ME 104

Sensors and Actuators

Fall 2003

Laboratory #8

Stepper Motors

Department of Mechanical and Environmental Engineering

University of California, Santa Barbara

Fall 2003 Revision

Introduction

In this Laboratory, you will write a *LabVIEW* VI for generating digital TTL signals that can be used to provide the stepping sequence for a four-phase unipolar stepper motor. You will use this VI to generate the stepping sequence for a unipolar stepper motor. You will then use a function generator and an analog drive circuit to control the rate of rotation of a unipolar stepper motor in full-step mode. As an extra credit exercise, you will build *LabVIEW* VI's for controlling the rotation of your stepper motor in full-step mode.

The stepper motor you will use in this Laboratory is a four-phase unipolar stepper motor produced by *Jameco Electronics* (Part #166705). The motor requires a DC **Supply Voltage** of 12 V and a **Current** of 140 mA. The motor has a **Step Angle** of 3.8°, **Phase Resistance** of 84 Ω, and **Phase Inductance** of 16 mH. The phase and field coil schematic for this stepper motor is shown in Figure 1.

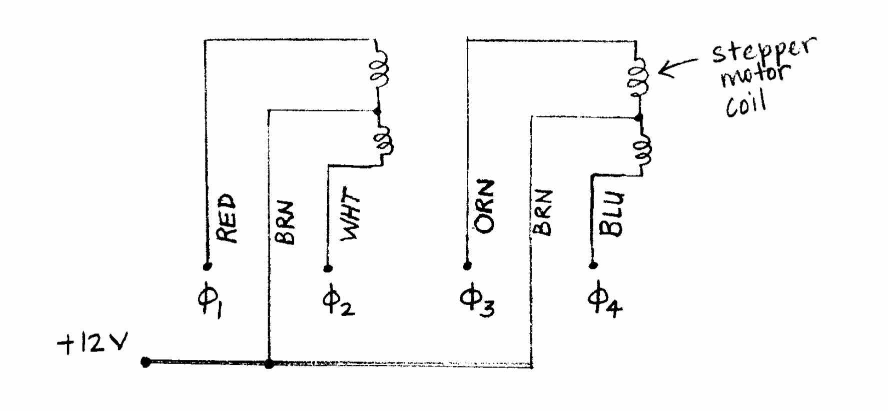

Figure 1: Jameco stepper motor phase and field coil schematic. (Colors refer to lead wires directly attached to motor casing).

Background Reading

Please read the following material prior to this lab:

1. Histand and Alciatore, **Introduction to Mechatronics**, Section **6.3** and Sections **10.6.**
2. **Data Sheet**, *IRLIZ34N HEXFET Power MOSFET*, International Rectifier. Available online at www.irf.com/product-info/datasheets/data/irliz34n.pdf.
3. **Data Sheet**, *74HC191 Synchronous 4-Bit Binary Up/Down Counters with Mode Control*, National Semiconductor Corporation. Available online at http://www.daqchina.net/daqchina/chips/74HC191.pdf

4. **Data Sheet**, *74HC86 Quad 2-Input Exclusive OR Gate*, National Semiconductor Corporation. Available online at http://www.eng.yale.edu/ee-labs/morse/compo/datasheets/MM74HC86.pdf.

Experiment 1: Build a LabVIEW VI for Simultaneous Generation of Four Digital Signals

In this experiment, you will write a *LabVIEW* VI that simultaneously generates four digital TTL signals that can be used to provide the stepping sequence for a four-phase unipolar stepper motor.

Recall that the PCI-6024E DAQ Board has eight digital input/output lines, all of which are **TTL** by nature. Of these eight lines, the first four lines (`DIO0, DIO1, DIO2,` and `DIO3`) have been configured to **Read** (acquire) TTL signals, while the last four lines (`DIO4, DIO5, DIO6,` and `DIO7`) have been configured to **Write** (generate) TTL signals.

1. Launch *LabVIEW* and build the VI shown in Figure 2. Each of the four buttons on the front panel is a **Square Light** that turns bright green when the button is in the ON position and turns dark green when the button is in the OFF position. To display the Boolean ON (or OFF) text values for each square light, right-click on the square light and select **Show>Boolean Text**. You can move the Boolean ON (or OFF) text values using the *Positioning* tool.

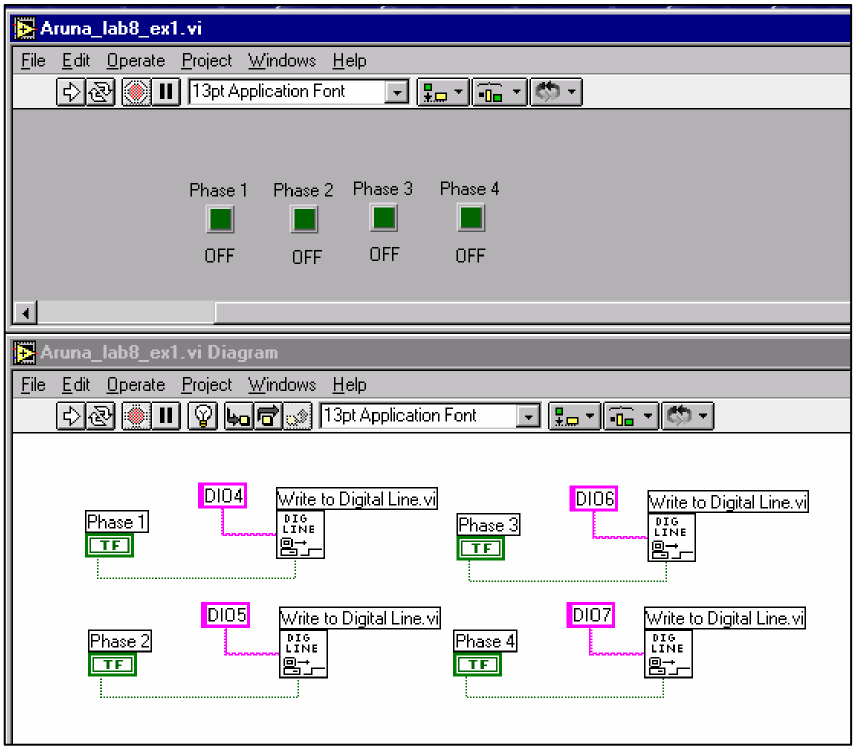

Figure 2. Front panel and block diagram for `yourname_lab8_ex1.vi`.

2. Choose an appropriate[*] **Digital Ground** (DGND) pin on your CB-68LP connector block and provide (define) ground to that pin using the black (-) terminal of the "5 V FIXED 3 A" output from your Tektronix PS280 DC Power Supply[†]. You **do not** need to turn on the power supply.
3. For viewing purposes, connect **Digital I/O Channel 4** (DIO4) and **Digital I/O Channel 5** (DIO5) to your oscilloscope.
4. Continuously run your VI by clicking the **Run Continuously** button. Press the different Phase buttons to toggle them between ON (high) and OFF (low) states. Using your oscilloscope display, verify that the outputs from **Digital I/O Channel 4** (DIO4) and **Digital I/O Channel 5** (DIO5) are what you expect them to be.
5. Now connect **Digital I/O Channel 6** (DIO6) and **Digital I/O Channel 7** (DIO7) to your oscilloscope and verify that the outputs from those two channels are what you expect them to be (i.e., repeat Step 3 above for those two channels).
6. Stop running the VI by clicking the **Abort Execution** (stop) button.
7. Save this VI as **yourname_lab8_ex1.vi**.
8. **Ask your TA to check that you have successfully completed Experiment #1.**

Experiment 2: Increment the Position of a Unipolar Stepper Motor Using an Analog Current Amplification Circuit

In this experiment, you will use the *LabVIEW* VI you built in Experiment #1 to produce the stepping sequence for the *Jameco* four-phase unipolar stepper motor. The current from the Digital I/O Channels on the DAQ Board is not sufficient to drive the stepper motor. Therefore, before being sent into the stepper motor, the current from each of those digital channels must to be amplified using a **Power MOSFET**. A power MOSFET is used to interface a low output current device such as a data acquisition board or computer port to another device (such as a stepper motor) that requires larger currents. The power MOSFET you will use in this laboratory is an **IRLIZ34N HEXFET Power MOSFET** manufactured by *International Rectifier* (see Figure 3).

Your breadboard has been pre-wired such that **red wire** indicates **+5V** voltage supply, **yellow wire** indicates **+12V** voltage supply, and **black wire** indicates **0V (GND)**.

[*] Choose a DGND pin that is reasonably close to the four Digital I/O lines (pins) you are using.

[†] Since the ground terminal from the DC Power Supply has already been connected to your electronic breadboard, you can use the ground connection from that breadboard to define ground.

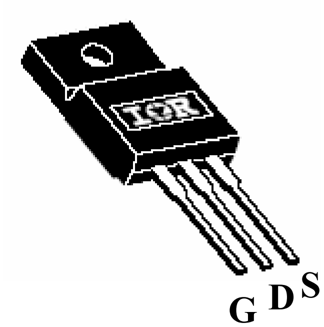

Figure 3. IRLIZ34N Power MOSFET: G = Gate, D = Drain, S = Source

1. Turn on your Tektronix PS280 DC Power Supply and adjust the appropriate voltage knob until the positive voltage is approximately `12.0` V.
2. Turn off the DC Power Supply.
3. Build and connect the circuits shown in Figure 1 and Figure 4. Your **gate phase** ($G\phi$) signals will be generated by the corresponding **Digital I/O Channels** on your CB-68LP connector block. As specified by your VI, obtain $G\phi_1$ from `DIO4`, obtain $G\phi_2$ from `DIO5`, obtain $G\phi_3$ from `DIO6`, and obtain $G\phi_4$ from `DIO7`.

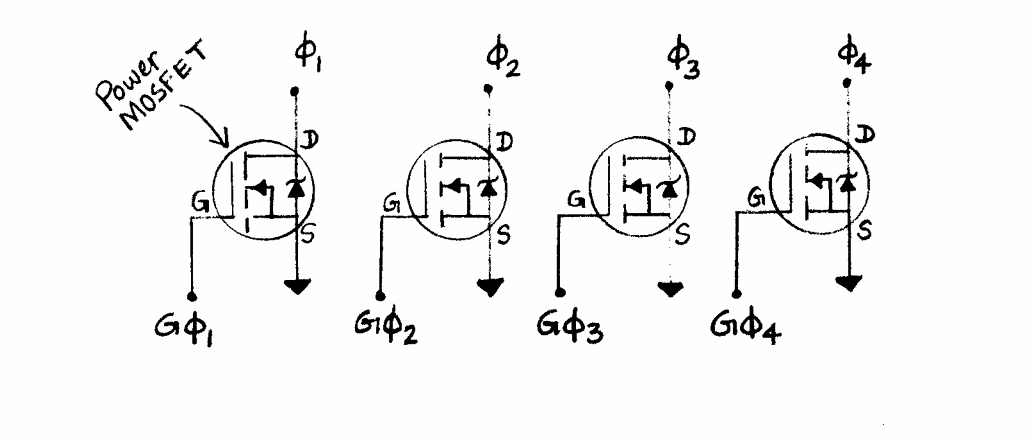

Figure 4: Current amplification circuit

Your goal is to verify that the phase sequence shown in Table 1[*] does, in fact, induce 3.8° full-steps (increments) in shaft angular position.

Table 1: Unipolar full-step phase sequence

Step	ϕ_1	ϕ_2	ϕ_3	ϕ_4
1	ON	OFF	ON	OFF
2	ON	OFF	OFF	ON
3	OFF	ON	OFF	ON
4	OFF	ON	ON	OFF

▲ CCW

▼ CW

4. Open `yourname_lab8_ex1.vi.`
5. Press the `Phase` buttons so that they specify **Step 1** (from Table 1) and then run your VI (once) by clicking the **Run** button. Your stepper motor shaft may move initially, but it will quickly come to a stop.
6. Now set (specify) **Step 2** (from Table 1) and then run your VI (once) by clicking the **Run** button. The position of your stepper motor shaft should full-step (increment) once by 3.8° in the clockwise (CW) direction[†].
7. Repeat the step sequence for **Steps 3** and **4** as indicated in Table 1 and verify that each step results in a CW full-step (increment) of 3.8°.
8. Now reverse the step sequence and verify that each step results in a CCW (counter-clockwise) full-step (increment) of 3.8°.

You can also verify that the phase sequence shown in Table 2[‡] does, in fact, induce 1.9° half-steps (increments) in shaft angular position. Notice that, to jump between any two adjacent steps shown in Table 2, you would need to change the state of only one phase at a time. Therefore, you can conveniently verify the half-step sequence by running the VI continuously.

Table 2. Unipolar half-step phase sequence

Step	ϕ_1	ϕ_2	ϕ_3	ϕ_4

[*] This is a direct copy of Table 10.1 from the Mechatronics textbook.

[†] Although it is not practical for the human eye to exactly verify a small angle such as 3.8°, you should make sure that the step (increment) you observe is within the ballpark.

[‡] This is a direct copy of Table 10.2 from the Mechatronics textbook.

1	ON	OFF	ON	OFF
1.5	ON	OFF	OFF	OFF
2	ON	OFF	OFF	ON
2.5	OFF	OFF	OFF	ON
3	OFF	ON	OFF	ON
3.5	OFF	ON	OFF	OFF
4	OFF	ON	ON	OFF
4.5	OFF	OFF	ON	OFF

9. Press the `Phase` buttons so that they specify **Step 1** (from Table 2).
10. Continuously run your VI by clicking the **Run Continuously** button. Your stepper motor shaft may move initially, but it will quickly come to a stop.
11. Now set (specify) **Step 1.5** (from Table 2). Verify that the position of your stepper motor shaft half-steps (increments) once by 1.9° in the clockwise (CW) direction.
12. Continue this process by specifying the half-step sequence shown in Table 2. Verify that each half-step results in a CW half-step (increment) of 1.9°.
13. Now reverse the half-step sequence shown in Table 2 and verify that each half-step results in a CCW (counter-clockwise) half-step (increment) of 1.9°.
14. Stop running the VI by clicking the **Abort Execution** (stop) button.
15. **Ask your TA to check that you have successfully completed Experiment #2.**

Experiment 3: Observe the Output of a Binary Counter Chip

In this experiment, you will use a function generator and an oscilloscope to verify the behavior of the four output bits of a binary counter chip. The binary counter chip you will use in this Laboratory is the **74HC191 Synchronous 4-Bit Binary Up/Down Counter** manufactured by *National Semiconductor* Corporation. The pin diagram for this counter chip is shown in Figure 5.

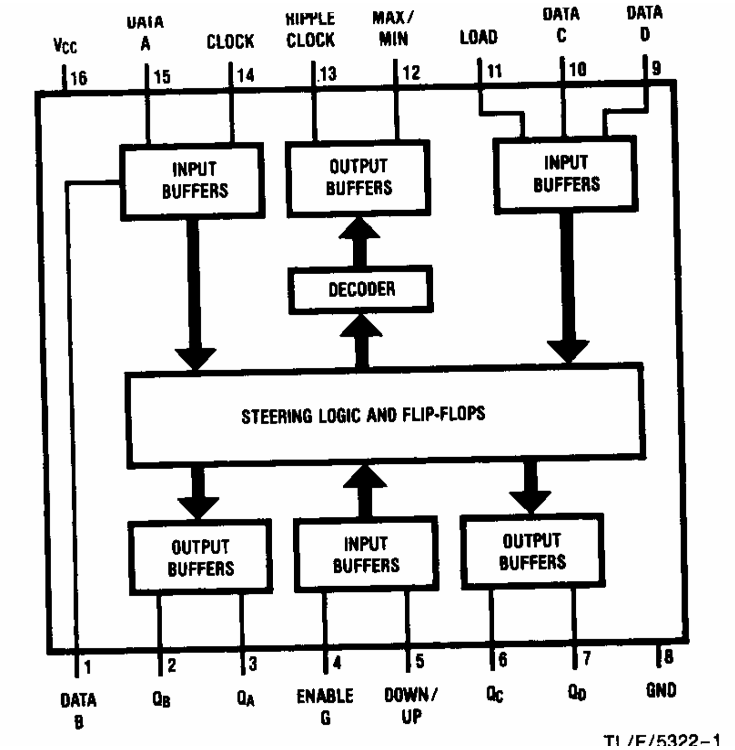

Figure 5: Connection diagram for 74HC191 binary counter chip

1. Make sure that your CB-68LP connector block is no longer connected to your power MOSFET's. In particular, **disconnect** the CB-68LP connector block from Pin 1 of each of the four power MOSFET's. (Recall that you made these connections during Experiment #2).
1. Connect the **MAIN OUT** terminal on the function generator to **Channel 1** on your oscilloscope. Set the vertical scale on your oscilloscope to 5.00 volts/division.
2. Find the **FUNCTION** selection buttons on your function generator. Select (press down) the square wave function (⊓⊔) button. Make sure that none of the other buttons on that row are pressed down.
3. On your function generator, select the "**0-20Vp-p** OPEN CIRCUIT" setting by pressing the button until it is in the "up" position. This will extend your voltage output range to 0-20 volts, peak-to-peak.
4. Set the amplitude of your square wave to **2.5V (5V peak-to-peak).**
5. Locate the knob to adjust the DC offset (next to the Amplitude knob). Gently pull the knob outward to allow adjustment of the DC offset. Adjust the DC offset such that **your square wave is entirely above 0V** (ground).
6. Set the frequency to **100 Hz**.

7. Verify the properties of your square wave using the oscilloscope display. The period of your square wave should be T = 10 milliseconds. For best viewing, set the horizontal scale on your oscilloscope to 10 ms/division.
8. Using an electronic breadboard, build and connect the circuits shown in Figures 1, 4, and 6. For the purposes of this Laboratory, the "**direction switch**" shown in Figure 6 will consist of a black wire. You can use this direction switch to make the counter count up or count down.

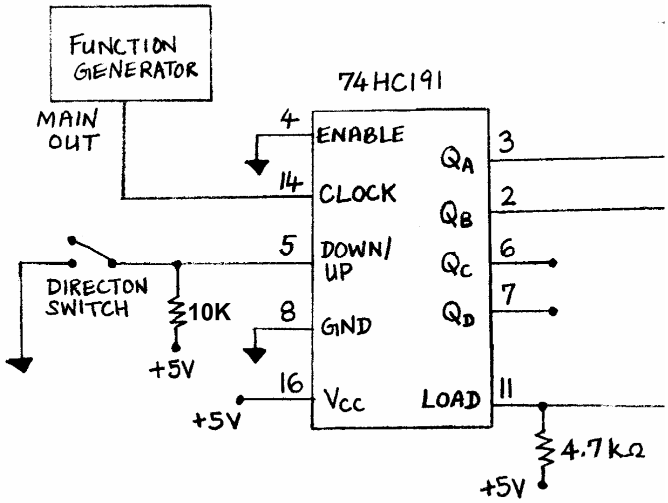

Figure 6: Binary up/down counter circuit

9. Close the direction switch. When you do this, Pin 5 on your counter chip is connected to 0 V (low), so your counter should count up (increment).[*]
10. Connect the least significant bit $\mathbf{Q_A}$ (pin 3) of the counter chip output to **Channel 2** of your oscilloscope. Verify that the output Q_A is a digital pulse train with period 2T, where T is the period of your clock input (from the function generator). Make a sketch.
11. Now connect the second least significant bit $\mathbf{Q_B}$ (pin 2) of the counter chip output to **Channel 2** of your oscilloscope. (Remove the connection to Q_A). Verify that the output Q_B is a digital pulse train with period 4T. Make a sketch.
12. Connect the third least significant bit $\mathbf{Q_C}$ (pin 6) of the counter chip output to **Channel 2** of your oscilloscope. (Remove the connection to Q_B). Verify that the output Q_C is a digital pulse train with period 8T. Make a sketch.

[*] See counter data sheet for explanation.

13. Connect the most significant bit **Q_D** (pin 7) of the counter chip output to **Channel 2** of your oscilloscope. (Remove the connection to Q_C). Verify that the output Q_D is a digital pulse train with period 16T. Make a sketch.

14. **Ask your TA to check that you have successfully completed Experiment #3.**

Experiment 4: Control the Angular Velocity of a Unipolar Stepper Motor Using an Analog Drive Circuit

In this experiment, you will use a function generator and an analog drive circuit to control the rate of rotation of a unipolar stepper motor in full-step mode. Your analog drive circuit will consist of a binary counter chip and an exclusive OR (XOR) gate chip. For this laboratory, the XOR gate chip we will use is the **74HC86 Quad 2-Input Exclusive OR Gate** manufactured *by National Semiconductor* Corporation. This XOR gate chip has four XOR gates (Figure 7), but you will only need to use three of them for this Laboratory.

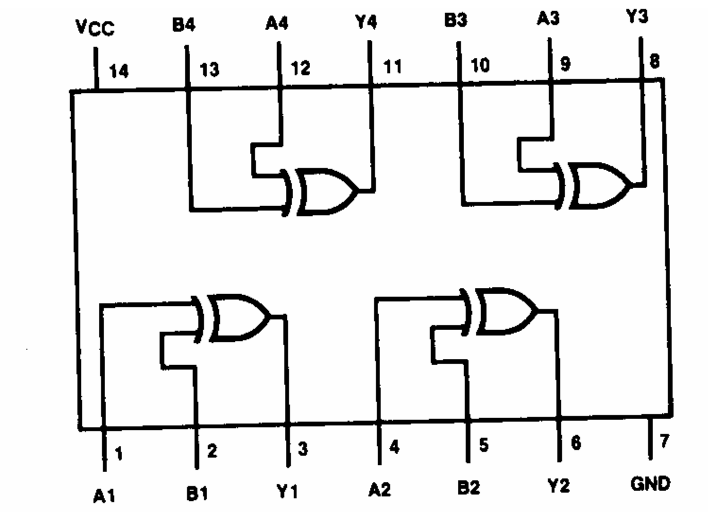

Figure 7: Connection diagram for 74HC86 Exclusive OR chip

1. Set your function generator to output a square wave that is entirely above 0V with an amplitude of **5V peak-to-peak, DC offset of 2.5 V,** and **frequency 1 Hz**.

2. Build[*] and connect the circuits shown in Figure 8. Don't forget to provide power (V_{CC} = +5V) and ground (GND) to your XOR gate chip.

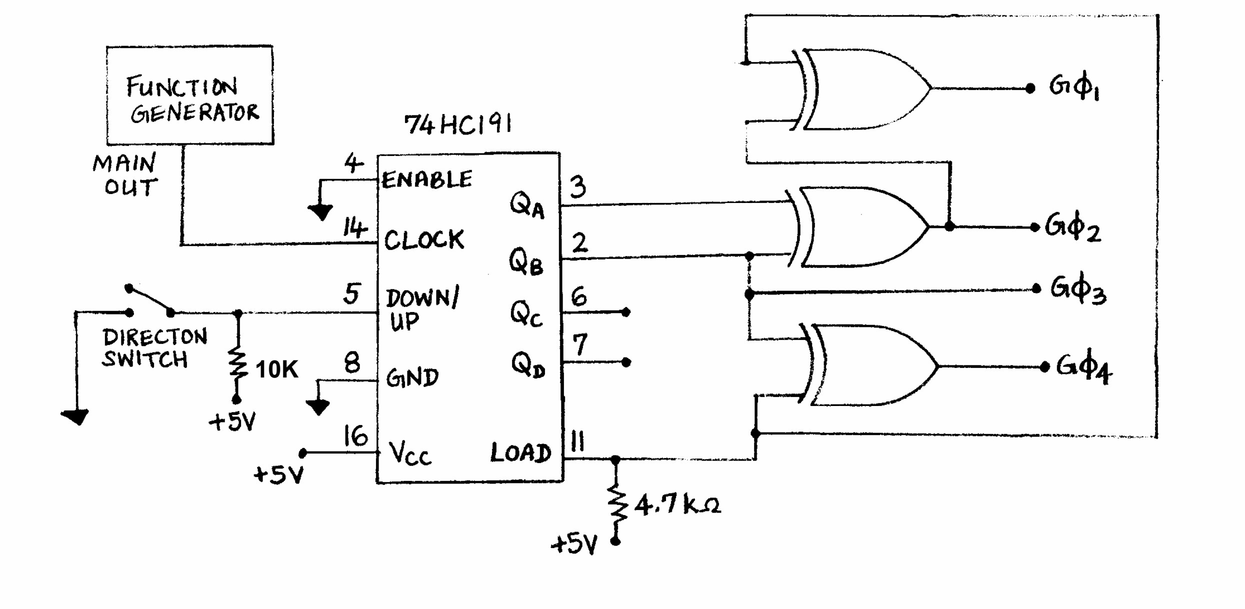

Figure 8: Unipolar stepper motor full-step drive circuit

3. Close the direction switch.
4. Vary the frequency of your clock signal (square wave from the function generator) between the range 0-400 Hz and observe the direction and rate of rotation of the shaft of the stepper motor. The rate of rotation should increase with frequency.
5. Verify that the direction of rotation changes depending on whether the **direction switch** is on (engaged) or off (disengaged).
6. **<u>Ask your TA to check that you have successfully completed Experiment #4.</u>**

Saving Files

Before you leave, remember to save all of your files to a **floppy disk** (for later use and backup purposes). For this laboratory, you should save the following file from the `Desktop`:

```
yourname_lab8_ex1.vi.
```

[*] Hint: before building the circuit, refer to Figure 7 and select appropriate pins on your XOR chip to implement the logic circuit shown in Figure 8. Write down those pin numbers on Figure 8.

Laboratory Report (NOT REQUIRED)

1. For the VI you wrote in this laboratory (listed in the preceding section), provide a printout that shows the front panel and block diagram.
2. In instruction 4 during Experiment #2, you were told that “Your stepper motor shaft *may* move initially, but it will quickly come to a stop.” In which scenario will the stepper motor not move at all? In which scenarios will the stepper motor initially move, but quickly come to a stop? Explain.
3. In Experiment #2, you ran your VI using the **Run** button to verify the full-step sequence, but you used the **Run Continuously** button to verify the half-step sequence. Explain why using the **Run Continuously** button to verify the full-step sequence would have been problematic.
4. Figure 10.28 in the Mechatronics textbook shows the timing diagram for the two least significant output bits of a binary counter. Draw a timing diagram that shows the clock input bit and the output bits for the binary counter you observed in Experiment #3. That is, provide a timing diagram that clearly shows **CLOCK, Q_A, Q_B, Q_C,** and **Q_D**. Make sure that your signals are in phase.
5. Redraw the analog drive circuit shown in Figure 8, but include the pin numbers you used for the XOR gate chip.
6. In this Laboratory, you controlled the angular position and/or the angular velocity of the stepper motor shaft. Would you describe your control methods as **open loop** or **closed loop**? Explain.
7. What is the (listed) detent torque and holding torque of the stepper motor used in this Laboratory?*

You are NOT REQUIRED to submit a Report for this Laboratory.

Additional Reading

Feel free to read the following material to learn more about *Jameco* stepper motor specifications.

1. **Jameco Electronics Catalog,** , Page **41 (Stepper Motors)**, Jameco Electronics. Available online at www.jameco.com/Jameco/Products/ProdCT/motor.pdf.

* Refer to **Additional Reading** section.

Extra Credit Experiment: Build a Virtual Instrument for Controlling the Angular Velocity of a Unipolar Stepper

In this extra credit exercise, you will replace your function generator with a *LabVIEW* VI that will enable you to control the angular velocity of a unipolar stepper motor using your VI front panel.

1. Disconnect the function generator from your analog drive circuit.
2. Build the VI shown in Figure 9.

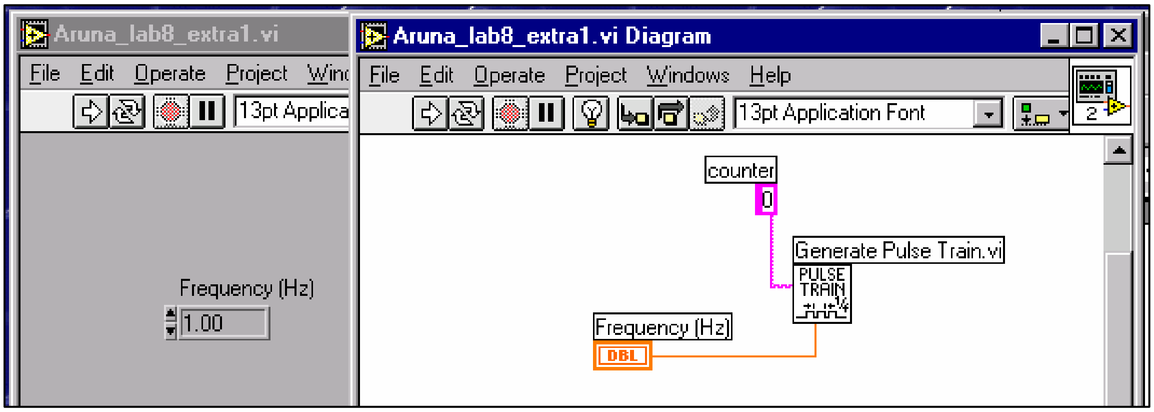

Figure 9

3. Connect the appropriate pin on your CB-68LP connector block to the **CLOCK** input pin on your counter chip. (Hint: See Lab #7, Experiment 5)
4. Set your `Frequency` to `1.00` Hz and run your VI by pressing the **Run** button. Your stepper motor shaft should rotate at the rate of 1 full-step per second. To change the rate of rotation, enter a different frequency and hit the **Run** button[*].
5. Save the above VI as `yourname_lab8_extra1.vi`.
6. **<u>Ask your TA to check that you have successfully completed Part 1 of the Extra Credit Experiment.</u>**
7. Modify the above VI so that you can also specify the direction of rotation of the stepper motor. (Hint: Remove the analog **direction switch**, including the 10 KΩ resistor, and replace it with a digital signal from your DAQ Board).
8. Test your new VI and save it as `yourname_lab8_extra2.vi`.
9. **<u>Ask your TA to check that you have successfully completed Part 2 of the Extra Credit Experiment.</u>**

[*] Recall from Experiment #5 in Lab #7 that the value you enter in the `Frequency` control will not be implemented until you hit the **Run** button. Also, the **Run Continuously** button will not work properly for this particular VI.