745 seconds to lin
k + .754 seconds in simulation
vlsicad6:/home/user2/vlsi24/vlsi2416/Lab3_E14102305/Lab3_E14102305/ProbB/fixedpo
int %

```

Your waveform (RTL & Synthesis) :

## RTL



## SYN



---

*Prob B-2: Practice fixed point (signed)*

---

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

- The frame code and testbench are given. Follow the frame code to finish this homework. The decimal part should be rounded.
- 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)
19.5	77.915522	6.2067e-02 mW

**Please attach your design waveforms.**

Your simulation result on the terminal.

RTL

```
Verilog : End Of traversing.
0100
in1 = f0, in2 = f0, out = 01
fb80
fb80
in1 = e8, in2 = 30, out = fb
fc40
fc40
in1 = ec, in2 = 30, out = fc

*****
**                               |\_|
** Congratulations !!          / 0.0 |
**                               / ^ ^ ^ \ |
** Simulation PASS!!          / ^ ^ ^ ^ \ |
**                               | ^ ^ ^ ^ |w|
**                               \m__m__|_|
*****

$finish called from file "fixedpoint_s_tb.v", line 82.
$finish at simulation time      4000
V C S   S i m u l a t i o n   R e p o r t
Time: 40000 ps
CPU Time:      0.560 seconds;      Data structure size:  0.0Mb
Mon Mar 18 19:58:04 2024
CPU time: .358 seconds to compile + .372 seconds to elab + .417 seconds to link + .599 seconds in simulation
vlsicad6:/home/user2/vlsi24/vlsi2416/Lab3_E14102305/Lab3_E14102305/ProbB/fixedpoint_s %
```

SYN

```
*****
**                               |\_|
** Congratulations !!          / 0.0 |
**                               / ^ ^ ^ \ |
** Simulation PASS!!          / ^ ^ ^ ^ \ |
**                               | ^ ^ ^ ^ |w|
**                               \m__m__|_|
*****

$finish called from file "fixedpoint_s_tb.v", line 82.
$finish at simulation time      40000
V C S   S i m u l a t i o n   R e p o r t
Time: 40000 ps
CPU Time:      0.810 seconds;      Data structure size:  0.5Mb
Mon Mar 18 20:41:48 2024
CPU time: 11.734 seconds to compile + .683 seconds to elab + .787 seconds to link + .847 seconds in simulation
vlsicad6:/home/user2/vlsi24/vlsi2416/Lab3_E14102305/Lab3_E14102305/ProbB/fixedpoint_s %
```

Your waveform (RTL & Synthesis) :

The screenshot shows the Signal Studio interface. The top toolbar includes buttons for File, View, Waveform, Analog, Tests, Window, and Help. The main area displays a waveform with a time scale of 0 to 0.001 seconds. Below the waveform, a logic analyzer table is visible, showing data for three channels: P0, P1, and P2. The table has columns for Time, P0, P1, and P2. The data rows show the state of these channels at different time intervals.

Time	P0	P1	P2
0.000000	0	0	0
0.000000	0	0	0
0.000000	0	0	0
0.000000	0	0	0

[illegible]

**superint.tcl (session\_0) - JasperGold Apps (...fixedpoint.s/gproject) - Main**

File Edit View Design Reports Application Window Help

---

**SuperInt** [Icons]

File Design Setup SuperInt Formal Verification Search

M Q, Search the Message Log

---

**Waiver List**

Filter on waiver expression

No waiver data>

**Violation Messages View**

Description (Order by Category)
Category CODINGSTYLE (3)
Tag ASS_MIS_RAND (1)
Unequal length operands in assignment in module/design-unit "fixedpoint_s". Length of RHS is less than LHS. LHS "to"
Tag INP_NO_USED (2)
"The input/output port 'in1' defined in the module 'fixedpoint_s' is unused (neither read nor assigned)"
"The input/output port 'in2' defined in the module 'fixedpoint_s' is unused (neither read nor assigned)"

---

**session\_0**

```

\ # Config rules
\ config rtids --rule enable --domain { LINT }
\ config rtl08 --rule disable --domain { RTL AUTO FORMAL }
\
\ # lvsca2023 constrain //
\ config rtids --rule disable --category { WARNING }
\ config rtl08 --rule disable --tag { TON_HR_XNOV_TON_HR_CXOV_TON_HR_SXXV }
MAY_HR_RQPI EXP_HR_ZYPS IFB_HR_EGGL JNS_HR_FOOD_HRD_HR_PGAT_HRD_HD_IPRO
PIP_HR_RCKC_PIP_HR_AUST_RCE_HR_TMAC JSB_HR_THPR_HRD_HR_GAC_STP_HR_ASVA
\ # lvsca2023 constrain //
\ # ..... Dont touch..... ##
\
\ # import and elaborate design //
\ analyze -vdb /fixedpoint.s.v. ## modify your file name ##
\ elaborate -shw true -sw fixedpoint.s. ## modify your top module ##
WARNING (NWDI1): /fixedpoint.s.v(11): multiply multi_bits (size 16) automatically black boxed.
Use the 'elaborate_shw_mtl' command to prevent automatic black-boxing.
fixedpoint.s
[embedded]-> \ # Setup clock and reset
[embedded]-> \ clock clk. ## modify your clock name ##
[embedded]-> \ rstaut rst. ## modify your reset name ##
[embedded]-> \
[embedded]-> \ # Extract checks
[embedded]-> \ check superint-extract
INFO (ISA016): Started extraction of structural checks
INFO (ISA016): Extracted 6 STRUCTURAL checks.
INFO (ISA016): Extracted 3 BASIC LINT checks.
3
[embedded]-> \
[embedded]-> \ wc -l fixedpoint.s.v
$? fixedpoint.s.v
[embedded]-> \

```

---

*Prob C: Performance comparison*

---

**Synthesize the 8\*8-bit CLA multiplier implemented in Lab2 and the given 8\*8-bit multiplier separately.**

You should answer the following questions:

**1. Determine the lowest achievable clock period for both, along with the corresponding area and power consumption.**

	<b>Clock period</b>	<b>Timing (slack)</b>	<b>Area (total cell area)</b>	<b>Power (total)</b>
<b>CLA multiplier</b>	<b>.5</b>	<b>0</b>	<b>213.995525</b>	<b>0.1263 mW</b>
<b>"*"operator</b>	<b>0.17</b>	<b>0</b>	<b>151.735682</b>	<b>9.7272e-02 mW</b>

**2. Considering clock period and area, which structure has the better performance.**

在 **POWER** 的部分可以知道，下面那個會比較好，所以我們可以以此推論，**CLOCK PERIODE AND AREA** 越小越好

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

太美了，這麼棒的課，可以督促自己專研硬體描述語言，還可以順練耐心，者麼棒的課我會推給學弟妹們上的，沒有一個跑得掉。

Problem		Command
ProbA	Compile	% vcs -R ALU.v -full64
	Simulate	% vcs -R ALU_tb.v -debug_access+all -full64 +define+FSDB
	Synthesis	% vcs -R ALU_tb.v -debug_access+all -full64 +define+FSDB+syn
ProbB-1	Compile	% vcs -R fixedpoint.v -full64
	Simulate	% vcs -R fixedpoint_tb.v -debug_access+all -full64 +define+FSDB
	Synthesis	% vcs -R fixedpoint_tb.v -debug_access+all -full64 +define+FSDB+syn
ProbB-2	Compile	% vcs -R fixedpoint_s.v -full64
	Simulate	% vcs -R fixedpoint_s_tb.v -debug_access+all -full64 +define+FSDB
	Synthesis	% vcs -R fixedpoint_s_tb.v -debug_access+all -full64 +define+FSDB+syn

*Appendix A : Commands we will use to check your homework*