Lab 4 Report

1. SOF file is located in [James\_Park\_70315379\_Lab\_4/rtl/lab4/output\_files/rc4.sof]
2. Code compiles and runs on the De1. Functions correctly. All three loops run correctly, and able to find secret key. LED[0] is on if no key can be found. LED[1] is on if key is found. Simulations not done. Did not do challenge.
3. Signal Tap
   1. Loop 1

A picture containing text, screenshot, line, font

Description automatically generated

state[0] is wren signal. Held high until q output is same as data value, confirming write.

In loop 1, data is same as address value.

fsm state[0] signals loop 1 to start.

A red lines on a white background

Description automatically generated with low confidence

state[2] is finish signal. Asserted high when q output is FF when wren is high, confirming last write finished.

address increments when state[1] high on clk high.

* 1. Loop 2

state[5] high: store q value from read to s\_j.

state[3] high: update index\_j. Here it is changed to 03.

state[1] high: read s\_memory at index\_i.

state[4] high: read s\_memory at index\_j.

state[2] high: store q value from read to register s\_i.

Loop 2 begins when start\_reg is high

A picture containing text, screenshot, line, font

Description automatically generated

A screenshot of a computer

Description automatically generated with medium confidence

state[7] high: increment index\_i and loop. If the address, currently set to index\_i is FF, that was the last loop so assert state[8] which is finish signal.

state[6] high: change address to index\_i.

state[0] high again, writing s\_j to s[i], completing swap.

state[0] is wren signal. Here it will write s\_i to s[j].

* 1. Loop 3

A screenshot of a computer

Description automatically generated with medium confidence

state[9] is wren\_d. Write signal for decrypted output. If the xor of q (which is f) and q\_m (which is encrypted\_input[k]) is not a lower case alphabet character or space, invalid\_key signal is asserted high, ending loop 3. Otherwise, increment index\_k and loop. If invalid\_key is never asserted high. Loop ends when index\_i is equal to message length of 32. Invalid\_key signal will cause all three loops to assert low reset\_n.

state[8] high: read from address calculated by adding s[i] + s[j], which here is 0x75 + 0x8B = 0x100 or 0x00 because overflow.  
At the same time read from m\_memory the encrypted\_input at index\_k. Read value is output to q\_m.

state[7] high: Change address to index\_i, and write value s\_j to s[i].

state[0] high: write s\_i to s[j].  
Index\_j and s\_i is 0x75 right now.

state[6] high: store s[j] value to register s\_j.

state[4] high: update index\_j. New index\_j is (index\_j + s\_i), which is 0x75 here.

state[5] high: read from s[index\_j]. Value is 0x8B.

state[3] high: store s[i] value currently in q to register s\_i.

state[2] high: read s[index\_i].  
index\_i is 01 here. s[i] value is 0x75.

state[1] high: increment index\_i.