CSE 331 - Spring 2021

Homework 3 Report

Coşkun Hasan ŞALTU
1801042631

1. Explanation of Moduls and Project

and4: This module gets two 4-bit input and give the answer of and operation of two input.

and32: This module gets two 32-bit input and give the answer of and operation of two input.This module use **and4** module 8 times.

or4: This module gets two 4-bit input and give the answer of or operation of two input.

or32: This module gets two 32-bit input and give the answer of or operation of two input.This module use **or4** module 8 times.

xor4: This module gets two 4-bit input and give the answer of xor operation of two input.

xor32: This module gets two 32-bit input and give the answer of xor operation of two input.This module
use x**or4** module 8 times.

nor4: This module gets two 4-bit input and give the answer of nor operation of two input.

nor32: This module gets two 32-bit input and give the answer of nor operation of two input.This module use n**or4** module 8 times.

not4: This module gets two 4-bit input and give the answer of not operation of two input.

not32: This module gets two 32-bit input and give the answer of not operation of two input.This module use n**ot4** module 8 times.

and32_4_numbers: This module gets 4 32-bit input and give the answer of and operation of four input.This module use **and32** module 3 times.

or32_8_numbers: This module gets 8 32-bit input and give the answer of or operation of eight input.This module use **or32** module 7 times.

full_adder: This module gets two 1-bit input and give the answer of add operation of two input.

_4bit_adder: This module gets two 4-bit input and give the answer of add operation of two input.This module use **full_adder** module 4 times.

_32bit_adder: This module gets two 32-bit input and give the answer of add operation of two input.This module use **_4bit_adder** module 8 times.

_32bit_subtracter: This module gets two 32-bit input and give the answer of substract operation of two input.This module gets negative of second number using **xor32** module and **_32bit_adder** module.After that,it add two numbers.

slt_32: This module gets two 32-bit input and give the answer of set less than operation of two input.This module subtract second input from first input using **_32bit_subtracter**.Than,it decide that's operation using most significant bit of substraction operation.

number_1_to_32: This module gets two 1-bit input and it gives 32-bit version of this number.

m81: This module gets eight 32-bit input and 3-bit select input.It chooses true input using select inputs.Firstly,this module find 32-bit version of select inputs using **number_1_to_32**.Secondly,it find not of this select inputs using **not_32.**Finally it decide answer using **and32_4_numbers** and **or32_8_numbers**.

m21: This module gets two 64-bit input and 1-bit select input.It chooses true input using select inputs.Firstly,this module find 64-bit version of select inputs using **number_1_to_32** two times.Secondly,it find not of this select inputs using **not_64** times**.**Finally it decide answer using **and32** and **or32** two times.

shift_right_64: This module gets one 64-bit input and it gives the number which 1-bit right shifted.

control: This module gets one 64-bit product and multiplicant.It just shift right product or it adds product left half and multiplicant after that shift product.It decide to use which operation using last bit of product.

datapath: This module gets two 32-bit multiplier and multiplicant.It call **control** module 32 times.

mult32: This module gets two 32-bit multiplier and multiplicant.It call **datapath** module.

alu32: This module is main module.It gets two 32-bit number,one carry_in,3-bit select and it gives one 32-bir output and carry_out.This module calculate ADD,XOR,SUB,MULT,SLT,NOR,AND,OR operation respectively.Finally,this decides true output using **m81** module.

2. Test Cases

This project tried with all select input and different 32-bit inputs

First test

```
VSIM 5> step -out -current
# time =  0, a =00000000000000000000000000001001, b=00000000000000000000000000000001, select = 000, out=00000000000000000000000000001010
# time = 20, a =00000000000000000000000000000101, b=00000000000000000000000000000100, select = 001, out=00000000000000000000000000000001
# time = 40, a =00000000000000000000000000001001, b=00000000000000000000000000000001, select = 010, out=00000000000000000000000000001000
# time = 60, a =00000000000000000000000000000111, b=00000000000000000000000000000101, select = 011, out=00000000000000000000000000100011
# time = 80, a =00000000000000000000000000001001, b=00000000000000000000000000000101, select = 100, out=00000000000000000000000000000000
# time = 100, a =00000000000000000000000000000101, b=00000000000000000000000000000100, select = 101, out=11111111111111111111111111111010
# time = 120, a =00000000000000000000000000001000, b=00000000000000000000000000001001, select = 110, out=00000000000000000000000000001000
# time = 140, a =00000000000000000000000000000101, b=00000000000000000000000000000001, select = 111, out=00000000000000000000000000000101
```

Second Test

```
VSIM 5> step -out -current
# time =  0, a =00000000000000000000000000001101, b=00000000000000000000000000001001, select = 000, out=00000000000000000000000000010110
# time = 20, a =00000000000000000000000000000101, b=00000000000000000000000000000100, select = 001, out=00000000000000000000000000000001
# time = 40, a =00000000000000000000000000001001, b=00000000000000000000000000001101, select = 010, out=11111111111111111111111111111100
# time = 60, a =00000000000000000000000000001111, b=00000000000000000000000000000101, select = 011, out=00000000000000000000000001001011
# time = 80, a =00000000000000000000000000001001, b=00000000000000000000000000000101, select = 100, out=00000000000000000000000000000000
# time = 100, a =00000000000000000000000000000101, b=00000000000000000000000000000101, select = 101, out=11111111111111111111111111111010
# time = 120, a =00000000000000000000000000001000, b=00000000000000000000000000001001, select = 110, out=00000000000000000000000000001000
# time = 140, a =00000000000000000000000000000101, b=00000000000000000000000000001001, select = 111, out=00000000000000000000000000001101
```