Suppose we want a memory device that has the following functionality … to ***HOLD*** the present memory value, to ***SET*** the memory to a '1', to ***RESET*** the memory to a '0', and to ***TOGGLE*** or invert the present memory value.

Let's begin by defining Q(t) as the present value stored in the memory and Q(t+1) as the next (or future) memory value. Labeling the input controls as ***J*** and ***K*** we can establish the operation of the memory device as:

J	K	Q(t)	Q(t+1)	Function
0	0	0	0	Hold
0	0	1	1	
0	1	0	0	Clear
0	1	1	0	
1	0	0	1	Set
1	0	1	1	
1	1	0	1	Toggle
1	1	1	0	

Using an SR Flip-Flop, if we connect the negation of S to the R input we get the ability to set or reset the memory only. Furthermore, the memory would be set if the input S were '1' and reset if S were '0'. If we consider the Q(t+1) column as the desired input to this structure, then we could create the desired memory device!

So, let's think about Q(t+1) as a desired logic function based upon the input variables J, K, and Q(t). Thus,

$$Q(t+1) = \sum m(1,4,5,6)$$

Using a K-map (or Boolean algebra), we can reduce the logic to …

$$Q(t+1) = J\,\overline{Q(t)} + \overline{K}\,Q(t)$$

Which can be implemented as …