Stepper Motor Driver
Digital Circuit Design and Implementation Report

Daniel Eisen
Partner: Nalin Aswan

Introduction.

This is a design for a stepper motor driver, using discrete cmos ICs to operate a brushless Solen Stepper motor. This design required and met the following specifications:

1. Have an internal driving clock
2. Generate and specific count sequence to drive the motor coil
3. Sequential and direction bit to dictate direction of rotation
4. Sequential enable bit the pause/resume rotation
5. 4 bit output to stepper motor PCB (supplied)

The following design/implementation requires the following [or functionally equivalent] IC chips and components:

- 1×74HCT112 Dual JK flip-flop with set and reset; negative-edge trigger
- 1×74HCT08 Quad 2-input AND gate
- 2×555 Timer, Monostable or Astable
- 10uF and 10nF capacitor
- 47k and 10k ohm resistors

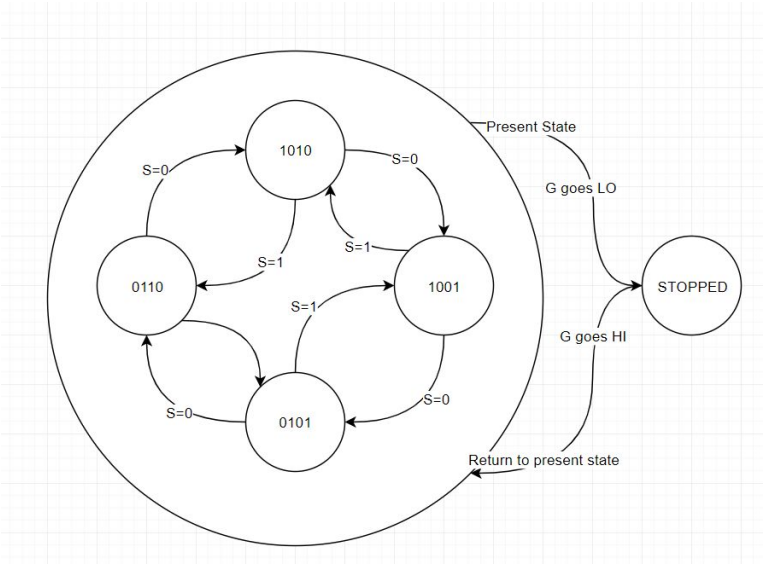
Note: the above resistor/cap values are to approximate 1hz, to manipulate frequency adjust the 10uF cap and 47k resistor according to the formulae stated on the diagram belong.

Following notation: S → direction bit, $W_{1,2,3,4}$ → counter output bits, G → enable bit (active HI)

Design Description.

State Diagram:

Shows forward and backward rotation states and stopped state (enable LO)



Output bit progression

S = 0	W4	W3	W2	W1	S = 1	W4	W3	W2	W1
	1	0	1	0		0	1	1	0
	1	0	0	1		0	1	0	1
	0	1	0	1		1	0	0	1
	0	1	1	0		1	0	1	0

Simplification: !W4 = W3 and !W2 = W1

So using 2 JK's (A, B) : W1, W2 → A, !A ; W3, W4 → B, !B

Transition Table

Present			Next			
S	B	A	B	A	JK_B	JK_A
0	0	0	0	1	0X	1X
0	0	1	1	1	1X	X0
0	1	0	0	0	X1	0X
0	1	1	1	0	X0	X1
1	0	0	1	0	1X	0X
1	0	1	0	0	0X	X1
1	1	0	1	1	X0	1X
1	1	1	0	1	X1	X0

K-Maps

J_B

!S	0	1	X	X
S	1	0	X	X
	!B!A	!BA	BA	B!A

K_B

!S	X	X	0	1
S	X	X	1	0
	!B!A	!BA	BA	B!A

J_A

!S	1	X	X	0
S	0	X	X	1
	!B!A	!BA	BA	B!A

K_A

!S	X	0	1	X
S	X	1	0	X
	!B!A	!BA	BA	B!A

$J_B = !SA + S!A$ $K_B = !S!A + SA$ $J_A = !S!B + SB$ $K_A = S!B + !SB$

If the Js and Ks are wired to be the same, ie FFs toggle at a specific times and given !AB+A!B is XOR : ⊕

The final simplification is:

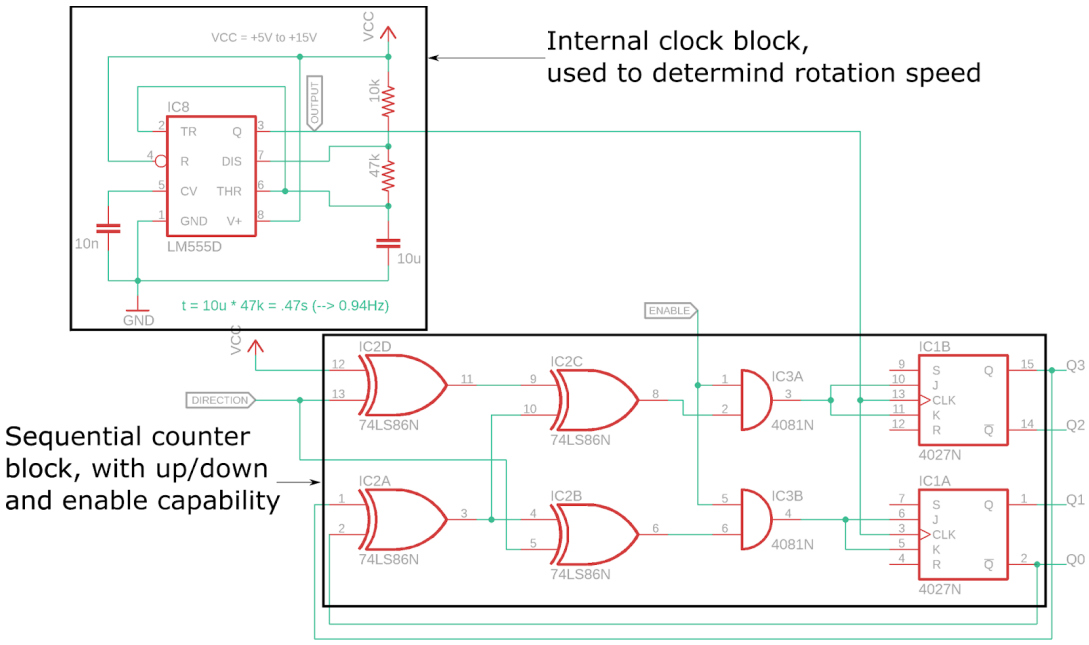
$\mathbf{JK_B} = (1 \oplus S) \oplus (B \oplus !A)$ $\mathbf{JK_A} = S \oplus (B \oplus !A)$

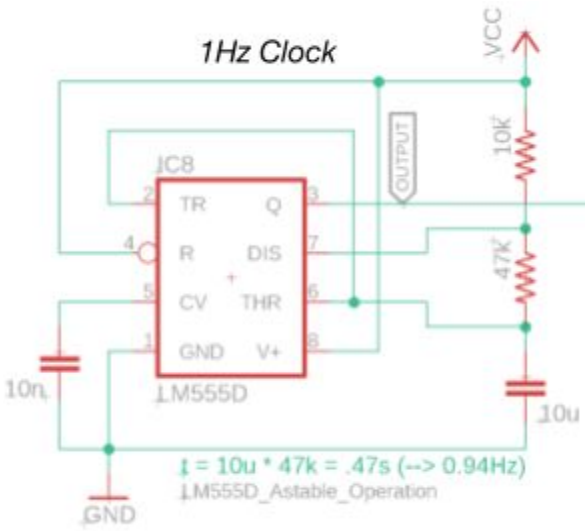
The only addition from this is the implement an an enable bit **G** (active HI) to set FF into no change state, ie J and K both LO.

We accomplished this by anding the the enable bit the inputs JK_B and JK_A

The resulting functionality is that of the specified bit progression above, ie both FF A, B hold o/p for 2 clk pulse then toggle.

Fully Implemented Circuit Diagram





Requires: 555 Timer; Monostable/Astable, 2 resistors (one more the 3x smaller), up to 2 caps (10nF may not be needed)

To supply a 1Hz CLK signal to the main counter block, we chose to construct a standard 555 timer in a Astable operation configuration. With a VCC supply voltage, and when grounded this circuit will continuously oscillate between HI and LOW output (at Q); thus supplying a squared clock signal to the rest of the circuit.
In short this operates by utilising the charge discharge cycle of the larger capacitor to periodically grounding the reset (active LO).

To determine the frequency of this oscillation use the formulae: f = 2(Large R * Large C), in our case 2 (10u * 47k) = 0.94Hz, this gave a T = 1/.94 → so our timer kept a 1.06 second resolution.
To modify rotation speed, change the components of this formula, maybe even a variable resistor.

Summary and Conclusions.

In summary the final circuit is a functional and relatively flexible, working exactly as specified and designed; direction is controllable and can be altered during operation, enable stops and starts the rotation and speed of rotation can be changed by altering the time constant associated with the astable 555 block, either by swapping out the resistor value or by including a variable resistor. Alternatively many variation of the counter block can be constructed from the transition table, with combinations of MUXes, D FF and JKs. We chose the above design to minimise IC use, as counter (minus the enable bit) was 2 ICs.

In conclusion I personally found the practical application of driving an external system very satisfying, with tangible results being very satisfying.