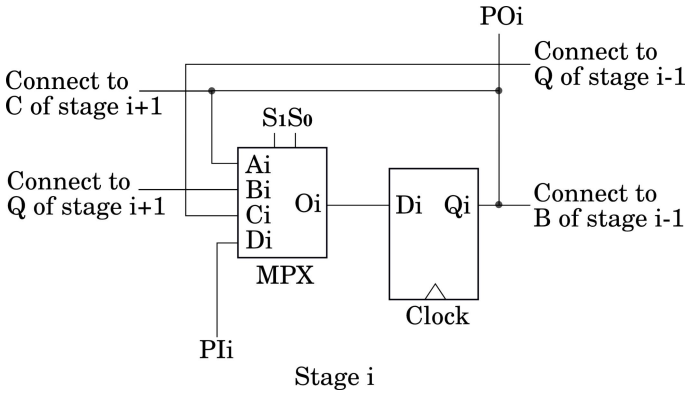
Coursework 3: Registers

In the lectures we designed a general purpose shift register in which each stage contained a multiplexer and a D-Q flip flop.

The clock and the two control inputs are common to every stage of the register. The values of S1 and S0 select one of four possible modes of action which are:

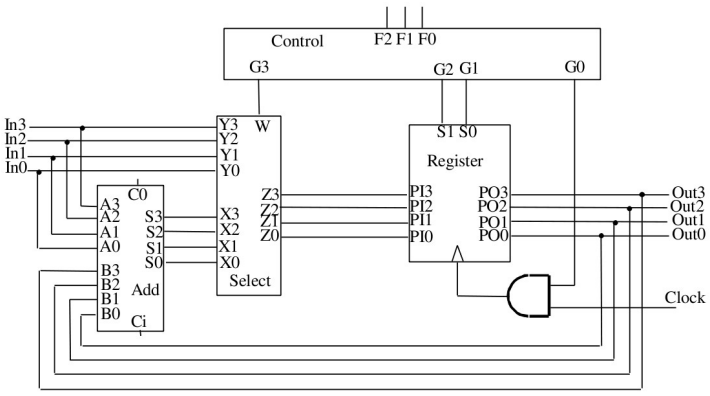
0	0	Hold
0	1	Shift Right
1	0	Shift Left
1	1	Parallel Load



Problem 1

For a new application the Hold function is not required, but instead we need a clear function. Redesign the circuit so that if S0=S1=0 the register outputs (POi) are cleared to zero.
We will now use this basic register as a building block in a new register design with added functionality. There is a need for 8 possible functions which are given in the table below.

0	Clear the output to zero
1	Shift Right
2	Shift Left
3	Parallel Load
4	Add to the register
5	Increment the register
6	Decrement the register
7	Hold



A first attempt at designing the new circuit is shown in the figure. This has the additional capability of 'Add to the Register' and 'Hold', but not 'Increment' or 'Decrement'. The box marked 'Register' is a four bit general purpose register as designed in part 1 above. The add to register function is implemented in this design with a conventional four bit full adder as described in the lectures. The parallel input to the register is chosen by a circuit marked 'Select' The input to the register is either from the adder, if the select input W is 0, or from the external input (In0 .. In3) if the select input W is 1.

Problem 2

Design a suitable circuit for the box marked 'Select' in the figure above.

Problem 3

The function is determined by the three inputs to the box marked 'Control', and is defined by the table above. Thus, for example, F2=1, F1=0 and F0=0 means that function four, add to the register, is to be selected. Notice that the hold function is neatly implemented by preventing the system clock from getting to the register clock input. Design a suitable circuit for the box marked Control, implementing the add and hold functions bearing in mind that we will be implementing the increment and decrement functions in due course.
Hint: Draw up a truth table for the outputs G0,G1,G2 and G3 assuming that for the increment and decrement options the register will be set to parallel load (G1=G2=1). You can solve using Karnaugh maps or by inspection.