A simple device CP is informally specified as follows:

CP stores values of type \( \sigma \). It has two inputs: \texttt{inp} carries values of type \( \sigma \) and \texttt{func} carries 2-bit words; it has an output line \texttt{outp} carrying the value stored. CP is a state machine whose next state depends on the value input at \texttt{func}:

- if \texttt{func} = 0 the value stored is unchanged
- if \texttt{func} = 1 the value on \texttt{inp} replaces the value stored
- if \texttt{func} = 2 the value stored is transformed by a function \( f_2 : \sigma \rightarrow \sigma \)
- if \texttt{func} = 3 the value stored is transformed by a function \( f_3 : \sigma \rightarrow \sigma \)

Define a predicate \( \text{CP} \) that formalises this specification in higher order logic.

Write down logical models of the following components:

- a combinational multiplexer that routes one of four \( \sigma \)-valued inputs to a single output, depending on the value of a 2-bit control input; [3 marks]
- a unit-delay register that holds values of type \( \sigma \). [3 marks]

Draw a schematic diagram of an implementation of CP built from these components. [3 marks]

Write down a formula that expresses the correctness of your implementation. [4 marks]

Discuss briefly how you would go about proving your correctness formula. You need not give a detailed proof, but you should aim to convince the reader that given time you could produce one. [3 marks]