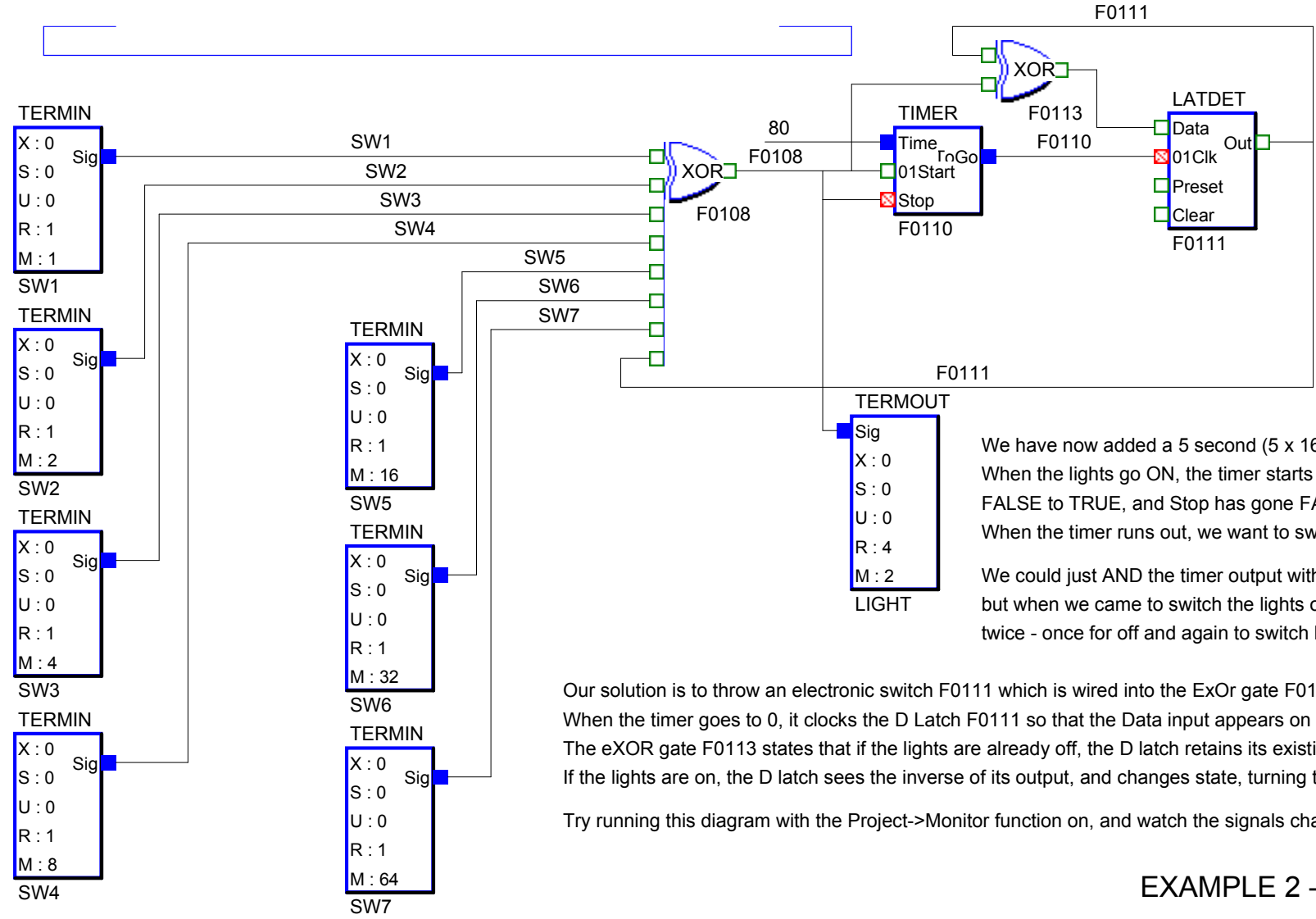


The corridor has 7 lighting switches in it. There is one light switch by each entrance door, and one outside each office door. Changing the state of any light switch should change the state of the lights. For example, Bella comes in by the right hand door and finds the lights off. She switches on the lights using SW7, goes to her office and turns the lights off using SW3. Dawn comes out of her office, switches on the lights with SW5, and forgets to turn them off. Finally, Bella goes home via the left hand door and switches the lights off by means of SW1.



We have now added a 5 second ( $5 \times 16 = 80$ ) timer to limit the time that the lights are on. When the lights go ON, the timer starts because the 01Start has gone from FALSE to TRUE, and Stop has gone FALSE (see how we have inverted its input pin). When the timer runs out, we want to switch off the lights if they are on.

We could just AND the timer output with the Lights-on signal F0108 but when we came to switch the lights on again, we would have to operate a switch twice - once for off and again to switch back on (you might like to try this).

Our solution is to throw an electronic switch F0111 which is wired into the ExOr gate F0108 along with all the other switches. When the timer goes to 0, it clocks the D Latch F0111 so that the Data input appears on the output. The eXOR gate F0113 states that if the lights are already off, the D latch retains its existing state. If the lights are on, the D latch sees the inverse of its output, and changes state, turning the lights off.

Try running this diagram with the Project->Monitor function on, and watch the signals change.

EXAMPLE 2 - Corridor Lighting with Timer