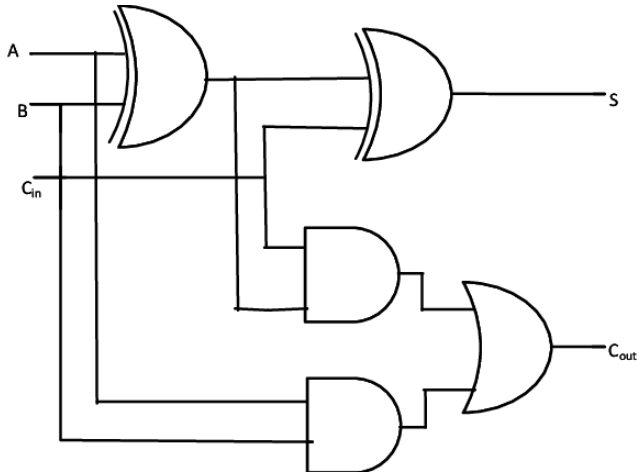
EECS 2070 02 Digital Design Labs 2020 Lab 1
學號：107062181　姓名：Calerb Louis Jean

1. 實作過程



1. Full Adder

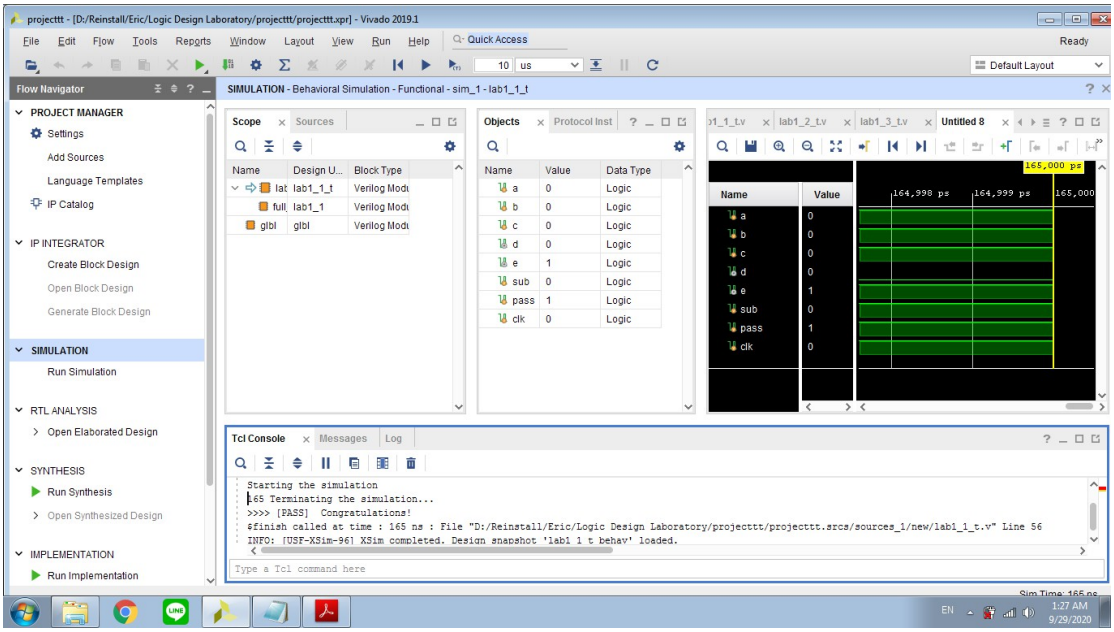
   In this block diagram, A=a, B=b, C_{in}=c, S=d, C_{out} = e.

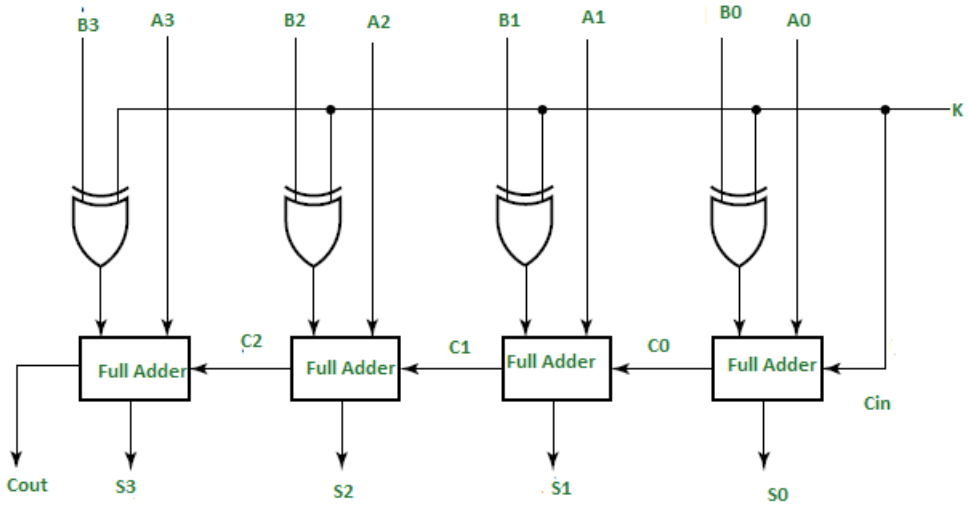
   From this homework full adder regulations, we have to use the sub as a signal to toggle the value of b.

   So, according to the circuit diagram, I used 1 xor gate and 2 and gates. By using the behavioral modeling, I, then use a switch case of the value of the sub to determine when to toggle the value of b.

   Once the value of d and e has been assigned correctly, the output and input value, is then connected to the test bench to check whether the implementation is correct by simulating the 2^n possible output.

   RESUL SNAPSHOT





2. 4-bit Adder using 4 1-bit full Adder

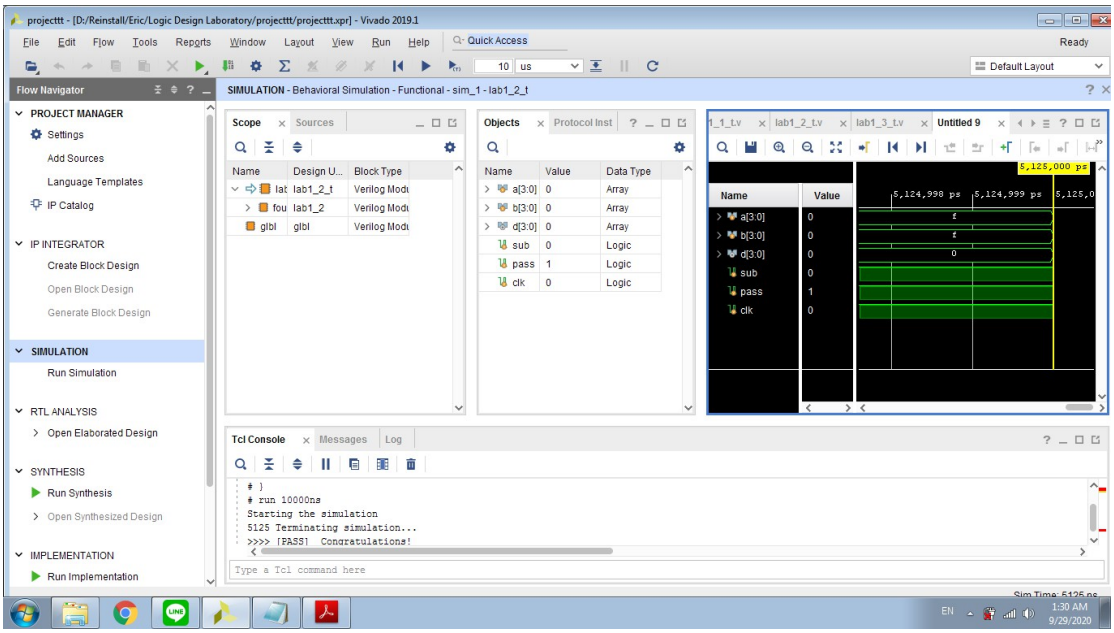
   In this block diagram $A_i = a_i$, $B_i = b_i$, K =Sub $S_i = d_i$,. My implementation process is simple, since I have previously implemented a 1-bit full adder, I just make an interconnection between my previous full adder and the 4 bit –adder by importing the filename of the previous full adder followed by the module name of each different adder. I also use a 4 xor gates which helps determine whether to toggle b or not.

```
lab1_1 full_adder1(.e(e[0]),.d(d[0]),.a(a[0]),.b(b[0]),.c(sub),.sub(sub));
lab1_1 full_adder2(.e(e[1]),.d(d[1]),.a(a[1]),.b(b[1]),.c(e[0]),.sub(sub));
lab1_1 full_adder3(.e(e[2]),.d(d[2]),.a(a[2]),.b(b[2]),.c(e[1]),.sub(sub));
```

lab1_1 full_adder4(.e(e[3]),.d(d[3]),.a(a[3]),.b(b[3]),.c(e[2]),.sub(sub));

in this snippet of code, each value is being passed to the full adder inside brackets. The reason that sub is passed to c, is because the c value implemented which is the C_{in} of the previous full adder is being used to toggle the b in this implementation and this same sub become the C_{out} of the 4-bit adder.

RESULT SNAPSHOT

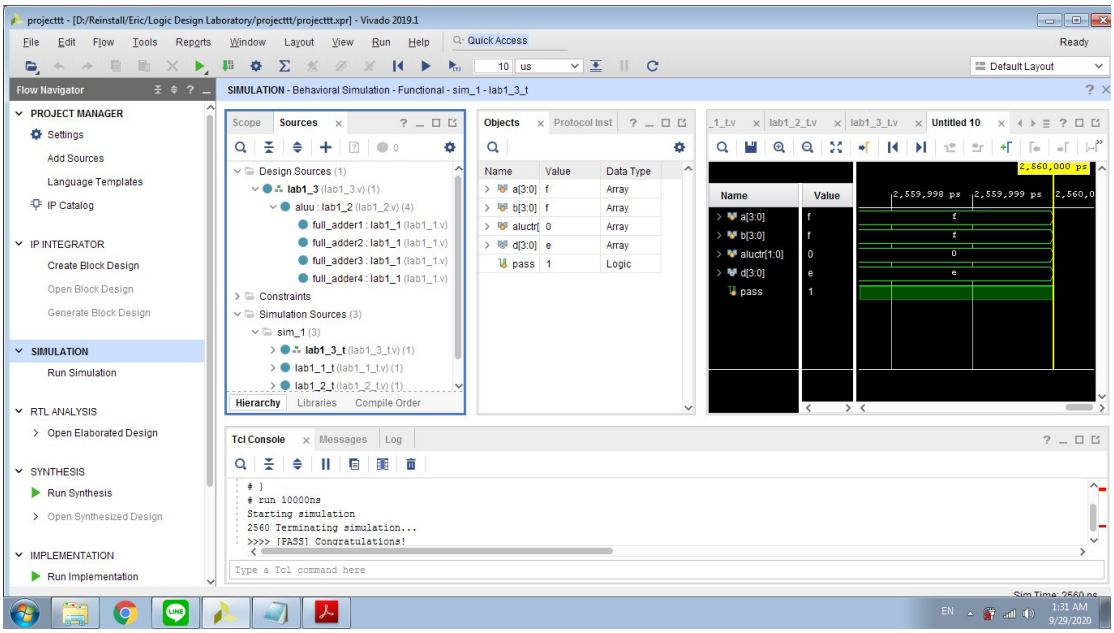


3. 4-bits ALU

I called the module of the 4-bits adder previously implemented in lab_1_2 by passing the corresponding value. I used a switch cased with the value of the aluctr to get the value of the sub and pass this value to the 4-bits adder because the sub is needed in the 4-bits adder to check whether to make addition on subtraction.

The value being return from the function call is afterward assigned to the output d according to the Aluctr conditions.

RESULT SNAPSHOT



1. Full Adder

   Skills of basic C programming language referring to the switch statement, using conditional operator in the if statement and function call of test and printerror.

2. 4-bit Adder using 4 1-bit full Adder

   Skills of basic C programming language referring to function call of test and printerror, use of flag.

3. 4-bits ALU

   Skills of basic C programming language referring to switch, function call, and ternary operation.

2. 學到的東西與遇到的困難

   1. Full Adder

      I was unable to use the variable declare with wire inside the always.

   2 4-bit Adder using 4 1-bit full Adder

      I learned how to import and use the full adder inside the 4-bit adder.

   2. 4-bits ALU

      I stuck with the simulation of lab_1_2 and lab_1_3, I learned how to navigate between design and simulation module by setting them as top.

3. 想對老師或助教說的話
   None!