The Word Generator
(Written for MultiSim V8)

Discussion

For those of you who are tired of using switches with pull-up resistors, or for those of you who have switching actions which have to happen in many different ways at more than one time in a simulation, Multisim has provided you the we have the Word Generator (WG). Because the Word Generator has to be programmed in Hex (base 16) or in binary (base 2) (not really a good idea but possible), it is necessary for the reader to review on his/her own how these systems work and how to convert between them.

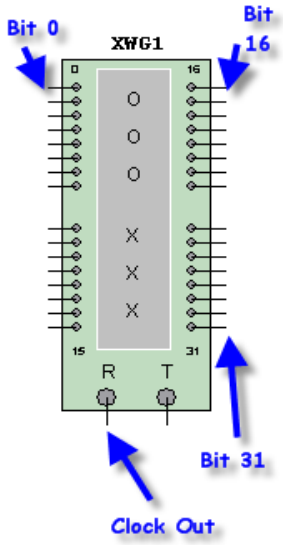
For a review of binary and hex please refer to the number system review tutorial (separate document).

The Word Generator (MultiSim)

Figure 1

Note that we are discussing the version 8 Word Generator. Before you start with the flip-flops, you need to learn how to use the most basic of functions made available to us by the Word Generator in MultiSim. Figure 1 shows where the Virtual Word Generator can be found in MultiSim.

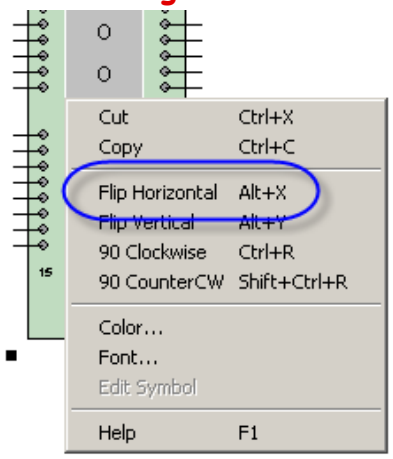
Figure 2



When you first place the Word Generator (WG) on the workspace you will see the device shown in Figure 2. It consists of a 32 bit output (Bits 0 to 31), a Clock output ("R" for 'Data Ready'), and a Trigger input (Discussed later in this tutorial).

Note that the default placement of the Word Generator places bits 0 thru 15 (the "Low Word") on the left side of the Generator. However, it seems that the most useful placement for the "Low Word" is on the right side instead.

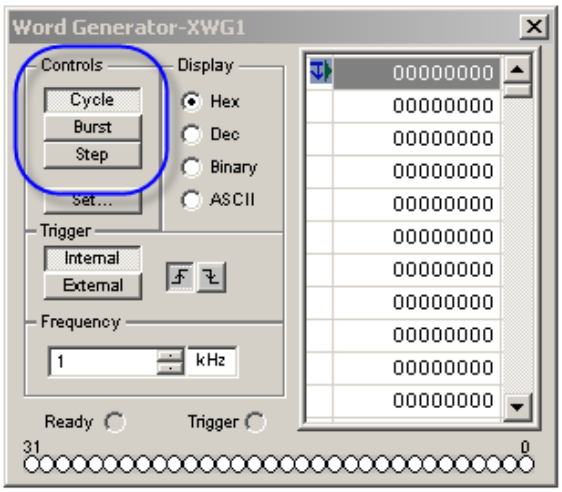
Figure 3



If you Right Click on the device the window shown in Figure 3 to the right will appear. Select the "Flip Horizontal" and the device will flip so that Bits 0 - 15 "Low Word" and the Clock output (R) are on the right where they will be more available. This is the way you will see them in my lab documents.

Lets now open up the Generator by performing a Double Left click and examine the contents.

Figure 4



As seen in Figure 4, there are 6 major sectors within the device.

The controls section circled in Figure 4 can now be discussed.

After entering data in the data section, you need to be able to decide how the data will be sent to your circuit. With each clock cycle, one row of data is made available on the 32 outputs in binary form (a 32 bit word). The control section controls the manner in which this data is sent to the circuit.

Cycle: This method will cycle continuously thru the entire program. (SEE COMMENT ABOUT BUG ON LAST PAGE)

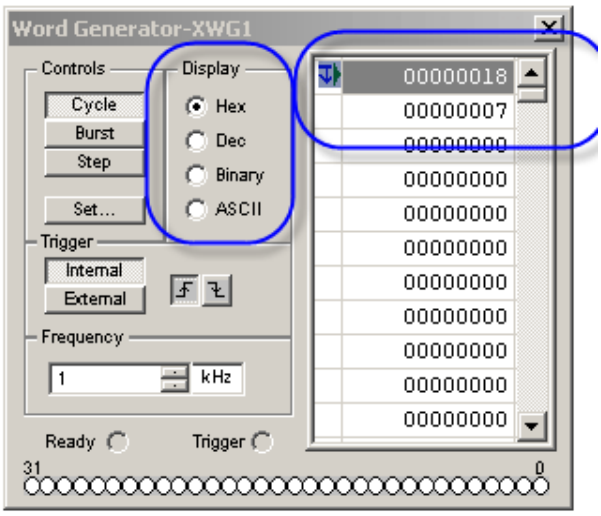
Burst: This method will start from row 0 and stop at the final row (unless a breakpoint, discussed later) is in place. It will wait until the "Pause/Resume" button next to the system on/off switch has been activated to perform another burst.

Step: This method steps thru the data one row at a time as the "Pause/Resume" button is activated. You will use this one to observe slowly how the circuit is operating, perhaps while troubleshooting.

In Figure 5 (next page) note that to the right of the controls section is a Display area. The programmer is given the choice of how the data in the area further to the right is displayed (and how it will be programmed). You are given a choice between Hex (preferred), Decimal, Binary, or ASCII. Lets assume that HEX is chosen.

The row of HEX numbers represents the 32 output terminals at the bottom of the window with each Hex digit controlling 4 of those bits. Note that the LSB is the farthest output terminal to the right. For example: Program line 0 has been programmed with a Hex 18 (decimal 24) which is a binary 11000_2.

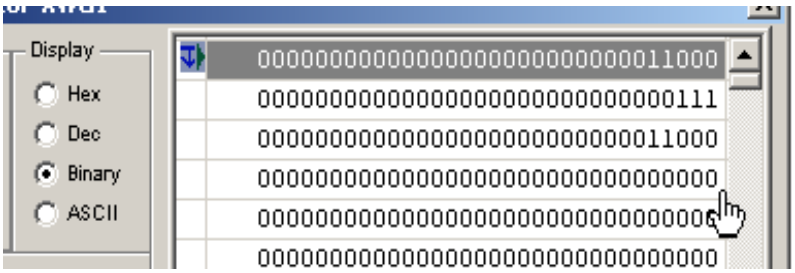
Figure 5



When this line is sent to the output terminals, a

0 will be output on bits 0, 1, and 2 outputs
1 will be output on bits 3 and 4 outputs

All higher bits will output 0's. These values will be presented to whatever wires are attached to the outputs as 5 volts (logic 1's) and 0 volt (logic 0's) signals. Note that you can view / enter these values as other than hex. The Figure below is what you would see if you were to have selected Binary within the display value area.



This can come in handy when you want to see what the actual logic values will be. On top of that, if you desire to enter decimal values, you can select Dec instead. I do find it easier to use HEX values but everyone is different.

Next lets examine the Set button as shown in Figure 7. When this button is selected, you will see a window like the one shown in Figure 8 below.

Figure 7

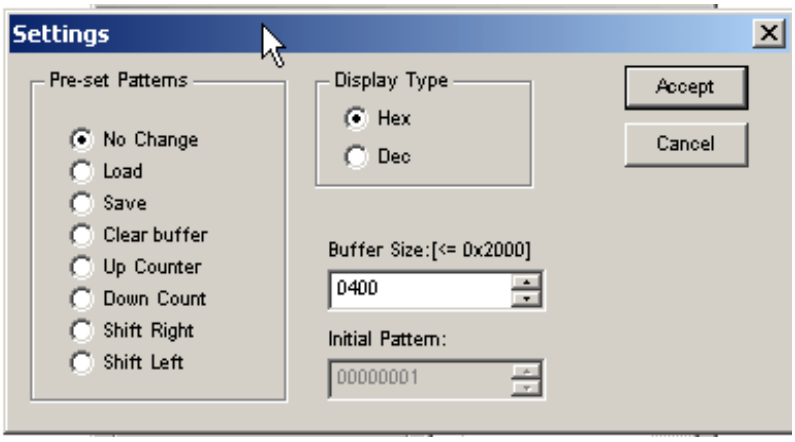


Figure 8

The Display Type section sets the default for the Display area in the main window.

The Buffer size sets the number of rows in HEX. This is essentially a method to set up the SIZE of the current program or the number of rows that will be displayed in the Display area.

The Preset Patterns section is used to set the initial value for the Up Counter, the Down Counter, the Shift Right, or the Shift Left choices. (Play with these to see what they do).

The Clear buffer does just what it says it does. It clears the program to all 0's.

The Save choice allows you to save a program to be Loaded later.

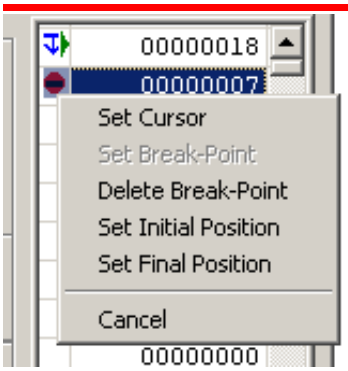


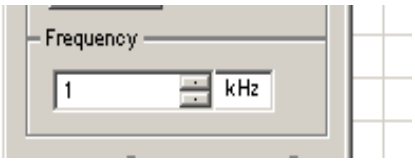
Figure 9

If you click in the area to the left of a row the window shown in Figure 9 will appear. It allows you to use several methods to troubleshoot your program as well as to set beginning and ending positions for running smaller portions of a larger program.

You can set the current cursor position (the program pointer in 'computerese') on a specific row,

Sometimes when troubleshooting it is handy to set a Breakpoint. This allows the burst or the cycle mode to run up to but stop before executing the Breakpoint Row. (Row 1 {the second visible row} has a Breakpoint set on it). Then you can single step thru the break-pointed line and as far as you desire after that to see how the rows affect the circuit.

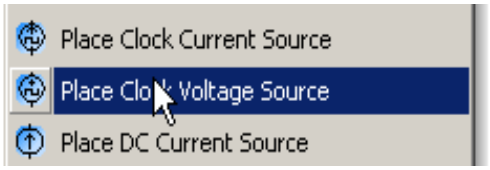
Sometimes it is desired to only operate a portion of a program in a Cycle or a Burst. You can do that by setting the Initial and Final Position of the rows you desire.

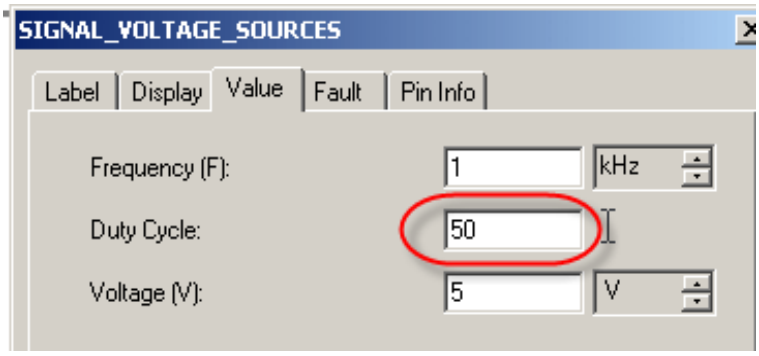


Finally we get to the Frequency section. The Word Generator will output a rectangular wave (NOT 50% Duty Cycle). {If a different duty cycle is desired, see the following page} Each cycle will cause one row to be placed on the output ports. You can set the frequency of the waveform in this section. If you are using any of the settings except for "Step", you will have to set in your desired frequency in the frequency section. The clock output is called "Data Ready." It is the "R" output on the device.

Obtaining a different Duty Cycle then the one output by the Word Generator's internal trigger

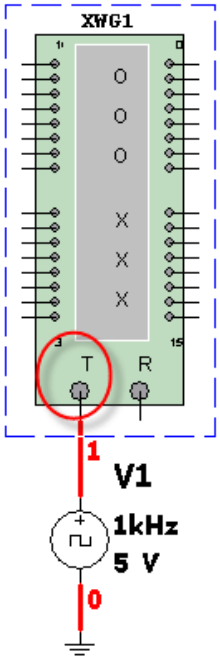
As indicated earlier, the Word Generator will output a rectangular wave (NOT 50% Duty Cycle). If this is not desired then Select External trigger as shown below. Then connect a Clock Voltage Source from the Sources Parts Bin.

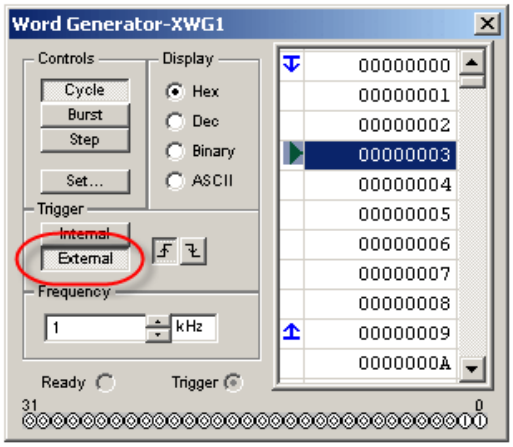




Once you have the source, double click on it and modify the Frequency and the Duty Cycle to the desired values.

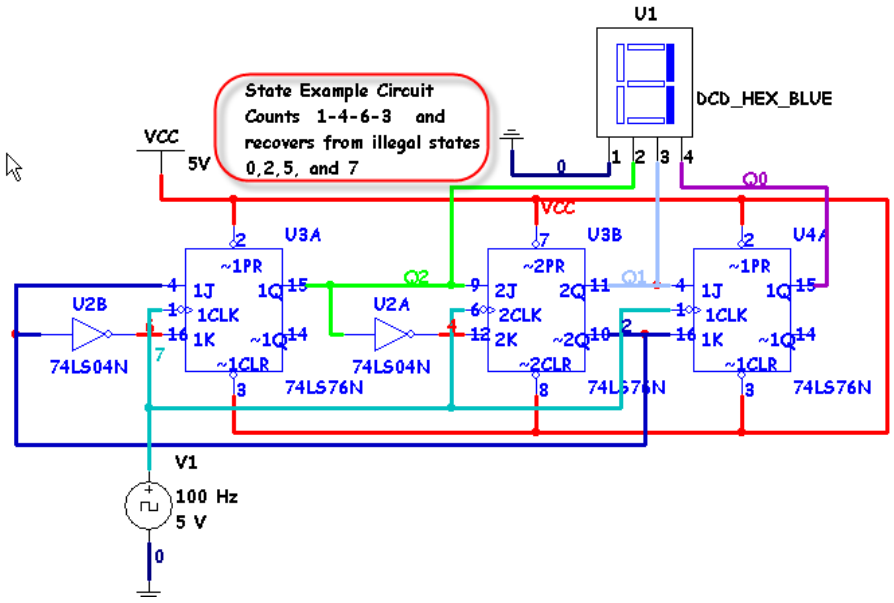
Finally, connect the clock source to the T input of the Word Generator. Select EXTERNAL within the Trigger section and select the edge of the clock from which you want to trigger the output.





Example 1:

There exists a state machine which will count the following sequence: 1-4-6-3
One of the issues with a state machine is that it should be able to recover without failure from the occurrence of an ILLEGAL STATE (for this state machine, 0, 2, 5, 7). It is desired to test this circuit to show that it can recover from an illegal state. To do this it is necessary to use the PRE and CLR inputs to place the state machine in an illegal state then return the PRE and CLR's to their inactive state (1).

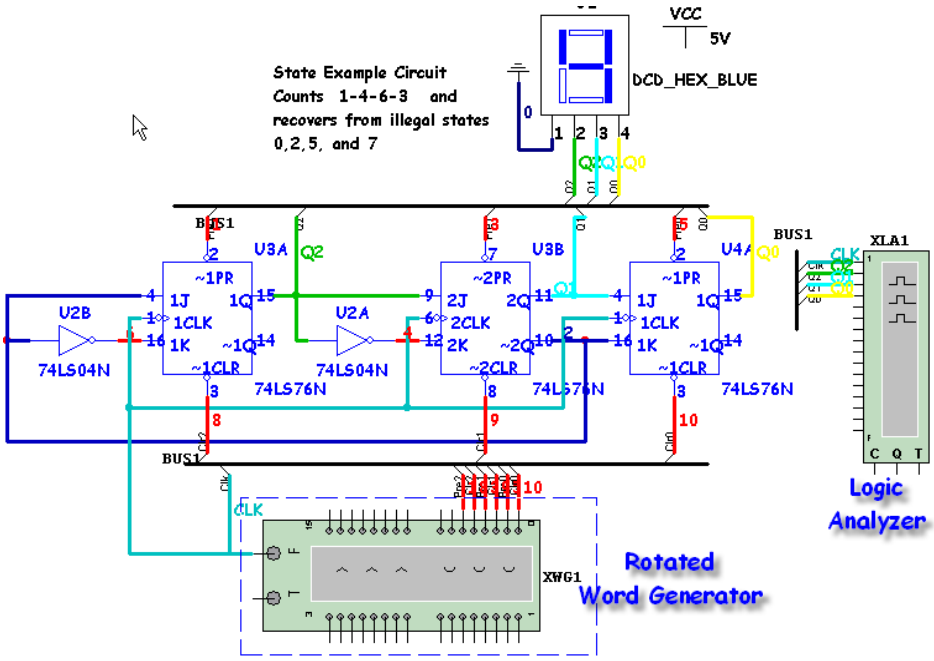


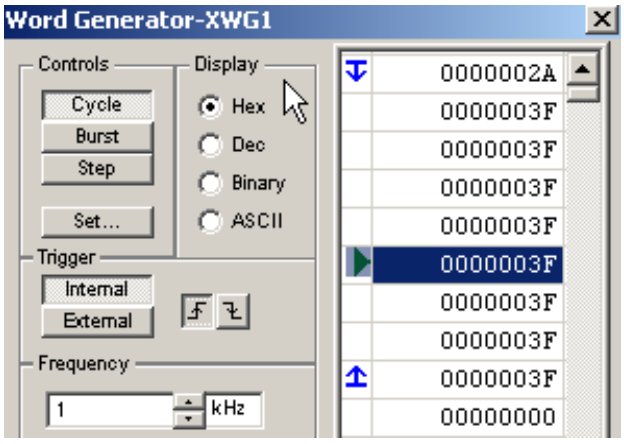
We will replace the clock with a Word Generator. We will connect the 3 PRE's and 3 CLR's (currently attached to V_{CC} to keep them inactive) with connections to the LOW Byte of the WORD GENERATOR.

Before doing so, lets calculate the necessary COMMAND LINES for each of the illegal states:

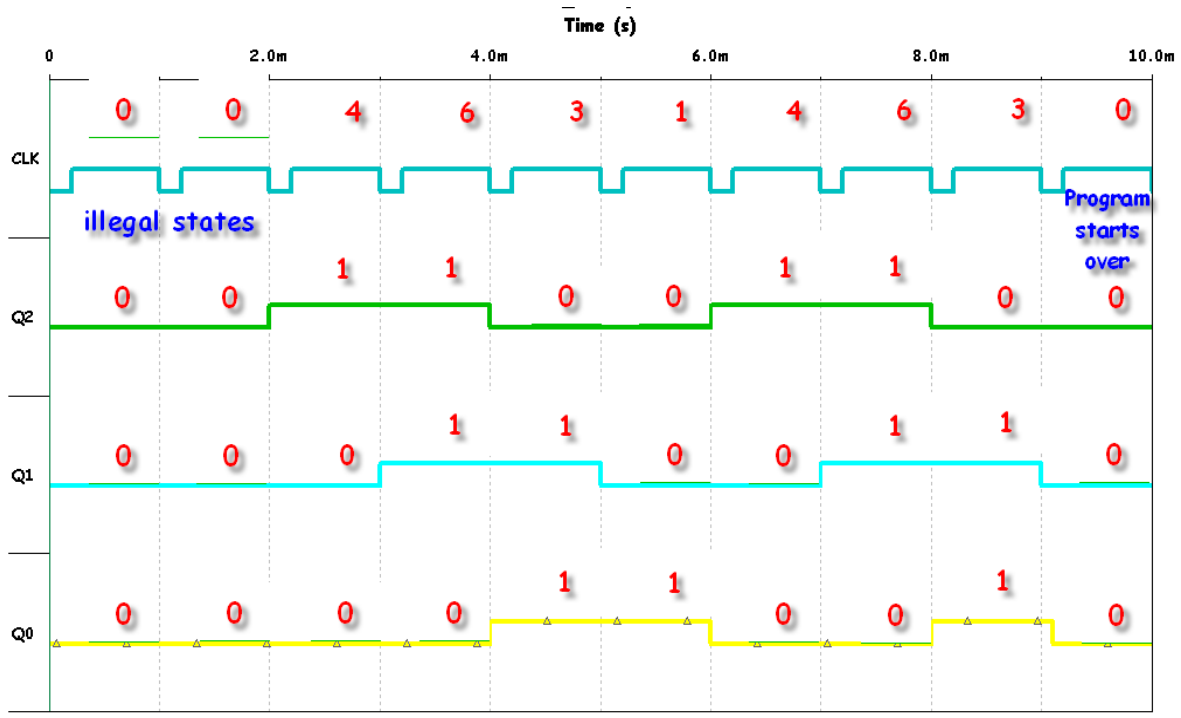
Illegal State	PRE2	CLR2	PRE1	CLR1	PRE0	CLR0	CMD
0 (000)	1	0	1	0	1	0	2A
2 (010)	1	0	0	1	1	0	26
5 (101)	0	1	1	0	0	1	19
7 (111)	0	1	0	1	0	1	15
inactive	1	1	1	1	1	1	3F

Next, attach the WORD GENERATOR and program it so that the State Machine starts in an illegal state and then the PRE's and CLR's are returned to an inactive state. The Word Generator will provide the Clock Pulses to allow the State machine to cycle thru two complete count sequences. Note that in order for this to happen there will have to be one program line per clock pulse. Two cycles plus one illegal state means that there has to be 9 program lines.



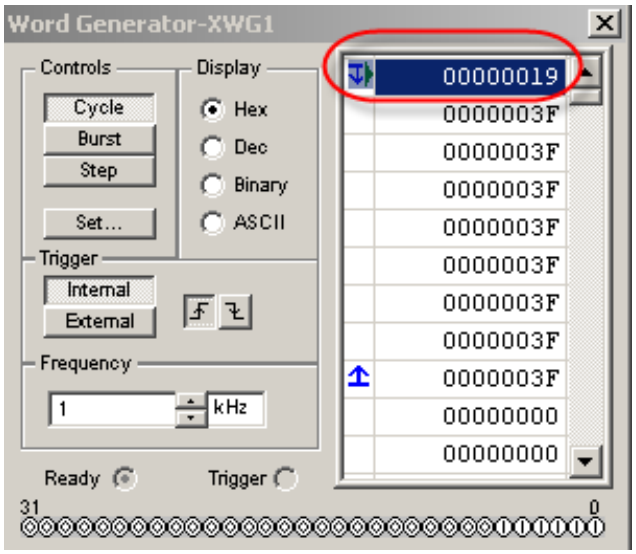


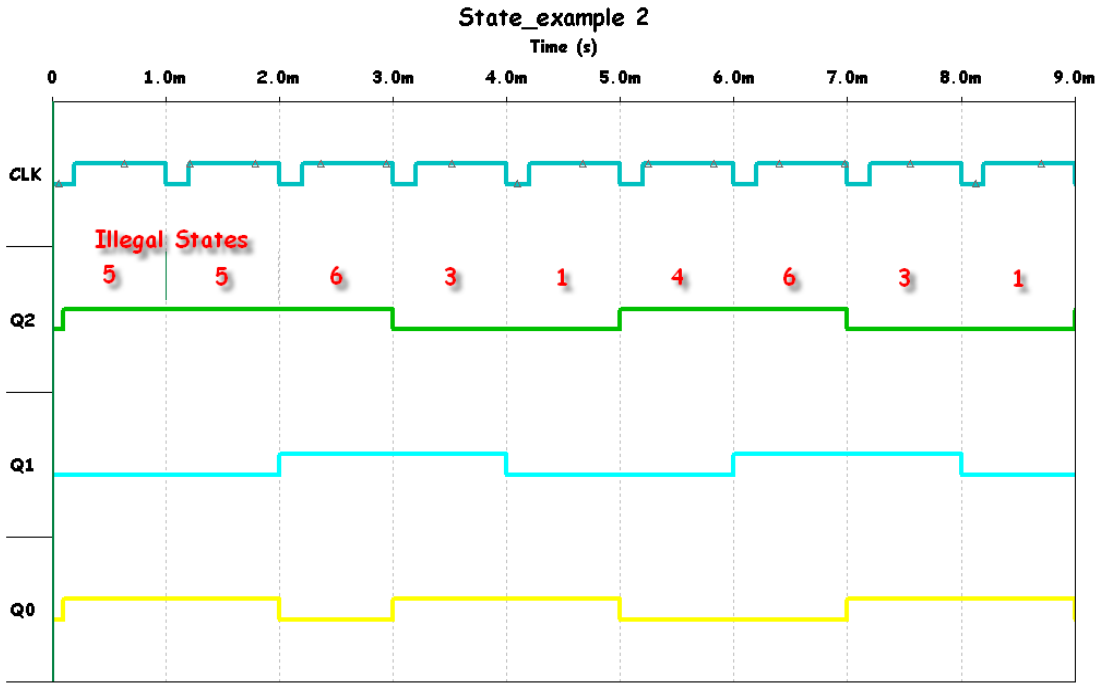
See the graph on the next page.



Lets reprogram the Word Generator to verify one more Illegal State recovery.

See the circuit on the next page.





It is left for the user to see if they can use the Word Generator to test the recovery ability of the State Machine to recover from the other 2 illegal states.

BUG In the WORD GENERATOR:

Ordinarily, if you desire to stop a simulation while using the Word Generator all that is necessary is to click on the on/off switch. Currently, in Version 8 there is a bug where if the Word Generator is in BURST or STEP Modes, the ON/OFF Switch does not work. The current fix for this is to first place the Word Generator into CYCLE mode and then use the ON/OFF Switch.