ECE 111 Winter 2022  
HW3  
Hao Le A15547504

**Johnson Counter**

Code

Text

Description automatically generated

RTL netlist

Diagram

Description automatically generated

Resource usage

Graphical user interface

Description automatically generated with low confidence

Testbench simulation waveform

Graphical user interface, application

Description automatically generated

Upon review of the simulation waveform, the module works as a Johnson Counter by observations:

* At 45ns, when present goes low, this sets the shift register values to load\_cnt, which is 4’b0000.
* Then on every positive edge of clk, we see 1’s start to shift left from the right as a result of the initial 0’s being negated and wrapped around
* After the shift register is filled with 1’s, we see 0’s start to shift left from the right by the same logic
* In total, there are 8 different states count can have, which is 2N where N = the 4 bits of the shift register
* We also see that upon clear going low, count goes and stays at 4’b0000 even at the positive edge of clk.
* When present goes low again, this time with load\_cnt at a different value of 4’b1000, that get’s loaded into the shift register

**Universal Shift Register**

Code

Text

Description automatically generated

Text

Description automatically generated with medium confidence

RTL netlist

Diagram, schematic

Description automatically generated

Resource usage

Graphical user interface, application

Description automatically generated

Testbench simulation waveform

Graphical user interface

Description automatically generated

Checking shift modes:

* 0000 (PIPO) works because when din is 1001, so is dout.
* 0001 (SIPO-L) works because from an initial shift register state of 1001, upon positive edge of clk, and sin being high, we see a 1 inserted from the right, and shift\_reg turns to 0011. Next shift also has sin as high, so another 1 is insert to make shift\_reg 0111. However, next positive edge of clk has low sin, so now a 0 is inserted to make 1110. While these shifts are happening, dout reflects the values of shift\_reg in parallel
* 0010 (SIPO-R) works similarly to 0001, but now new values from sin are inserted from the right.

Graphical user interface

Description automatically generated

* 0011 (PISO-L) works because upon load being high, din with 1000 is loaded into register. Then upon positive edge of clk, it is shifted left, and a 0 bit is inserted to the rightmost register and the leftmost bit is moved to sout which is both 1’s for the next two clocks, but 0 for the third. Also, dout is 0 since this is a serial out shift mode.
* 0100 (PISO-R) works similar to 0011 but bits in shift\_register are shifted to the right, the 0 bit is inserted to the leftmost register, and the rightmost bit goes to sout.
* 0101 (SISO-L) works because upon each positive edge of clk, the value of sin is loaded into the rightmost register and the leftmost register of shift\_register is loaded into sout. Dout stays 0000 because this shift operation is serial.

Graphical user interface, timeline

Description automatically generated

* 0110 (SISO-R) works similar to 0110 but sin is loaded into the leftmost register, and the rightmost register’s value is fed into sout.
* For all shifting operations, when reset is high, sout and dout is always low which is expected