

Chapter 12 Simon Says Game

Use four illuminated push buttons to build the game Simon Says.

Use a fifth button to start the game.

Use a sixth light to signal the game is done good.

Use a seventh light to signal the game is done bad.

Simon Says is a sequential game that plays a four-note tune. This lab has no sound so lights will have to do. After the start button is pushed, one light at a time is turned on. The player must mimic or push the button attached to the light. First one light is lit, then two, then three, etc., until a final number is attained. Assume 15 is the final number of steps in the game. The steps are repeated each time the game is played. The values are to be **stored in a table and re-used**.

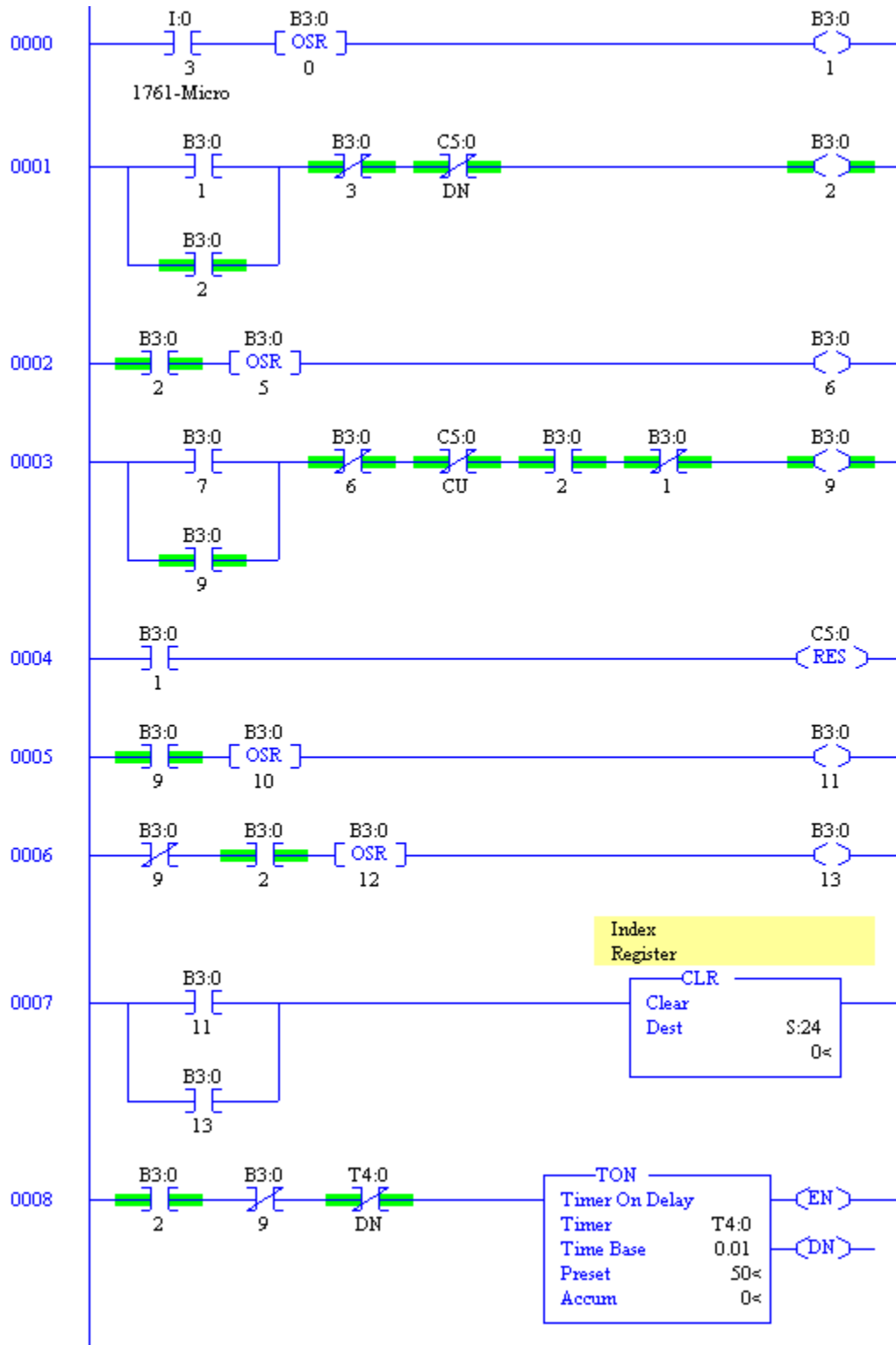
- A** Add a three-position switch to allow for easy, middle, hard ranges varying the number of steps from 10 to 15 to 20.
- B** Automatically rotate through 3 different sets of conditions for each range – easy, middle, and hard.
- C** Vary the speed that the lights are display as the player gets closer to the last step.
- D** Create a teach mode using a separate button to teach the game a sequence of steps.

Use the program starting on the next page as a suggested beginning point for your logic:

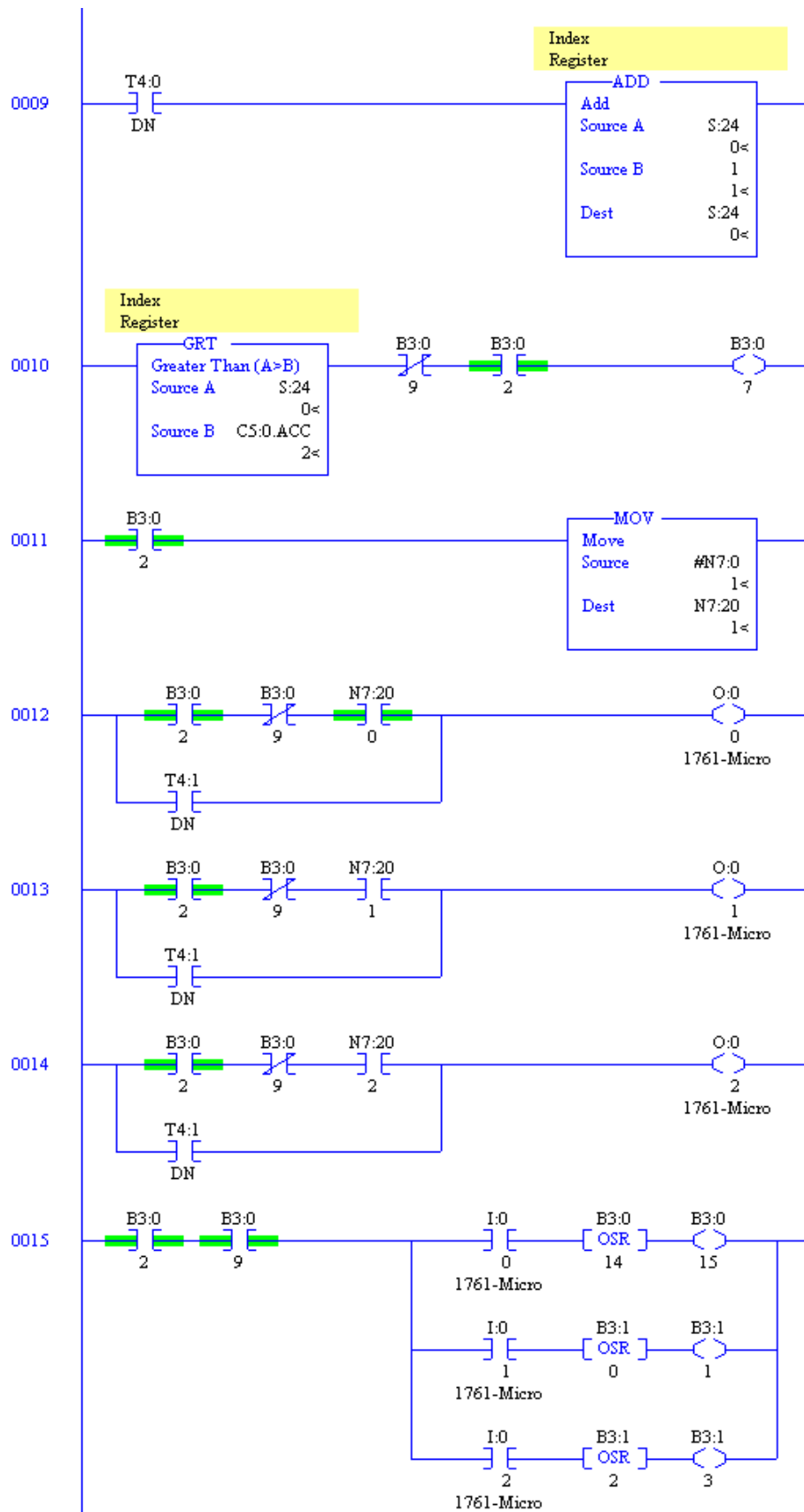


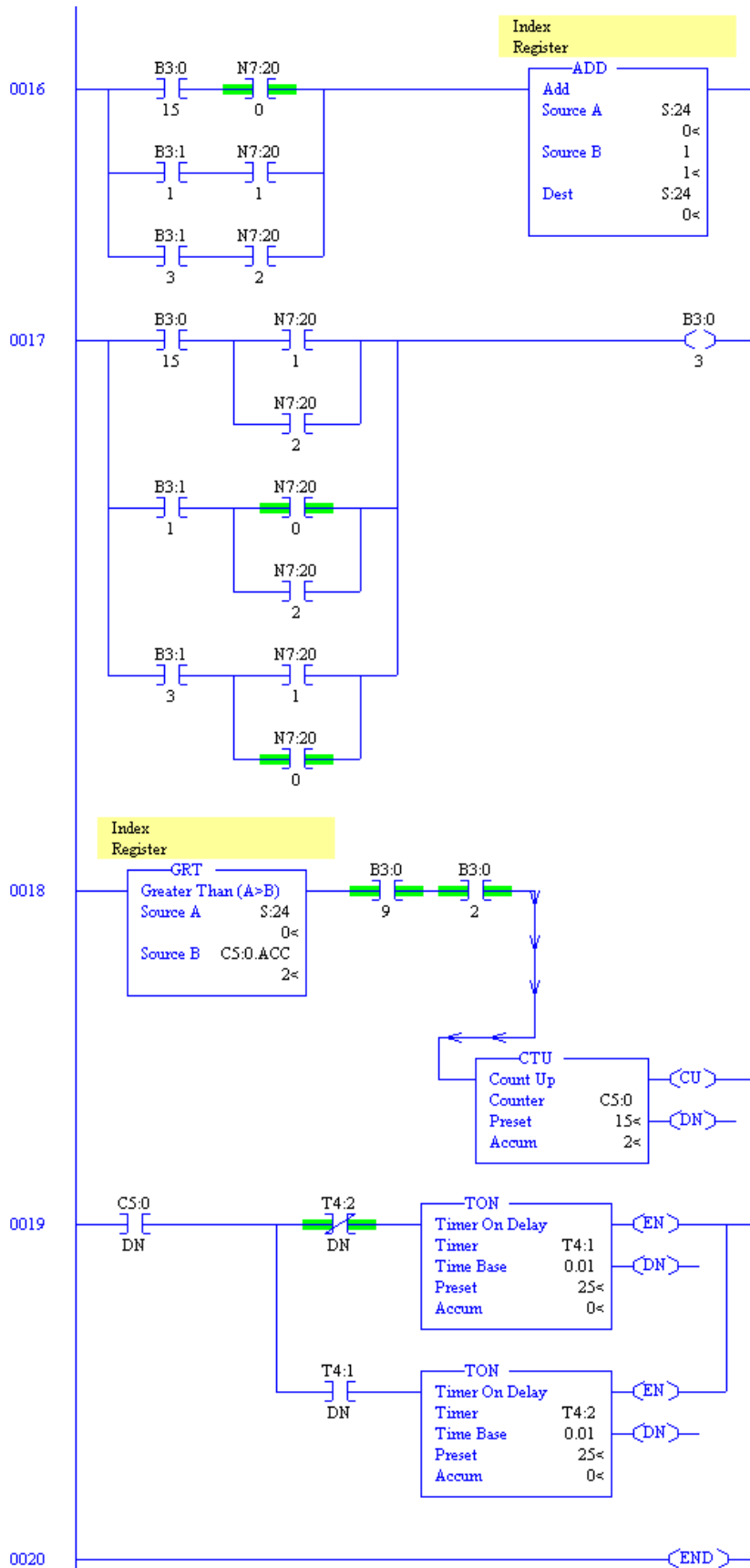
Simon Says Game

The program below is programmed for three, not four buttons and is programmed using indexed addressing from the SLC architecture PLCs from A-B.

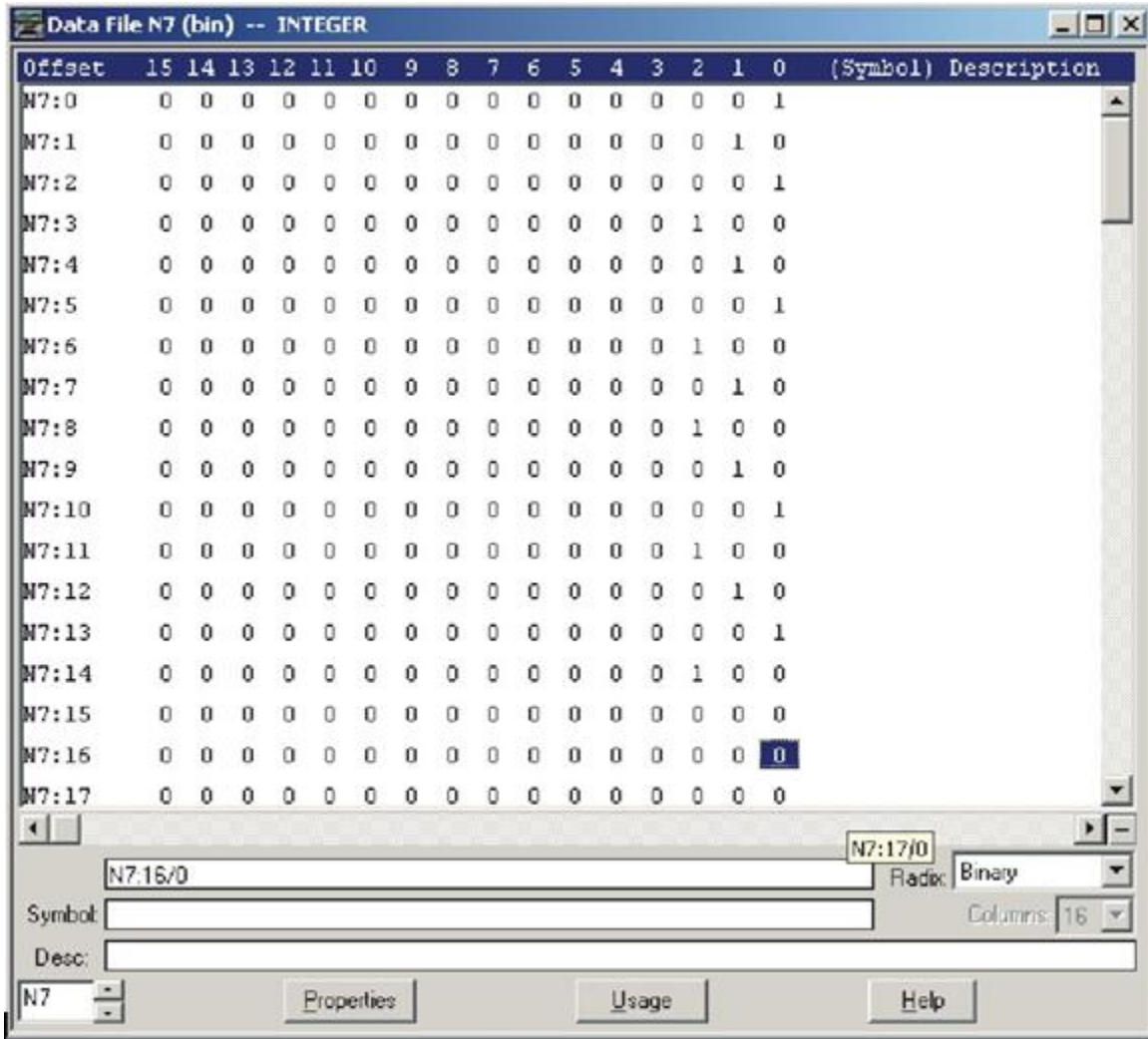


Simon Says Game Program in SLC





In addition to the program, a table must be filled in starting at N7:0 in the N7 file.



Simon Says Game Program in SLC – Array of Lights

The file above must be inserted for the game to work correctly. Set the Radix to *Binary* and enter each word from N7:0 to N7:14 with either a 1 in the bit 0, 1, 2 or 3 position. The entry may also be entered in the *integer* format using numbers 1, 2, 4, and 8. The game will be played using these entries.