CS221: Digital Design

Assignment 3 **Full Marks: 50**

Submission Deadline: 10th October 2020 in MS Teams

Instructions:

1. Submit a single zip file which contains softcopies of all files. The name of the file would be "RollNo_M3A1.zip". The TA will run your submission in Linux system to evaluate.

2. **Please give me your word that you will not use any unfair means. You can use Mano's book and my video lectures and course lecture slides to solve the assignment. You have all the options and had all the time and opportunities communicate answers to one another, or discuss with other, or copy the solution from Internet and from your friend. But, I trust you and your conscience to follow these guidelines. Hope you will repay my trust. Here is an amazing story about this experiment of trust: (https://journosdiary.com/2017/04/21/exam-without-invigilation-an-experiment-in-trust/).**

Question:

Consider the design of 4-bits Carry look ahead adder of signed two's complement numbers with overflow detector. The inputs to your design are two 4bits 2's complement numbers A and B and a control signal M. When M=0, perform A+B, when M=1, perform A-B. The output of your design a 4-bits output A +/- B and an overflow indicator V. V=0 means there is no overflow occurs during operation. V=1 indicates there is overflow during the operation A +/- B. Your design should be modular. Design a carry look ahead generator module (like Fig 4.11), a half adder module. Use these modules to develop 4bit carry look ahead adder (in Fig. 4.12) and then use that module to create your actual design.

Following diagrams are useful to develop your design. You will get all details in Lecture 15 and Lecture 16 and also in Mano's book Chapter 4. Note that Fig 4.11 is a 3bits carry look ahead generator whereas you have to develop a 4-bits one. In the diagram in Fig 4.13 below, you have to use the 4bit carry look ahead adder of Fig 4.12 in place of the Full adder chain.

Submission:

1. Write Verilog code for your modular design.
2. Write test bench Verilog code to verify if that your design is working. The test bench should contain corner cases of the possible inputs. Also, test bench should automatically check if the results are correct and report if the verification is successful/fails.

You can use Icarus Verilog or any other Verilog compiler available in Linux to compile and verify the design. You may also get online Verilog compiler as well. We will verify with Verilog compiler available with Linux.
Sample command to compile and simulate your design (adder_testbench.v is the testbench code and adder.v is the adder module).

$ iverilog adder_testbench.v adder.v -o runme
$ ./runme

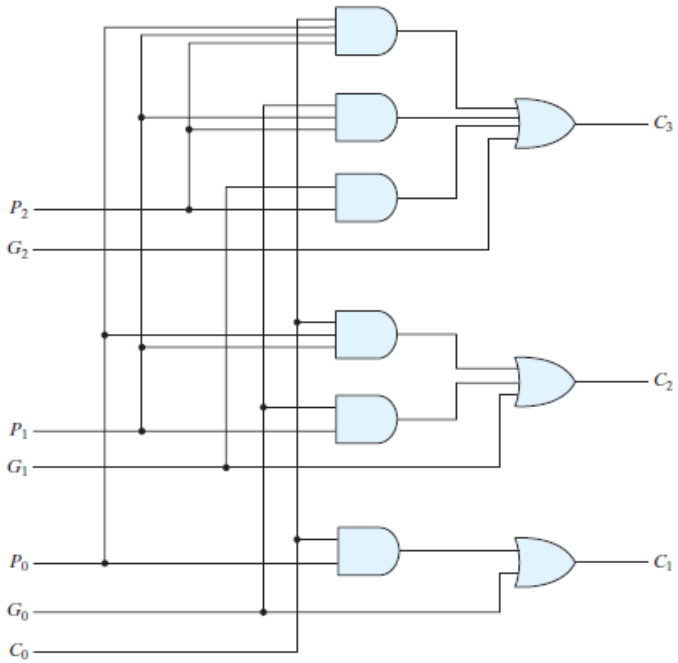

FIGURE 4.11
Logic diagram of carry lookahead generator

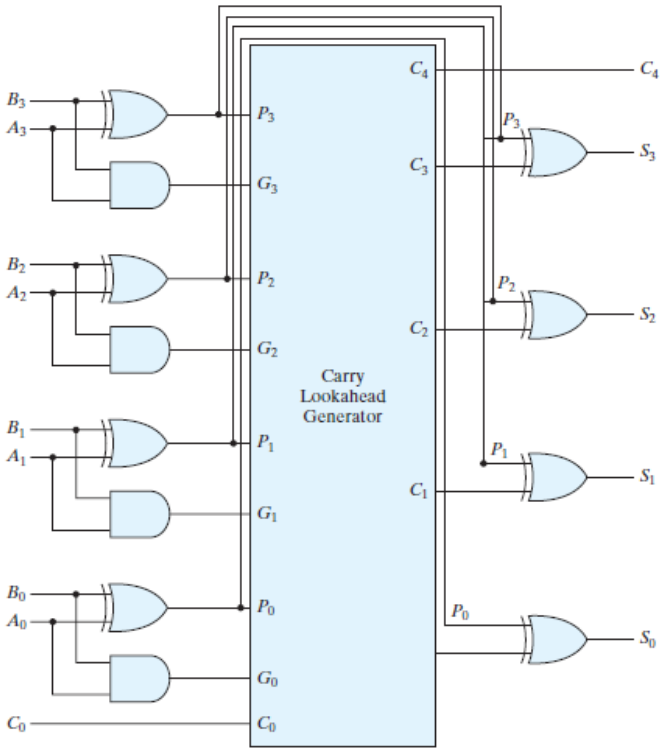

FIGURE 4.12
Four-bit adder with carry lookahead

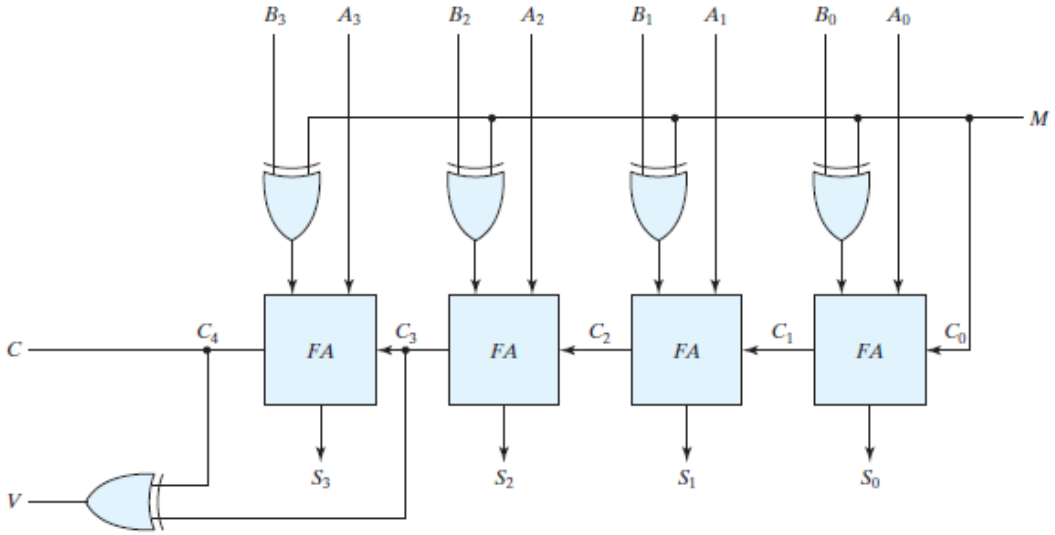

FIGURE 4.13
Four-bit adder–subtractor (with overflow detection)