READ THE FINE PRINT!
TESTING CONTINUED

The above "fine print" appears at the end of the data sheet for the optoisolators found on the NOTES & THINGS" page. Note especially the part which reads (in larger print),

> ***. . . products are not designed, intended, or authorized for use in components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any application in which the failure of the … product could create a situation where personal injury or death may occur.***

These are not just idle words. The manufacturer is making it clear that its product is not guaranteed to be trouble free to the extent that they were willing to be the subject of a law suit if someone dies as a result of a failure. As a good design engineer, you need to be just as careful! How careful you need to be depends on the application. If you are producing a kids game, you probably do not need to be as careful as you would in the design of a piece of life-support equipment (other than to make sure the game doesn't kill the kid).

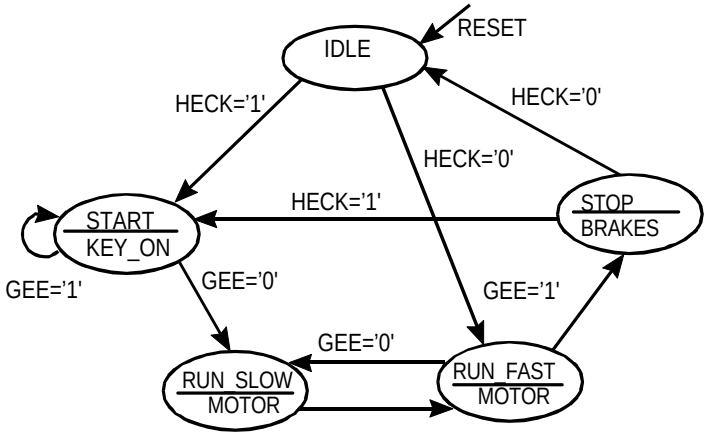
The previous notes provided for only minimal function testing. In the unconditional transfer from RUN_SLOW to RUN_FAST we assumed that HECK didn't matter and only tested for one case (HECK = 0). To be really sure, we should have verified that it made this transition with hECK = 1.

To be sure that the machine is fully functional, a machine in N states and I inputs, we have:

$$\text{ESSENTIAL TESTS} = N * 2^I$$

As before, there will likely be some axillary tests needed to get the machine in the proper state. Consider the slightly revised machine shown to the right.

Here we have a second input, GEE. Thus there are 20 essential tests pulse an unknown number of auxiliary tests.



The previous procedure may be modified to allow for all possible inputs at each state. Previously, we tested that path IDLE, RUN_FAST, STOP, IDLE first. We shall do so again but this time we shall do so twice. In the below table the inputs and outputs will be listed in abbreviated form (as are the state names) - R = reset, G = gee, H = Heck, K = Key_on, B = Brakes, M = MOTOR. In the "scoreboard" only Gee and Hee are listed in that order (01 = Not Gee and Heck)

TEST TABLE									
TESTS TO MAKE					"SCOREBOARD" (INPUTS USED)				
#	PRES	INP.	NEXT	OUT	IDLE	STRT	R_S	R_F	STOP
1	?	R	IDLE	-	-	-	-	-	-
2	IDLE		R_F	none	00				
3	R_F	G	STOP	M	00			10	
4	STOP		IDLE	B	00			10	00

Now, we proceed through this sequence again with the "don't care" input at the opposite level.

5	IDLE	G	R_F	none	00,10			10	00
6	R_F	GH	STOP	M	00,10			10,11	00
7	STOP	G	IDLE	B	00,10			10,11	00,10

Now we test the other transition from state IDLE (HECK = 1)

8	IDLE	H	STRT	none	00,10, 01			10,11	00,10
9	STRT	G	STRT	K	00,10, 01	10		10,11	00,10
10	STRT	GH	STRT	K	00,10, 01	10,11		10,11	00,11
11	STRT		R_S	K	00,10, 01	10,11, 00		10,11	00,11

We have now tested three of the four transitions at START, we must return to START to text the fouth, via RUN_SLOW, RUN_FAST and STOP, as before.

12	R_S		R_F	M	01,10, 01	10,11, 00	00	10,11	00,11
13	R_F	G	STOP	M	01,10, 01	10,11, 00	00	10,11	00,11

14	STOP	H	STRT	B	00,10, 01	10,11, 00	00	10,11	00,10, 01
15	STRT	H	R_S	K	00,10, 01	all	00	10,11	00,10, 01

This completes all of the tests for the START state. We need to ho on and take care of RUN_SLOW and RUN_FAST

16	R_S	H	R_F	M	00,10, 01	all	00,01	10,11	00,10, 01
17	R_F		R_S	M	00,10, 01	all	00,01	10,11, 00	00,10, 01
18	R_S	G	R_F	M	00,10, 01	all	00,01, 10	10,11, 00	00,10, 01
19	R_F	H	R_S	M	00,10, 01	all	00,01	all	00,10, 01
20	R_S	GH	R_F	M	00,10, 01	all	all	all	00,10, 01

Now, where are we?

					00,01, 10	all	all	all	00,01, 10

We have one remaining transition from states IDLE and STOP. We shall test the last transition from STOP first. Why? Wait and see!

21	R_F	G-	STOP	M	00,01, 10	all	all	all	00,01, 10
22	STOP	GH	STRT	B	00,01, 10	all	all	all	all
23	STRT	RST	IDLE	K	00,01, 10	all	all	all	all
24	IDLE	GH	STRT		all	all	all	all	all

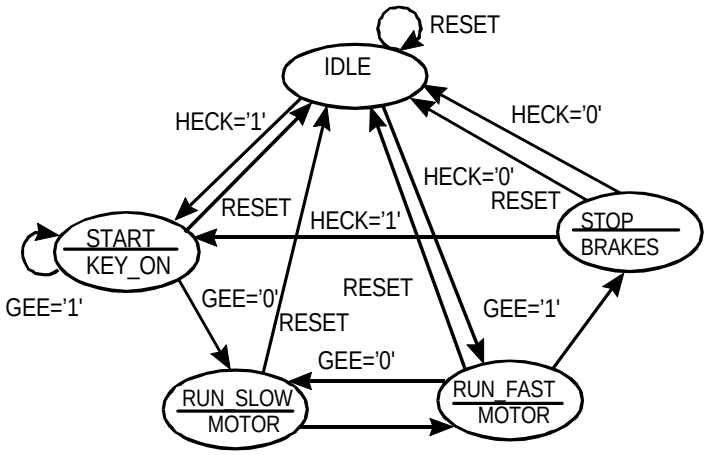
The reason for leaving the final IDLE transition to last was that we can always get to that state via the RESET. So 24 tests are needed to cover the 20 essential tests.

A word concerning RESET is in order here. Isn't it an input too? It has been assumed that RESET is asynchronous - RESET is tied directly to the asynchronous resets on the flip-flops making up the state variables. As such, a failure to reset would be a circuitry problem, not a design error. Of course, one should make sure that the flip-flops do, indeed, reset. Only one test would be needed. Place the machine in the state where all state variables are set and then RESET it. This tests all flip-flops at the same time.

Note: since this machine has only 5 states, there may be no "usable" state with a "111"

assignment. This opens a totally new can of worms that is not to be considered here.

If the RESET is synchronous and, hence, part of the Next State Decoder design, then it must be treated like any other input. This is illustrated in the diagram to the right where the explicit RESET transitions are shown.



There are now 15 transitions compared to the original ten. For the simplified test, 15 + A tests would then be required where A is now likely to be a quite a bit larger. Doing the more exhaustive test as in the process above would require at least 5*8 = 40 tests - double the present number.
