Instruction of test bench of one bit ALU

Input:

A, B : in std\_logic;

-- 'X00' for 1\_bit\_adder, '0X1' for and, '1X1' for or, 'X10' for stl/sltu

sel : in std\_logic\_vector(2 downto 0);

cin : in std\_logic;

inv : in std\_logic;

less : in std\_logic;

output

result : out std\_logic;

cout : out std\_logic;

temp : out std\_logic

Control

ADD: sel = “000”

SUB: sel = “100”

AND: sel = “0X1”

OR: sel = “1X1”

SLT/SLTU: sel = “X10”

Instruction of test bench of 32 bit ALU

Input:

signal ctrl\_t : std\_logic\_vector(5 downto 0);

signal A\_t : std\_logic\_vector(31 downto 0);

signal B\_t : std\_logic\_vector(31 downto 0);

output:

signal cout\_t : std\_logic; ---> carry out

signal oflow\_t : std\_logic; ---> overflow

signal ze\_t : std\_logic; -- ?1? -> is zero

signal R\_t : std\_logic\_vector(31 downto 0);

Control:

ADD: ctrl = "0XX000"

SUB: ctrl = "0XX001"

AND: ctrl = "0X0X1X"

OR: ctrl = "0X1X1X"

SLTU: ctrl = "00X001"

SLT: ctrl = "01X001"

SLL: ctrl = "1XXXXX"