Starfleet command has commissioned a new class of starship to be used in the outer reaches of the galaxy. One of its most advanced defensive features is its ability to regenerate its thought-probe shields when under intensive mental attack. The most dangerous such attack is a telekinetic attack, that disrupts the black hole matrix harnessed by the wormhole drive engines. It isn’t immediately obvious when a black hole has broken free of its constraints, but inevitably it starts consuming everything around it, and when that happens, the only thing to do is to abandon ship.

That brings us back to the thought-probe shields. You’ve been given the task of designing a logic circuit that can sound the abandon ship klaxon when the shields have been breached. Now, thought-probe shields are pretty tenuous things, and it’s virtually impossible to determine whether they’re intact or not. A telepath under ideal conditions can check them out in a couple of days, but under battle conditions that isn’t feasible.

However, the self-regenerating behaviour of the shields is well understood. They will regenerate successfully provided three probes never occur within three thought cycles. Starfleet command consider this to be a statistically acceptable characteristic, because of the rarity of telepaths who are capable of mounting a full thought-probe attack.

Fortunately the thought-probes themselves can easily be detected, and hardware that can detect 0, 1, 2, or 3 probes within a single thought-cycle exists. One of the output lines below will be set to true at each clock cycle.

Design a circuit that will send an activate signal to the klaxon if three thought-probes have occurred within three thought-cycles. You may assume that you have available

- The detector described above, which is reset after each thought-cycle
- A clock running at the same frequency as thought-cycles

Hints:

Don’t attempt to solve this problem with an ASM.
Consider the use of a shift-register
Starfleet Admiral is concerned about the possibility that the fleet has been infiltrated by an enemy agent, who will assassinate her to provoke an ill-considered reaction from her second-in-command. She has therefore adopted an elaborate smokescreen to conceal her whereabouts. This involves, amongst other more secret ploys (if I told you what they were, I’d have to kill you), the use of two flagships (Horrible and Terrible). She will move between the two flagships at random, using an automatic shuttle (no crew to betray her), leaving behind her a body double - deprived of the ability to make major decisions – whenever she travels to the other ship. To further muddy the waters, the Admiral has decreed that the shuttle should be available to transfer other items between the two flagships. Therefore, at any time the shuttle may be at the ship she’s in, or at the other.

You task is to design an ASM (ASM diagram, transition tables and circuit diagram) to control movement of the shuttle between flagships. In each ship, when the Admiral or someone else with cargo to transfer, calls the shuttle, a flip-flop is set to TRUE (outputs from the flip-flops are called CARGO_FOR_HORRIBLE, and CARGO_FOR_TERRIBLE). Your ASM has access to the output from these flip-flops, and should reset them (using signals RESET_CFH, RESET_CFT) when the shuttle has reached its destination. The shuttle generates a signal, DOOR_CLOSED, when its door is shut. The ASM should open the door for at least one clock cycle before the shuttle leaves a ship, to allow the Admiral or other cargo to be loaded. The ASM should generate the following signals: GO_TO_HORRIBLE and GO_TO_TERRIBLE (these have to be true for the entire trip), OPEN_DOOR, CLOSE_DOOR (these only need to be set for a single clock cycle). All other details of the shuttle, such as firing up its power source, navigating between the two ships, and so forth, are being handled by other systems.

The diagrams below show how the two flip-flops are configured. When a crew members on either ship presses the Transfer_to_Horrible_request button or the Transfer_to_Terrible_request button, the appropriate flip-flop is clocked and loaded with the value TRUE, so that the ASM can check the value of the output from the flip-flop as is convenient.