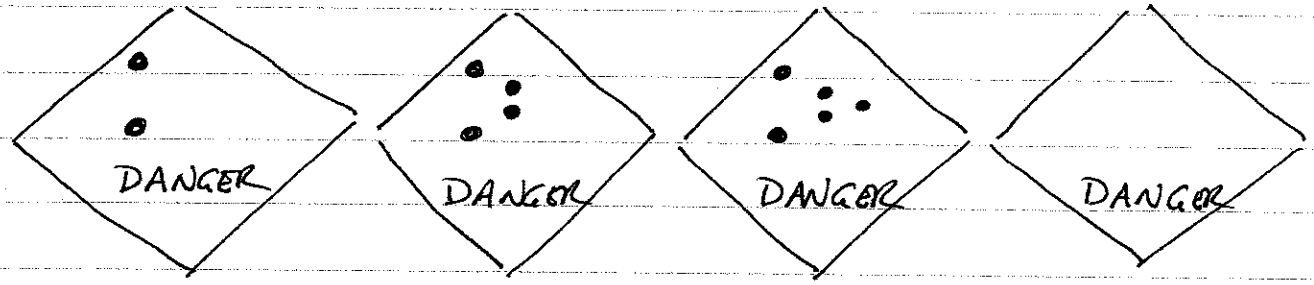
EECE Dept
The University of Texas

EE 306 
Supplemental Note
Sept. 27, 2006

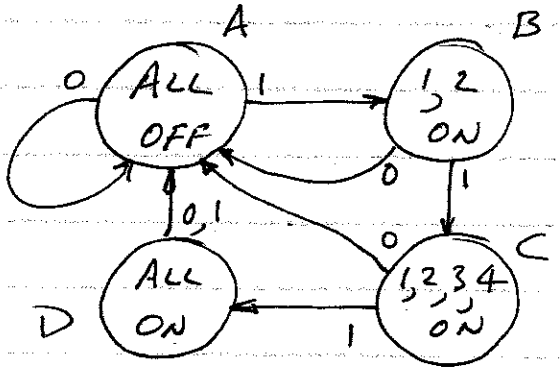
On Monday, I designed for you a combinational (or, rather sequential) lock, complete with Master-Slave flip-flops. This is a tough concept, one worth repeating. So, in these notes, I will try to carefully take you through the design of another sequential circuit, again complete with Master-Slave Flip-flops.

First, a statement of the problem: Design a controller for a warning sign that operates as follows when the input switch is in the ON position. That is, with the Switch ON, the goal is to have the lights flash as shown below: 1st two ON, then four, then all five (completing the "arrow"), then all off, then two, etc. With the Switch OFF, all lights remain OFF.




1 2 3 4
5 6 7 8
9 10 ...

We can represent the behavior of our controller with a finite state machine, thus:




Note there are four states, which we label for convenience A, B, C, D.

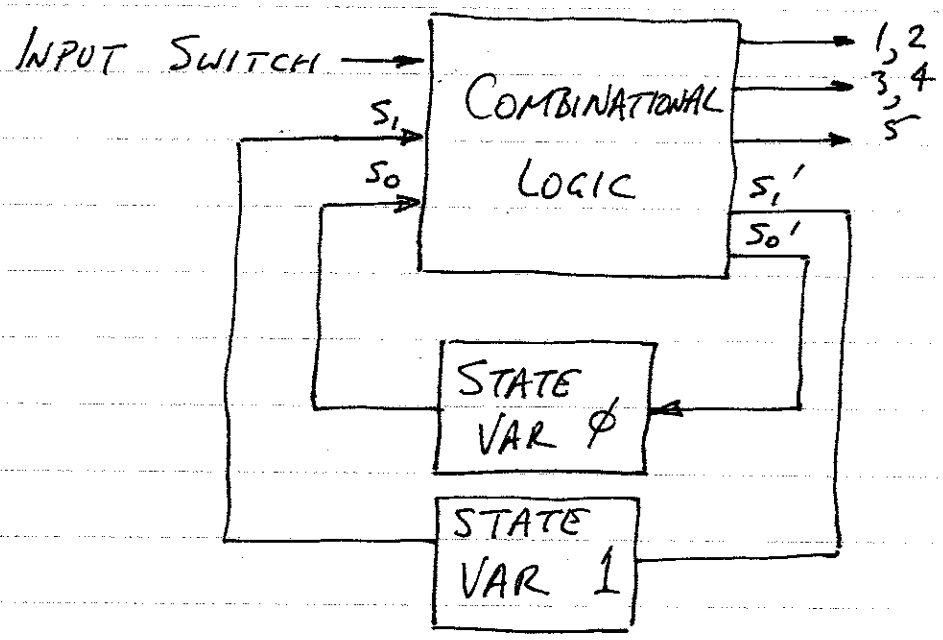
Note there is one input (the Switch). When ON, we cycle through the states. When OFF, we (go to and) remain in State A

Note that we associate the outputs with the States ONLY

We can also represent the finite state machine as a State Table, and assign a unique code to each state:

| $S_1 S_0$ | | SW OFF | SW ON | 1, 2 | 3, 4 | 5 |
|---|---|---|---|---|---|---|
| (00) | A | A | B | 0 | 0 | 0 |
| (01) | B | A | C | 1 | 0 | 0 |
| (10) | C | A | D | 1 | 1 | 0 |
| (11) | D | A | A | 1 | 1 | 1 |

Our job is to design the logic to implement this state table. First an overall block diagram



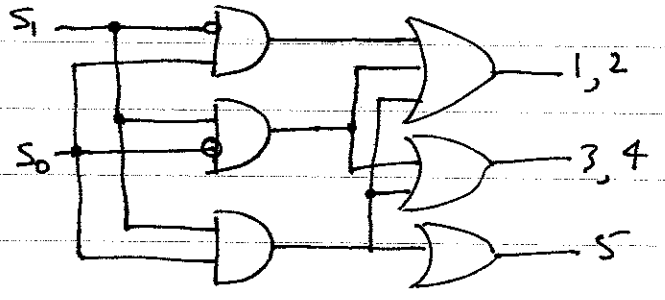

WE HAVE THREE TASKS:

- COMBINATIONAL LOGIC FOR THE LIGHTS
- COMBINATIONAL LOGIC FOR THE "NEXT STATE"
- DESIGN OF THE FLIP-FLOPS

THE OUTPUT LIGHTS:

| | STATE | 1,2 | 3,4 | 5 |
|---|---|---|---|---|
| A | 0 0 | 0 | 0 | 0 |
| B | 0 1 | 1 | 0 | 0 |
| C | 1 0 | 1 | 1 | 0 |
| D | 1 1 | 1 | 1 | 1 |

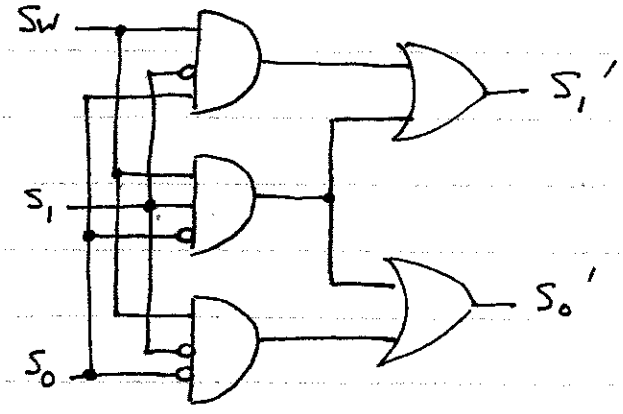
COMBINATIONAL LOGIC



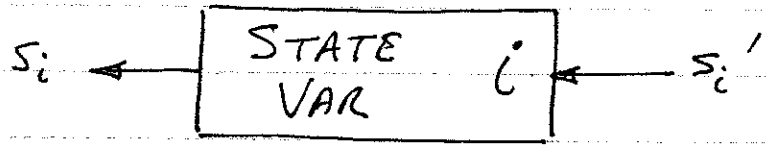
NEXT STATE FUNCTION:

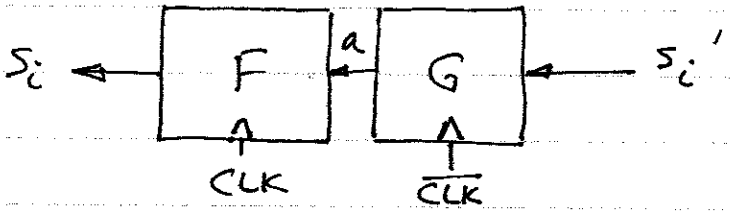
| | $S_1$ $S_0$ | $S_1'$ $S_0'$ |
|---|---|---|
| SW | STATE | NEXT STATE |
| 0 | 0 0 | 0 0 |
| 0 | 0 1 | 0 0 |
| 0 | 1 0 | 0 0 |
| 0 | 1 1 | 0 0 |
| 1 | 0 0 | 0 1 |
| 1 | 0 1 | 1 0 |
| 1 | 1 0 | 1 1 |
| 1 | 1 1 | 0 0 |

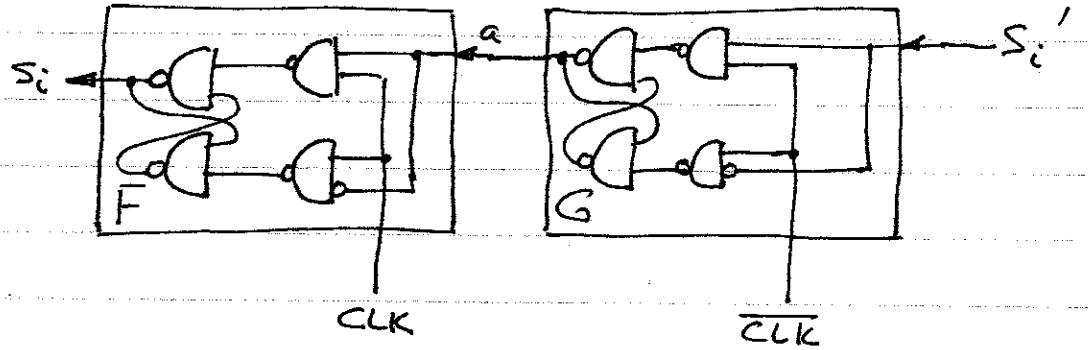
COMBINATIONAL LOGIC

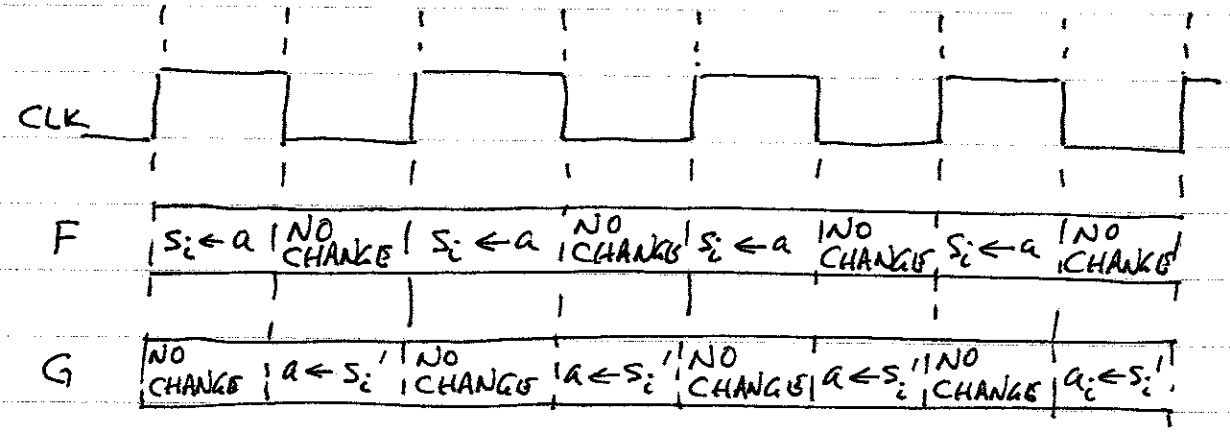


# THE MASTER-SLAVE FLIP-FLOP

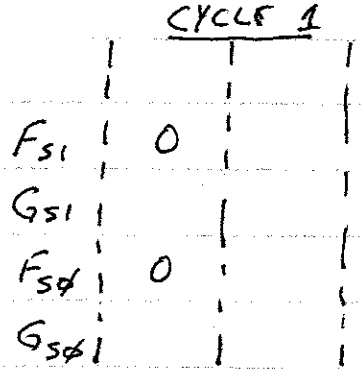



EXAMPLE: WE START IN CYCLE 1 IN STATE A (0,0) WITH THE SWITCH ON.




IN THE FIRST HALF OF CYCLE 1, $\overline{CLK}$ PREVENTS $G_i$ FROM LATCHING $S_i'$

IN THE SECOND HALF OF CYCLE 1, $\overline{CLK} = 1$, SO $S_i'$ CAN BE LATCHED IN $G_i$. BUT, SINCE $CLK = \phi$, THE OUTPUT OF $G_i$ (WHICH IS $q_i$) CAN NOT BE LATCHED IN $F_i$. THE RESULT:

| | CYCLE 1 | |
|---|---|---|
| $F_{S1}$ | 0 | 0 |
| $G_{S1}$ | | (0) |
| $F_{S0}$ | 0 | 0 |
| $G_{S0}$ | | (1) |

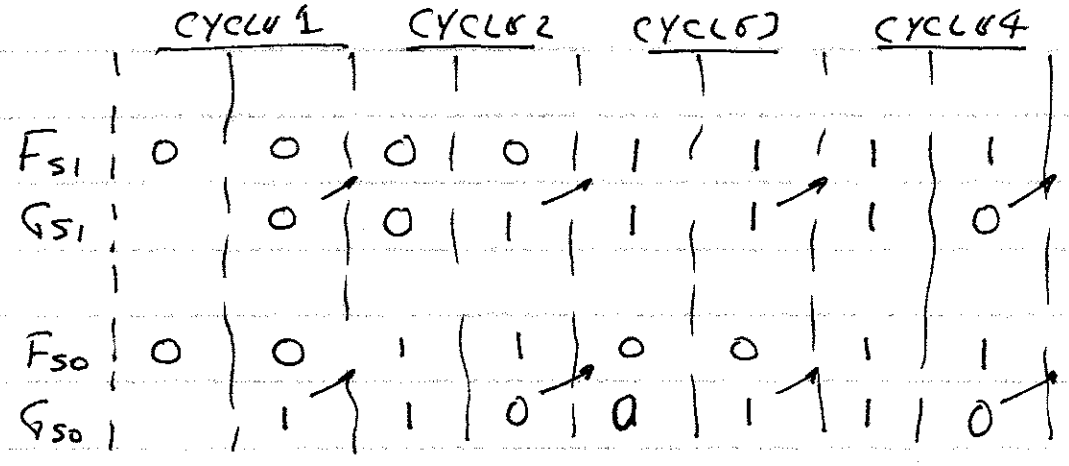
AT THE START OF CYCLE 2, $CLK = 1$, SO $F_i$ CAN LATCH $q_i$ (THE OUTPUT OF $G_i$). SINCE $\overline{CLK} = 0$, THE SUBSEQUENT GENERATION OF THE NEW $S_i'$ CAN NOT BE LATCHED IN $G_i$. THE RESULT

| | CYCLE 1 | | CYCLE 2 | |
|---|---|---|---|---|
| $F_{S1}$ | 0 | 0 | (0) | |
| $G_{S1}$ | | 0 | 0 | |
| $F_{S0}$ | 0 | 0 | (1) | |
| $G_{S0}$ | | | | |

IN THE SECOND HALF OF CYCLE 2, $\overline{CLK} = 1$, SO THE NEXT $S_i'$ CAN BE LATCHED IN $G_i$. BUT SINCE $CLK = \phi$, THE OUTPUT OF $G_i$ CAN NOT BE LATCHED IN $F_i$.

| | CYCLE 1 | | CYCLE 2 | |
|---|---|---|---|---|
| $F_{S1}$ | 0 | 0 | 0 | 0 |
| $G_{S1}$ | | 0 | 0 | (1) |
| $F_{S0}$ | 0 | 0 | 1 | 1 |
| $G_{S0}$ | | 1 | 1 | (0) |

WE CAN CONTINUE ON, TO CYCLE 3, THEN CYCLE 4, ETC. THE RESULT




STATE: 00 01 10 11 00

$F_{S1}$, $F_{S0}$ SPECIFY THE STATE

$G_{S1}$, $G_{S0}$ LATCH THE NEXT STATE IN THE SECOND HALF OF EACH CYCLE.