Extra Credit Assignment – ECE 3714, Section 04 – Reese

Some students have been asking for extra credit opportunity in the form of another test, homework, etc.

You can earn optional extra credit by creating and simulating the following circuits using Maxplus. Each problem will earn points that can be directly applied to a test grade. There are due dates by each problem – you must turn these in at class time. I want a printout of the schematic and waveform, as well as a 3.5" disk that contains the work. Be sure your 3.5"disk has your full name on it – I will return all diskettes during the final exam. Your diskette must *have all of the files from Maxplus* necessary to compile and simulate your design. You must supply a waveform file with your design that can be used to simulate the design. If I can't simulate your design, I will not assign any credit.

Because this is extra credit, grading will be pass/fail. Either your design works or it does not. Very little partial credit will be given, if any.

This must be your own work – if I suspect academic dishonesty (see the Policy/Syllabus) I will deduct 20 points from a test grade as well as file a formal charge of academic dishonesty.

In doing these designs, you MAY NOT USE the Altera megafunctions that allow you build parameterizable counters, muxes, etc. You must use primitive gates, DFFs, etc. The only complex gate you can use from the Altera library is the 2/1 Mux function. You also may not use any VHDL.

I will supply test waveforms for each of these problems (to be posted later in ZIP archives). A total of 25 points can be earned via this opportunity. These points will be added to the regular test grade points. If you earned all 25 pts, this would raise your final average by 4.5 pts (almost half a letter grade!).

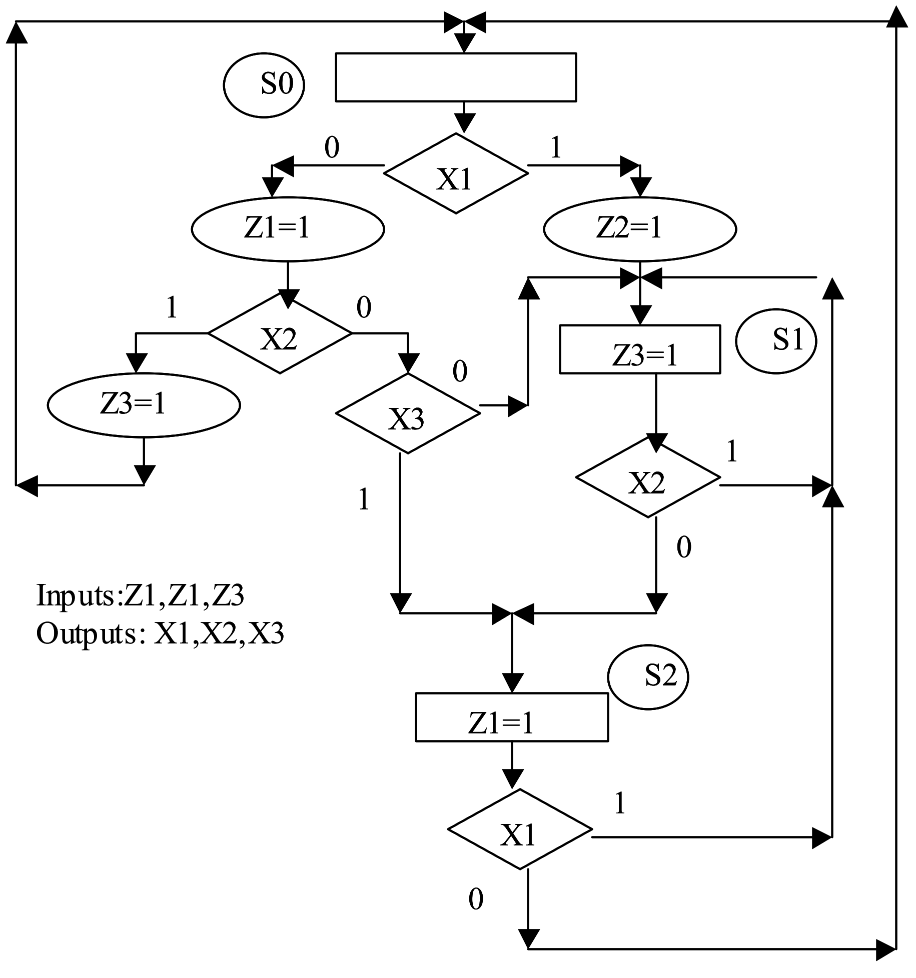
Due Tuesday, November 13th:

a. 3 pts. Build an 8-bit decrementer circuit. This combinational building block has the same interface as the incrementer, except that when EN=1, the output Y = DIN –1 (hint. Use the same circuit form as the incrementer, but the with minor modifications).

b. 4 pts. Build an 8-bit counter that has a synchronous clear, synchronous preset, asynchronous clear, parallel load, and an increment capability. You can choose your own operator precedence.

Due Tuesday, November 20th:

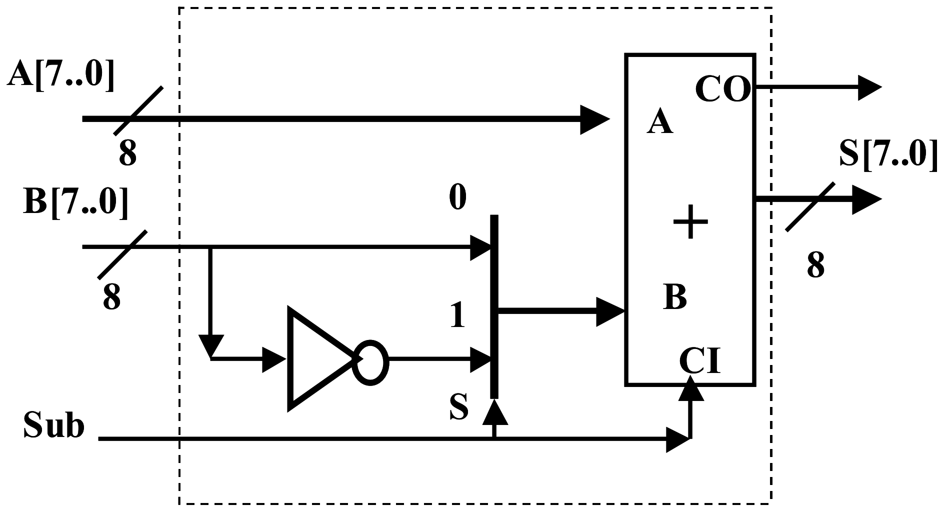
a. 4 pts. Build an 8-bit shift register that can shift either left or right, has parallel load, synchronous clear.
b. 4 pts. Implement the FSM below. You must use the following state encoding: S0=00, S1=01, S2= 11 and DFFs for state storage.



Due Tuesday, November 27th:

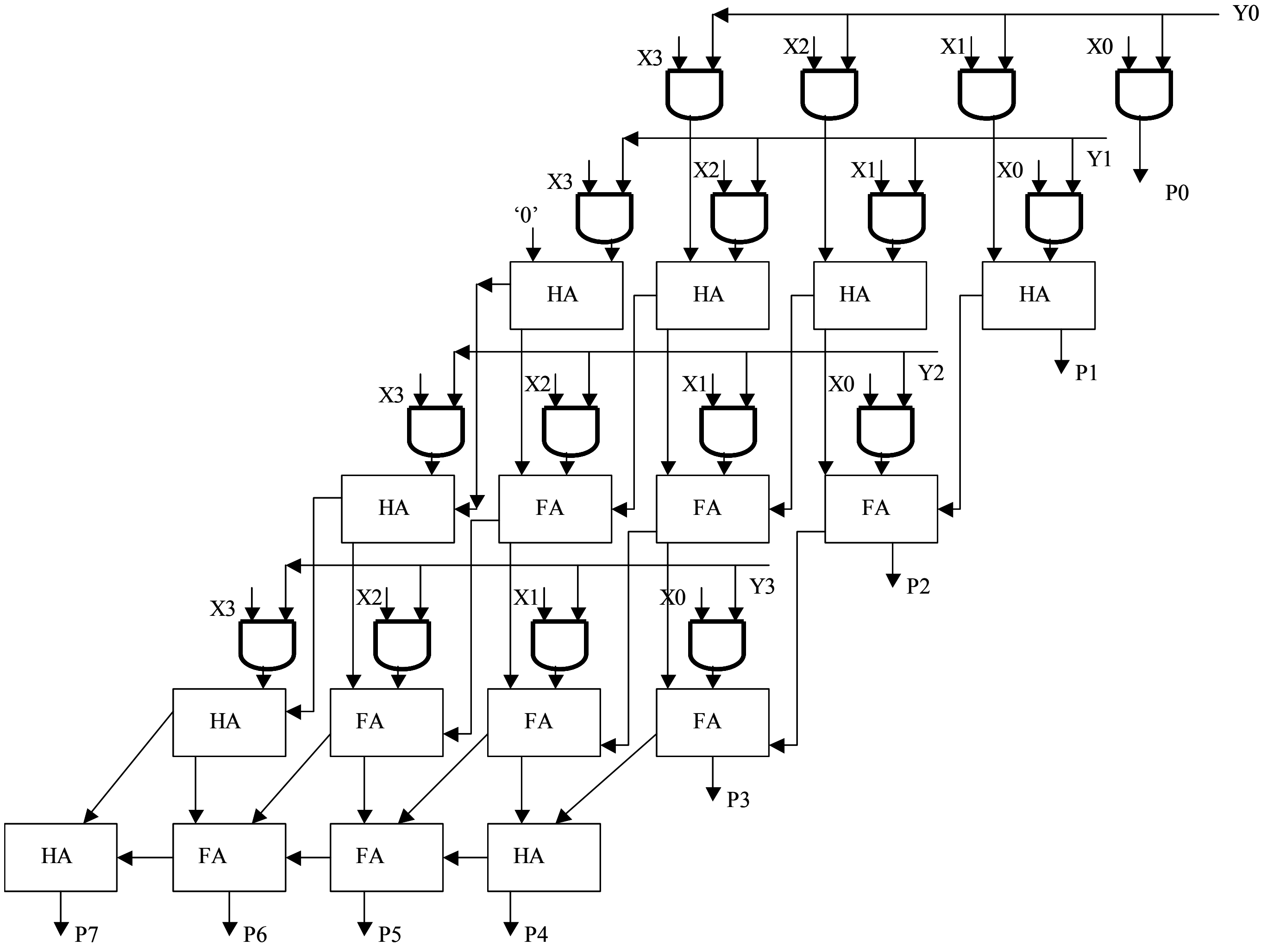
a. (4 pts) Build an 8-bit adder using the ripple carry adder discussed in the notes. The best way would be to build a 1 bit full adder cell, then connect 8 of these to form an 8 bit adder. Your 8-bit adder should have inputs of A[7..0], B[7..0], CI (carry in) and outputs of S[7..0] (sum) and CO (carry out).

b. (4 pts) Build an 8 bit adder/subtractor using your 8-bit adder as a building block. An adder/subtractor can be built as shown below. When SUB=0, an ADD is done (S = A + B). When SUB = 1, a subtraction is performed (S = A – B).



Due Tuesday, December 4th (worth 8 points)

A 4x4 combinational multiplier circuit is shown below:



A 4x4 multiplier takes two four bit inputs (X[3..0], Y[3:0]) and produces an 8-bit output P[7..0]. The blocks marked as FA are full adders (signal entering from right side is *carry in*, signal exiting from right side is *carry out*, *sum* exits from bottom of FA). The blocks marked as HA is a half-adder which implements the sum of two bits A plus B (Sum = A xor B, Carry Out = A and B). A half-adder can be thought of as a full adder with the carry-in set to '0'. Note that on the very last Half-adder, the sum bit produces P7 (most significant bit of product), and the carry out is unused. Also note that the left-most half-adder on the first row has one of its inputs set to '0'.

Create a schematic/simulation that implements this 4x4 combinational multiplier.