Create a classic state diagram for the state machine described here.

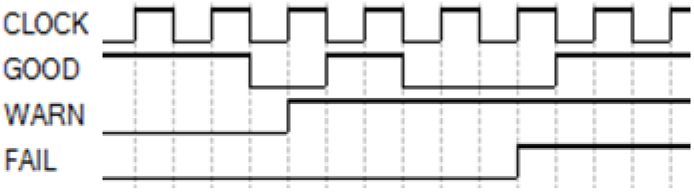
You have been asked to design a state machine to monitor the quality of widgets produced by a machine. The machine produces one widget each cycle, and a sensor determines if that product is good or bad. The goal of the state machine is to keep track of bad widgets, as described later. The edges of the state machine's clock are synchronized with the machine's output, so that the state machine gets clocked once each production cycle.

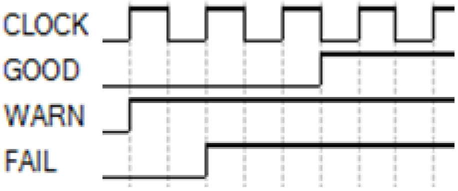
The state machine has one input named "GOOD". The signal is active ("high", 1) when a good widget was produced during that cycle. Since a widget is produced every cycle, if GOOD is *not* asserted, then the widget was bad.

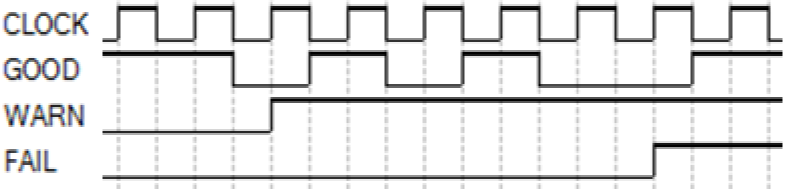
The state machine has two outputs named "WARN" and "FAIL". The following rules describe how the outputs should behave:

1) The outputs should be low after reset.
2) The first time that a bad widget is produced, the WARN output should be asserted ("high", 1). It should remain asserted until the state machine is reset, regardless of future good or bad widgets.
3) If two bad widgets are produced in a row, the FAIL output should be asserted ("high", 1). It should remain asserted until the state machine is reset, regardless of future good or bad widgets.

Here are some examples of correct behavior. Keep in mind that transition arcs in the state diagram represent possible transitions on rising edges of the clock.


CLOCK
GOOD
WARN
FAIL


CLOCK
GOOD
WARN
FAIL


CLOCK
GOOD
WARN
FAIL