LFSR and Counter

lfsr-and-counters:~# Two important implementations of sequential circuits are as a pseudo number randomizer and as a step counter. Both of these applications can be constructed from D Flip Flops and appropriate logic gates. These provide the necessary building blocks to construct advanced logic construction of complex computational necessities.

GitHub repo: https://github.com/schwarzschlyle/electronics-and-instrumentation/tree/master/lfsr-and-counters

Current circuit implementation only uses Arduino as a clock. Hence, code slide sections are omitted due to redundancy

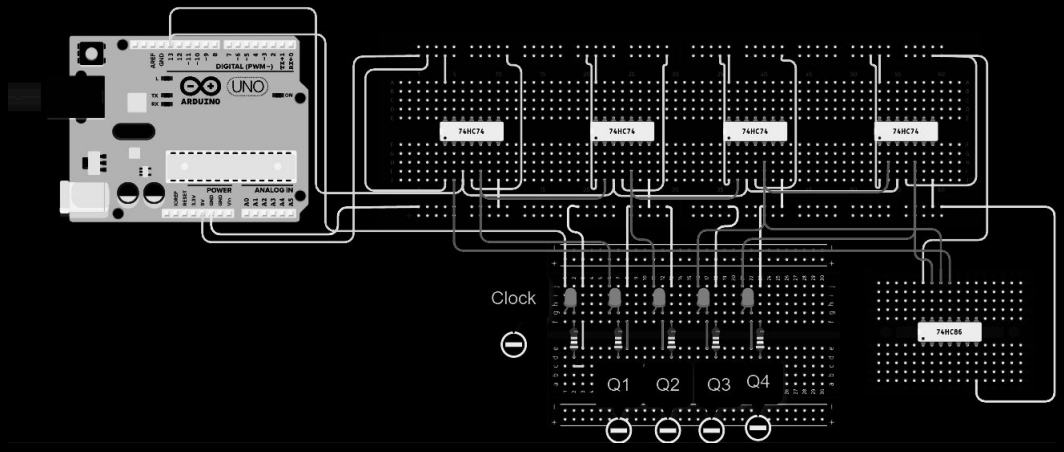
binary-states-convention:~# Throughout the exploration, black-filled LEDs correspond to a HIGH digital state while white-filled LEDs correspond to a LOW digital state

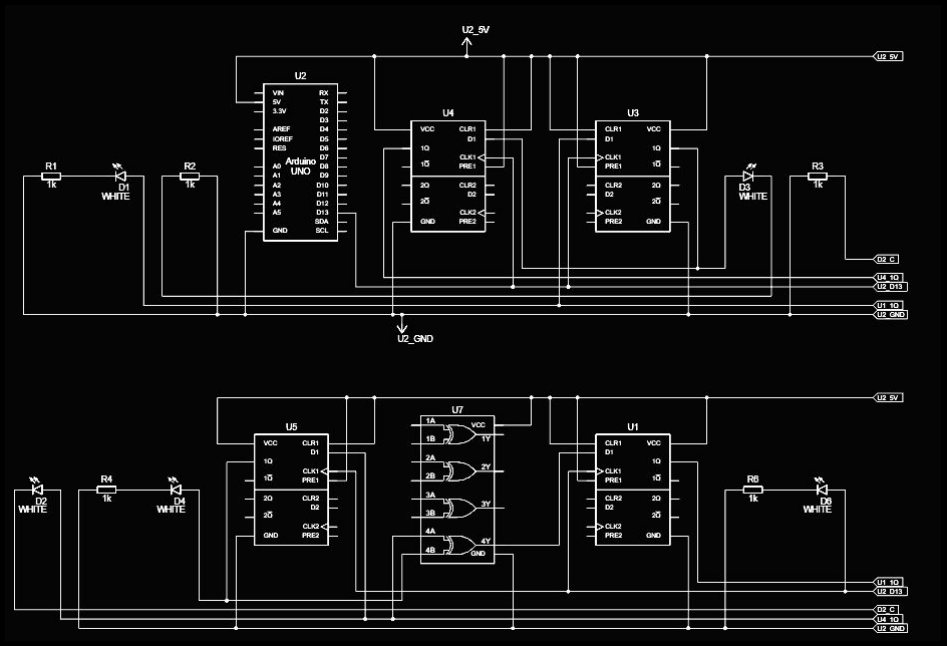
Geraldez-LK-2019-11336 184-WFU-FX-2 ~ (lfsr-and-counter)

4-bit-lfsr-XOR:~# Linear-feedback shift register is a variation of the shift register with the input being a linear function of the previous state; such circuit is commonly used to generate RNG. To explore such circuit, we aim to:

$ To construct a 4-bit linear feedback shift register using XOR gate,

$ To test and tabulate the logic output of the constructed 4-bit XOR LFSR





4-bit XOR LFSR logic cycle

1	0111
2	0011
3	0001
4	1000
5	0100
6	0010
7	1001
8	1100
9	0110
10	1011
11	0101
12	1010
13	1101
14	1110
15	1111

One common application of LFSR is as a pseudo random number generator. The circuit above shows 4-bit state cycle which appears to lack definite pattern. Starting from a seed state of (0111), the output cycles through 15 different states (shown on the truth table on the left), before going back to the initial state.

This may be used to re-generate the random cycle (hence, pseudo-random) seeded by the initial state. A different cycle can be generated by replacing the XOR gate with a XNOR gate as shown on the next slide.

Observe that the state (0000) is missing from the cycle.

4-bit XOR LFSR clock

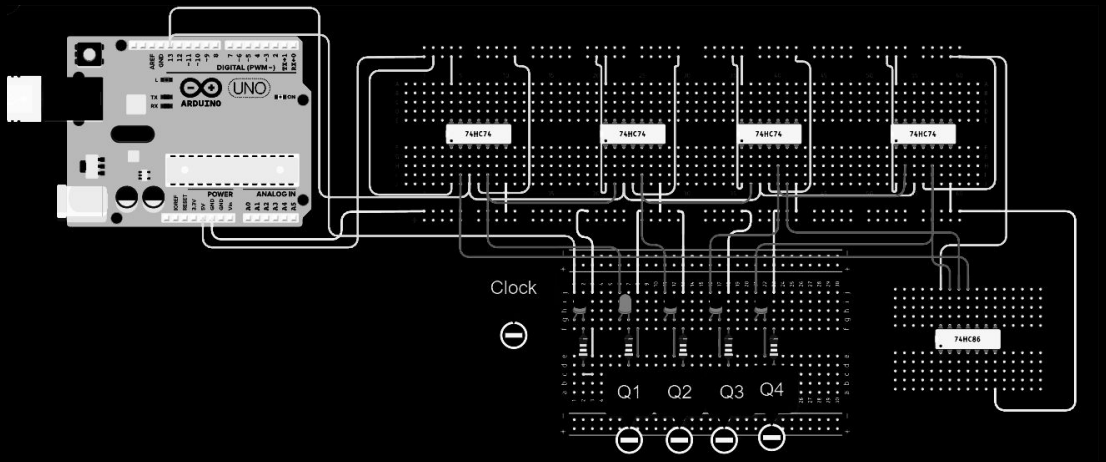
```
...
digitalWrite(clock, LOW);
delay(390);
digitalWrite(clock, HIGH);
delay(300);
...
```

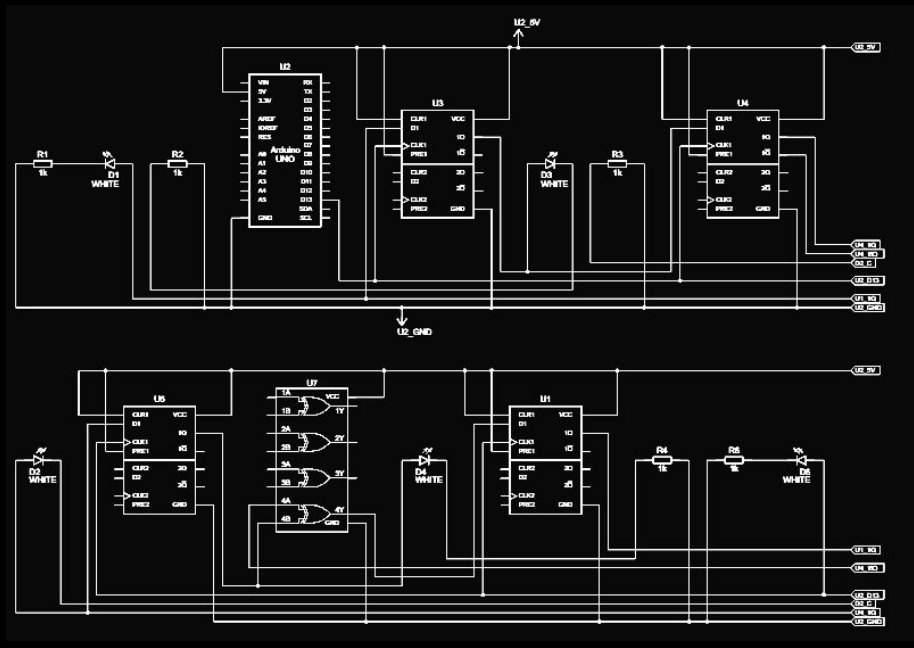
Geraldez-LK-2019-11336 184-WFU-FX-2 ~ (lfsr-and-counter)

4-bit-lfsr-XNOR:~# Linear-feedback shift register is a variation of the shift register with the input being a linear function of the previous state; such circuit is commonly used to generate RNG. To explore such circuit, we aim to:

$ To construct a 4-bit linear feedback shift register using XNOR gate,

$ To test and tabulate the logic output of the constructed 4-bit XNOR LFSR





4-bit XNOR LFSR logic cycle

```
1 0111
2 1011
3 1101
4 0110
5 0011
6 1001
7 0100
8 1010
9 0101
10 0010
11 0001
12 0000
13 1000
14 1100
15 1110
```

Replacing the previous circuit's XOR gate with a XNOR gate, the random cycle changes in sequence as can be observed on the left logic table.

This time, the state (0000) is present but lacks the state (1111) from the cycle. Hence, in both cases of XOR and XNOR LFSR, the cycle only contains 2^n-1 states for an n-bit LFSR.

4-bit XNOR LFSR clock

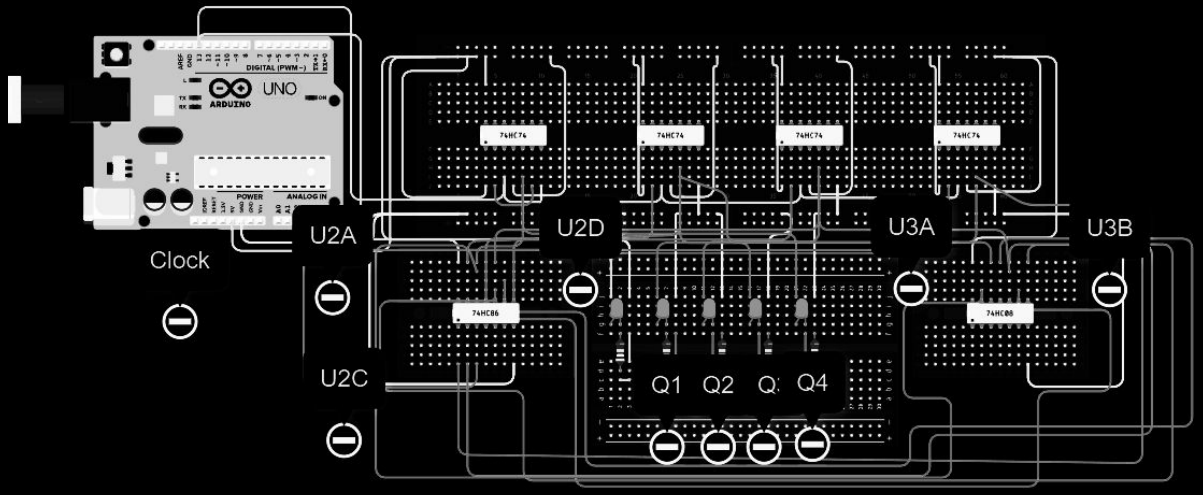
```
...
digitalWrite(clock, LOW);
delay(390);
digitalWrite(clock, HIGH);
delay(300);
...
```

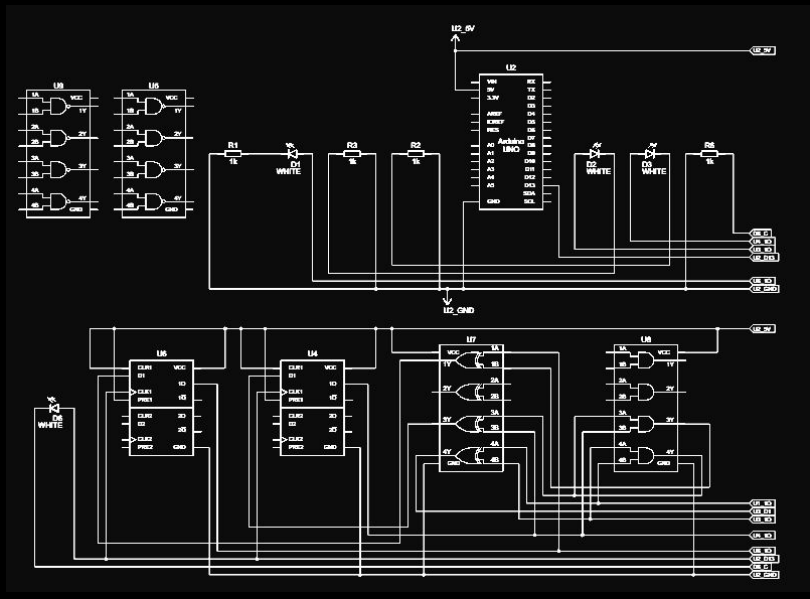
Geraldez-LK-2019-11336 184-WFU-FX-2 ~ (lfsr-and-counter)

4-bit-counter-full-count:~# A digital sequential circuit can also be designed in such a way that the an n-bit output cycles through increasing binary value from 0 to 2^n-1. To explore such circuit, we aim to:

$ To construct a digital counter sequential circuit,

$ To test and infer the timing diagram of the constructed counter.

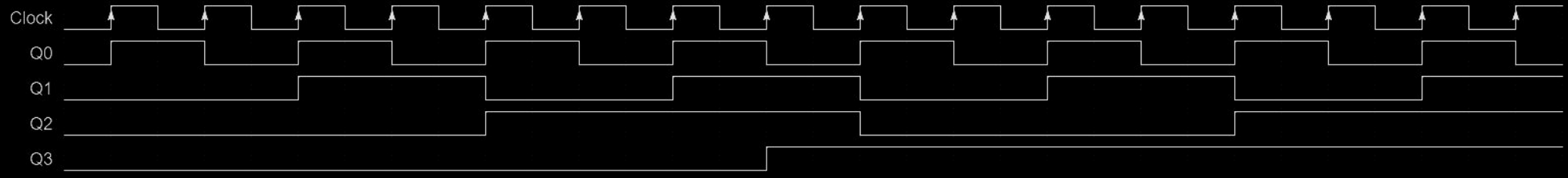




The logic applications of the counter sequential circuit is straightforward. Starting from an initial state of (0000), the counter produces output of increasing binary value: (0000) > (0001) > (0010) > …. and so on until (1111) from when the circuit resets back to (0000). Observing from the timing diagram below, one can notice that the logic output is the familiar frequency divider that has been previously implemented.

4-bit counter clock

```
...
digitalWrite(clock, LOW);
delay(390);
digitalWrite(clock, HIGH);
delay(300);
...
```

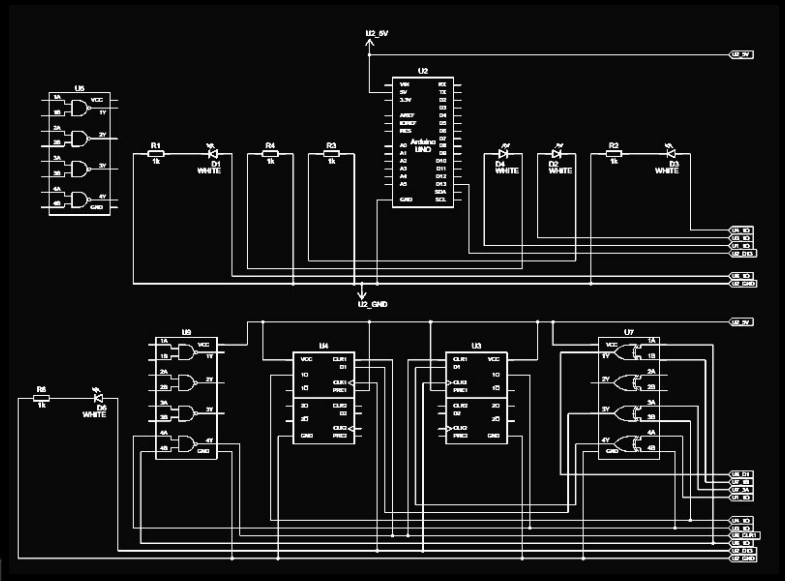


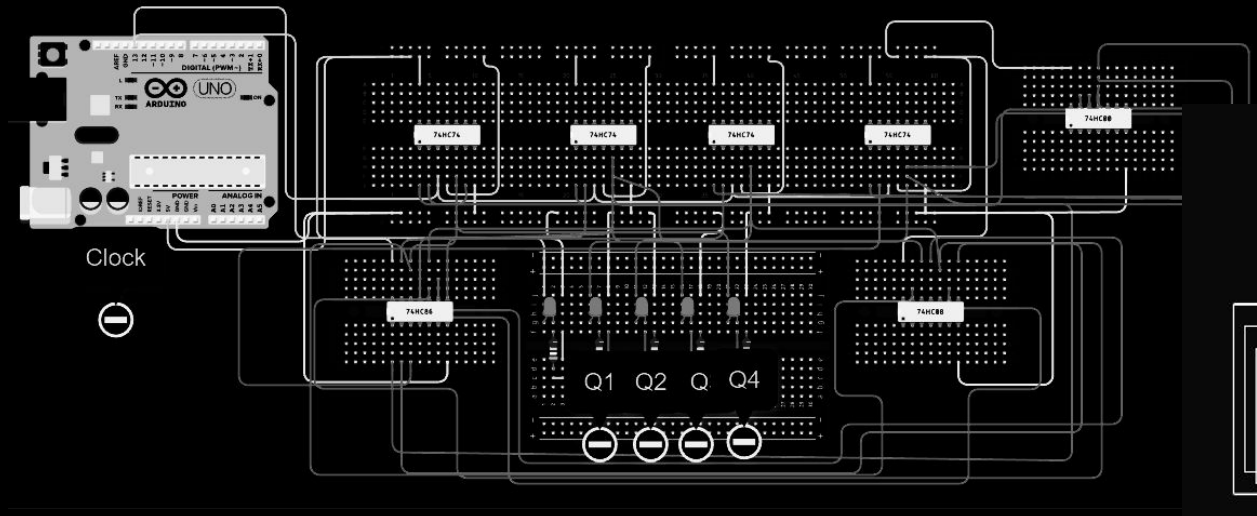
Geraldez-LK-2019-11336 184-WFU-FX-2 ~ (lfsr-and-counter)

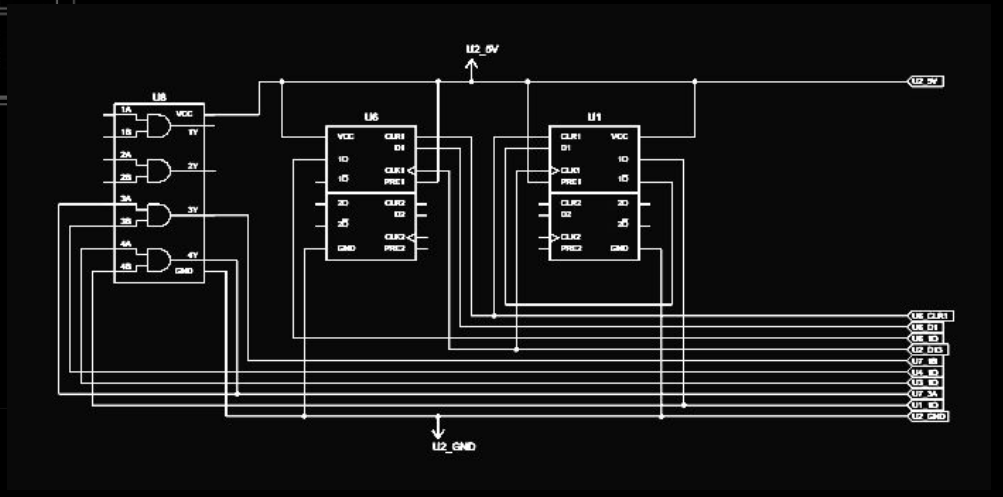
4-bit-counter-partial-count:~# One can prematurely halt the complete cycle of the digital counter resetting it after reaching a certain binary state. To see this, we aim to:

$ To construct a digital counter sequential circuit,

$ To implement a resetting mechanism for the constructed counter after state (1001).







To implement the resetting mechanism, the active low reset pins of the dual D flip flops are being fed by the output of a NAND gate. The two input legs of the NAND gate are being fed by the highest bit (decimal value: 8) and the second the the lowest bit (decimal value: 2). Hence, from state (0000) to state (1001), the NAND gate produces a high digital state. Upon reaching (1010), the both legs are in HIGH state producing an digital LOW state output. Feeding a LOW state to the active low reset pins resets the entire digital counter back to (0000). This completes the resetting mechanism after reaching (1001) state (decimal value: 9)

4-bit counter clock

```
...
digitalWrite(clock, LOW);
delay(390);
digitalWrite(clock, HIGH);
delay(300);
...
```