Approach:

Mux is used which passes initial address using selector as PULSE which initially low and becomes permanently high after first edge of CLK. PULSE is generated using a JK flip flop whose inputs J and K are set to be 1 and 0. As soon as clock to flip flop goes high first time ,the output PULSE becomes permanently one.

The output of this mux is supplied to adder in which addend is decided by the state of clock and CO(which becomes one when right end is reaches). Whenever the clock is low addend is 0,otherwise it depends on CO. When CO is low addend is equal to 1 and becomes 2 when CO is one. The output of the adder is passed through ROM.

When clock is low ROM’s output is data of that node. This output data along with its address are directly supplied to comparators which compares them with previously stored data and address giving higher priority to data. If the output is required to be updated then the whole set of comparator will give high output. The Logic AND of CLK and this output is used as a clock to the D flip flops which stores minimum node data and their address. These flip flops are connected to logic probe to display the node data and its address.

A positive-triggered D-Flip flop is used in which clock(clk5) is supplied such that it has twice the frequency of CLK. This is done with the help of JK flip flop. Due to such arrangement this flip flop triggers at both positive and negative edge of CLK. Input to this flip flop is given through a mux. When clock is low mux outputs the address of the node and in case of high, outputs the address of the next node.

As the clock goes from 0 to 1, D flip flop stores the address of current node. This address is passed to the adder which increases the address by 1 or 2 depending on the direction of traversal (for right it is 1 and for left it is 2). This address when passed through ROM gives the address of next node. As the clock goes from 1 to 0 , D flip flop stores this next node’s address. This process repeats itself until address of next node from ROM becomes 255. As soon as this happens addend changes itself to 2 if it was 1. Due to this, output from ROM gives the address of the previous node. Now the process right traversal changes to left traversal. At the end of left traversal, when the address of previous node becomes 255 , now as the addend is already 2 it does not change and output of ROM remains 255. The arrangement is made such that when clock is high and output of ROM is 255 then clock remains high if output of ROM does not change. Due to this when the left end reaches output of ROM remains high as addend does not change and hence clock remains high forever.