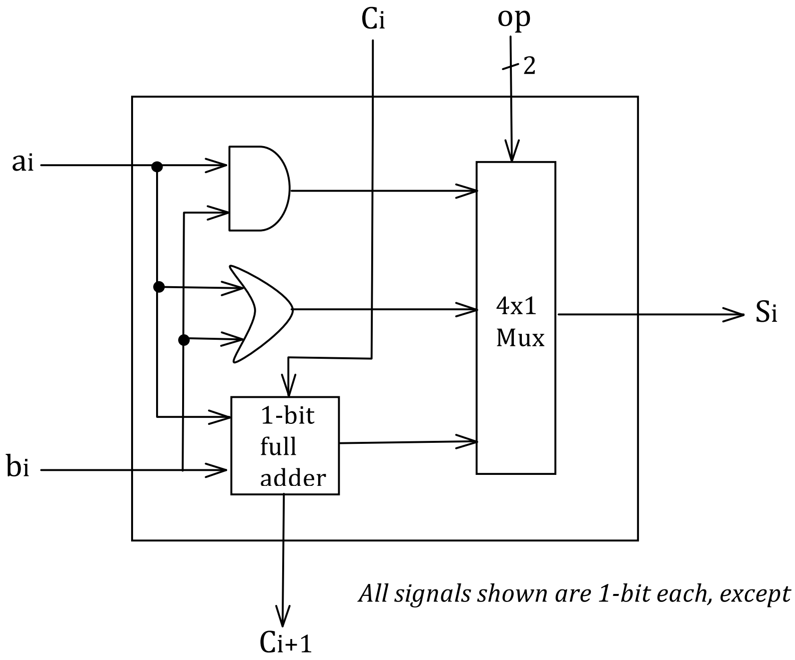
CSCI 112 (Spring 2015) Simulation Programming 1 25 pts Due: 04/17 (F)

16-bit ALU simulation, Phase1: 1-bit ALU simulation

As the first step of simulating a 16-bit ALU, which performs AND, OR, Add, Sub, and overflow checking operations, write a simulation program for 1-bit ALU in 80x86 assembly language.
In Phase1, include the following three functionalities in the 1-bit ALU:
AND, OR, and Add
Subtraction and overflow checking functionalities will be included in Phase2.

As explained in the class, the simulation granularity is the following 4 sub-components:
AND gate, OR gate, 1-bit full adder, and 4x1 multiplexor
Your 1-bit ALU simulator should have the above four sub-components as procedures (with parameters). You should also define a procedure for the 1-bit ALU (with parameters) and the main program (driver) calls it with passing argument values.

The 1-bit ALU is controlled by op signal, i.e.,
op (2bits): 00 – AND operation
01 – OR operation
10 – Add operation



All signals shown are 1-bit each, except op (2bits)

Suggested steps:
Step1. Define procedures for AND gate, OR gate, 1-bit-full adder, and 4x1 Mux;
Test each sub-component for the correctness of the logic, before implementing Step2.
Step2. Define a procedure for the 1-bit ALU module, which calls the sub-components.
Step3. Write the main driver code, which prompts and accesses input values; calls the 1-bit ALU procedure by passing input values to parameters; displays input/output values in an appropriate format.

Submission:

Include good documentations in the code (global doc. and each procedure head doc.) and
- submit hardcopies of the source code and run time output.
- Also send your source code by Email to jpark@csufresno.edu

Test run data is shown in the next page.

Test run data (input values) – please use these values for the output submission:

ai	bi	ci	op	
0	0	0	00	--- your output msg box should include these 4 input values and outputs (Si, Ci+1)
0	1	1	00	
1	0	0	00	
1	1	1	00	

0	0	1	01	
0	1	0	01	
1	0	1	01	
1	1	0	01	

0	0	0	10	
0	0	1	10	
0	1	0	10	
0	1	1	10	
1	0	0	10	
1	0	1	10	
1	1	0	10	
1	1	1	10	