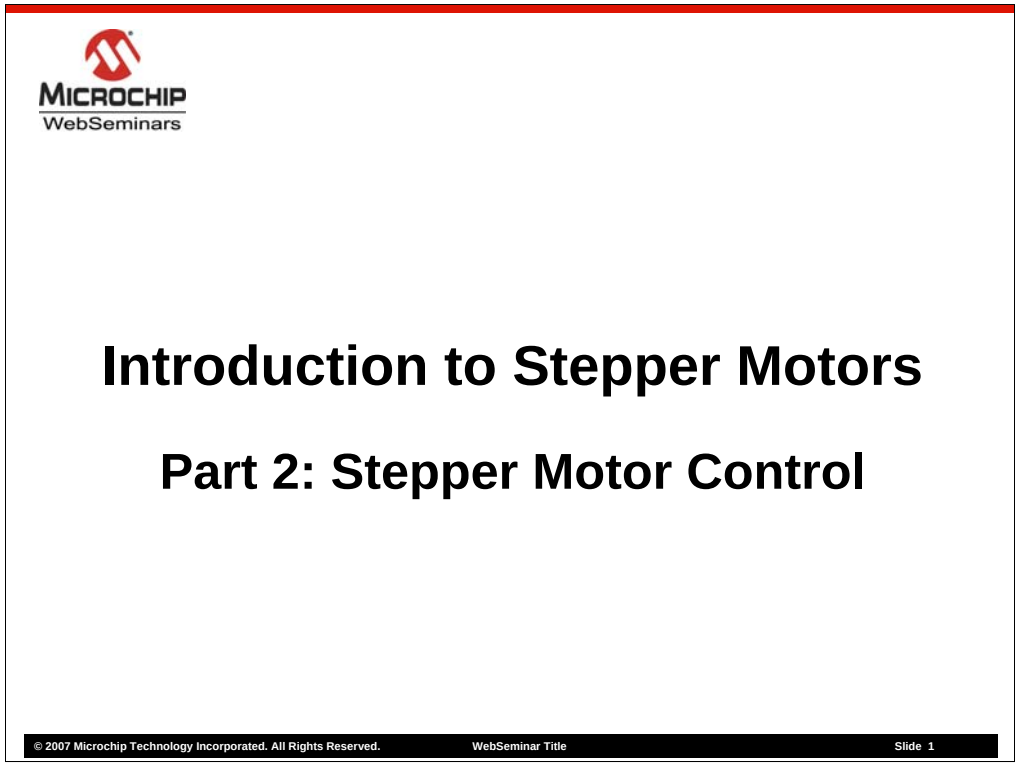
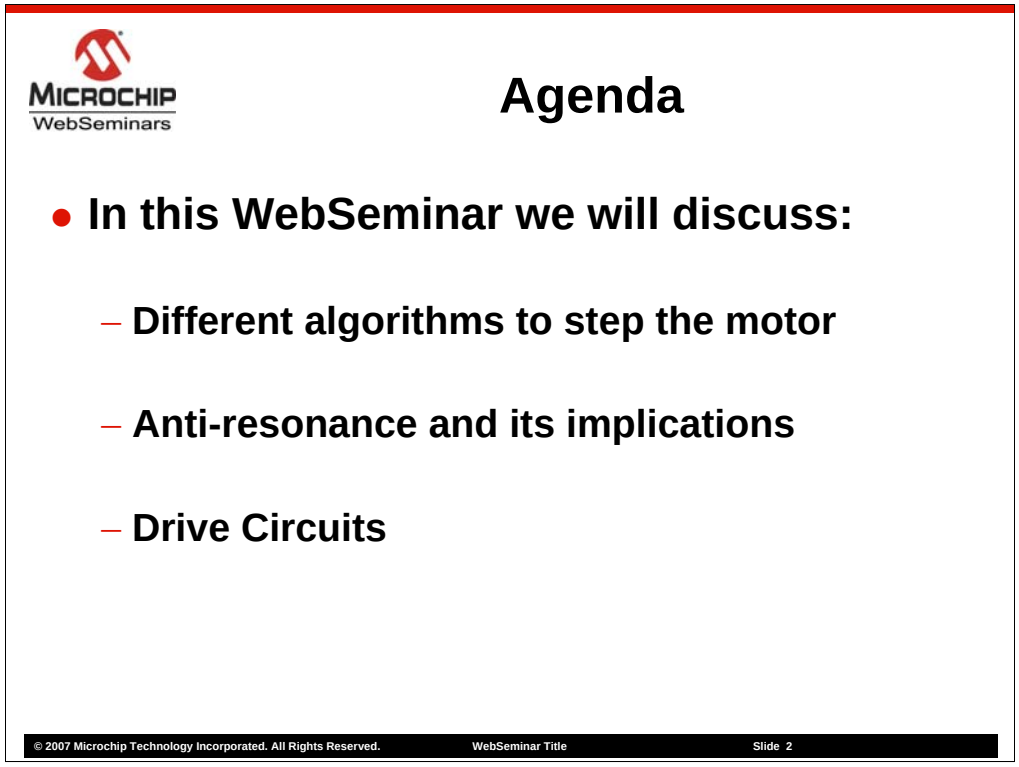


My name is Marc McComb, I am a Technical Training Engineer here in Microchip's Security, Microcontroller and Technology Division. Welcome to Part 2 in the "Introduction to Stepper Motors" series of Web Seminars, "Stepper Motor Control"

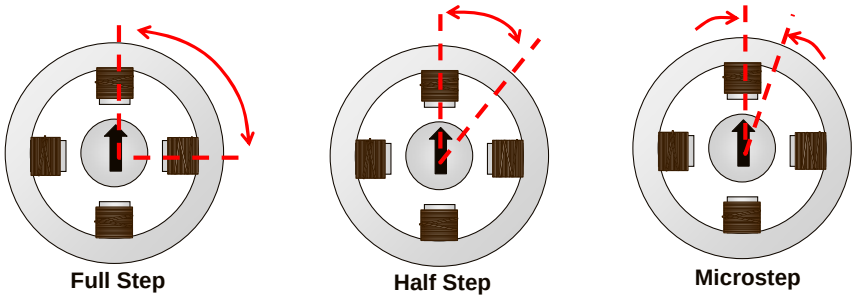


In the following webseminar we will expand on Part 1 by discussing different stepping algorithms to improve step resolutions. We will also discuss anti-resonance and how to deal with it. Finally, we will look at some basic circuitry to interface a stepper motor to a microcontroller.

MICROCHIP
WebSeminars

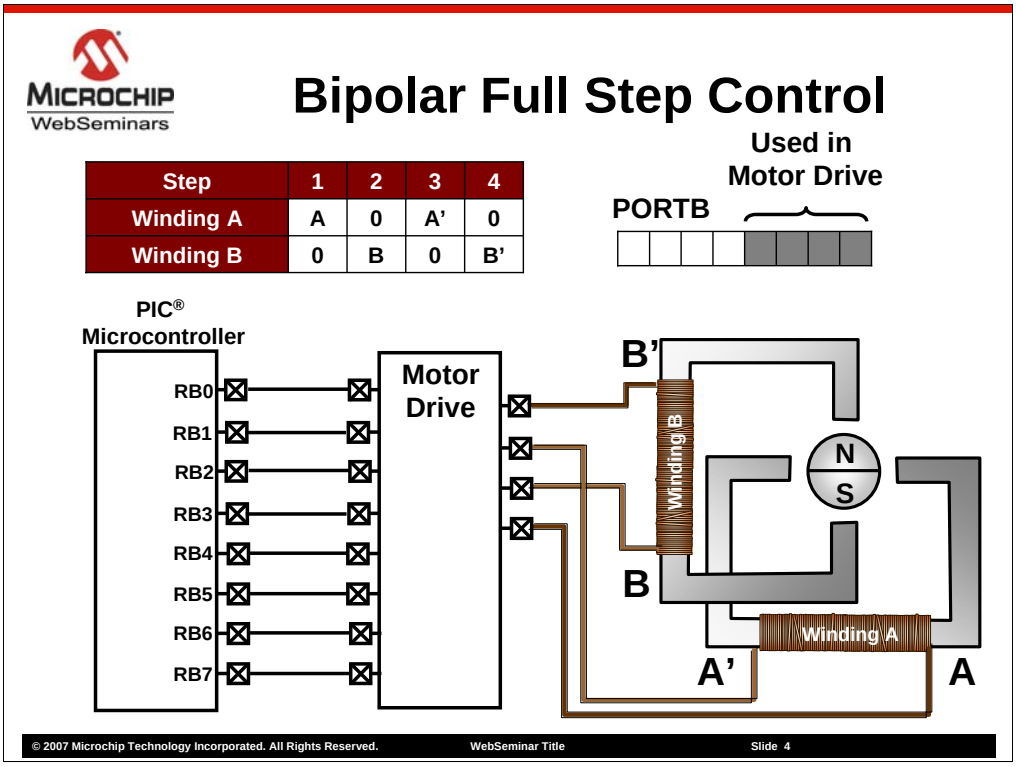
Stepping the Motor

- **Different step modes produce different step angles**

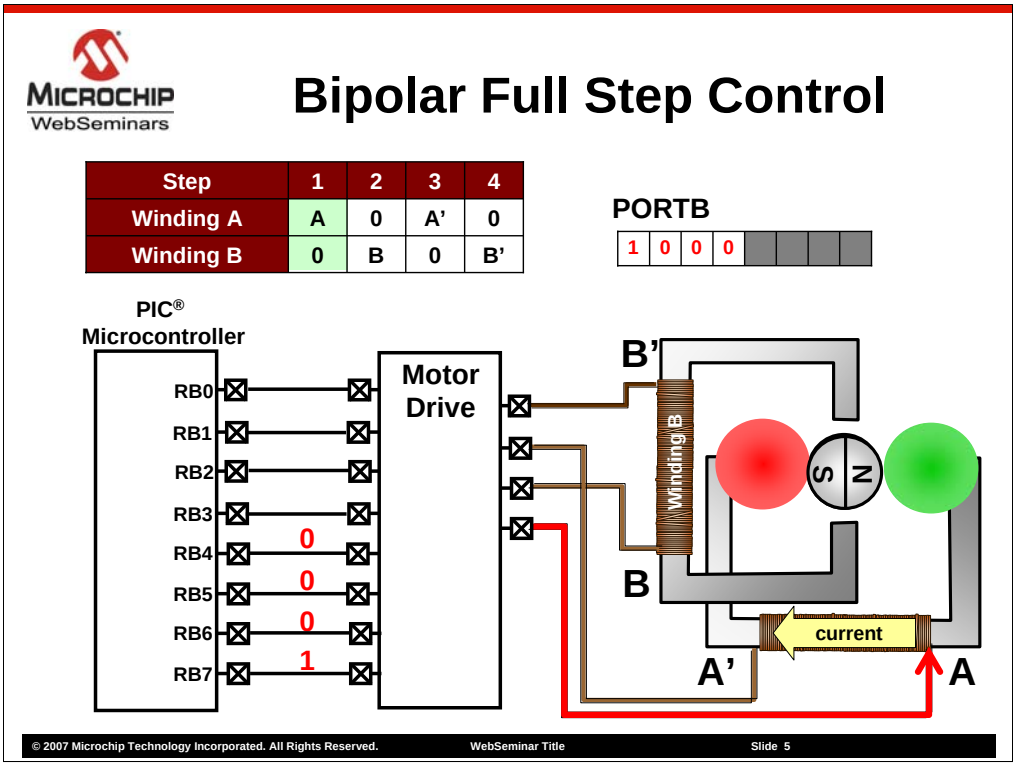


© 2007 Microchip Technology Incorporated. All Rights Reserved.

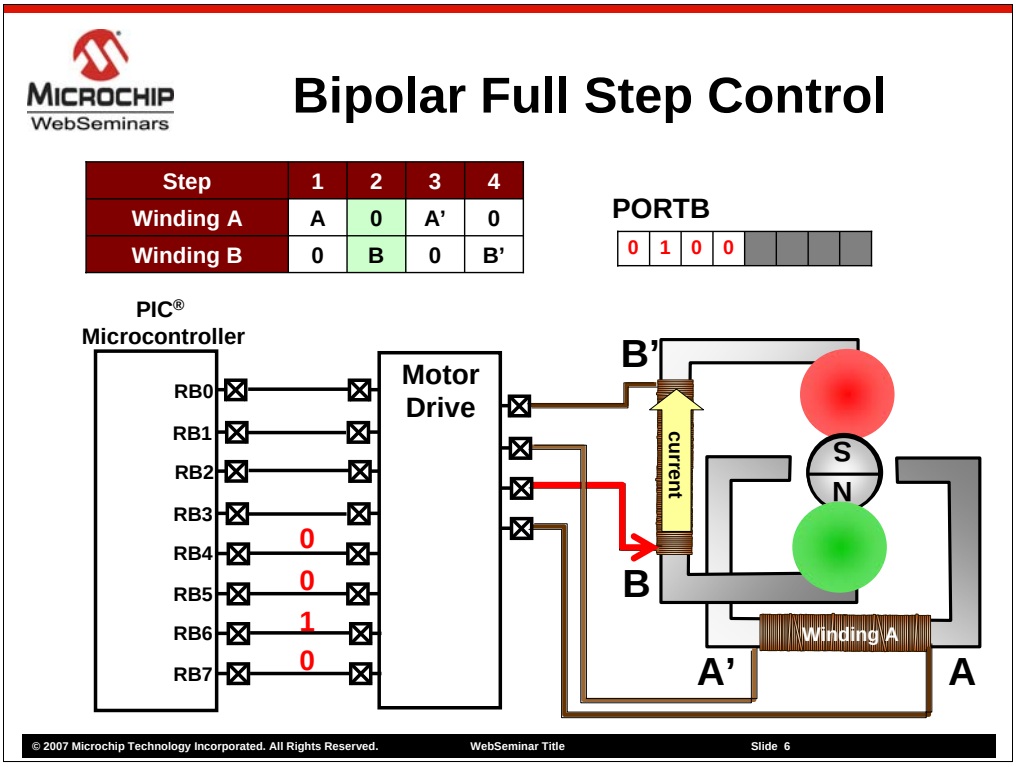
As we saw in Part 1, each stepper motor will have a defined step angle associated with it. In the above example we can see that with 2 phases, we have a step angle of 90 degrees. If we implement some basic techniques we can improve the resolution of the motor by decreasing the stepping angle.



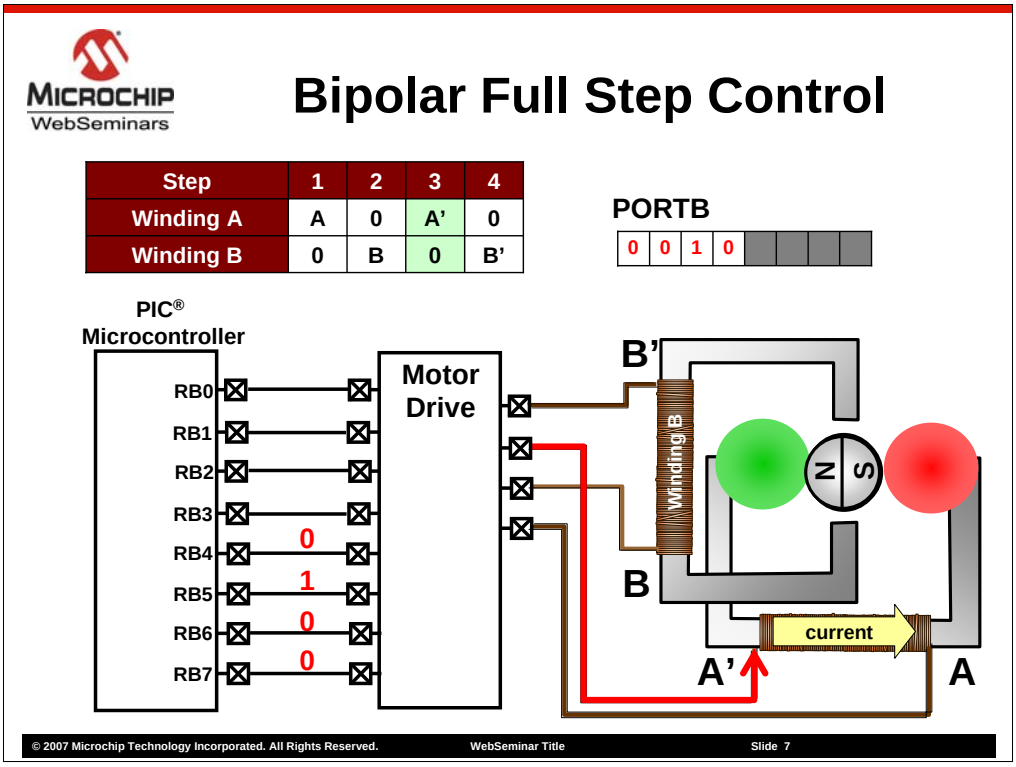
First, let's discuss full stepping. Here the rotor is rotated at its specified step angle resolution. In the above diagram, two windings are connected to a motor drive circuit which we will specify as a black box at this point. Later in this presentation we will look inside this black box. For now though we will concentrate on the motor windings and the PIC® Microcontroller. Notice that we are using a simple General Purpose Input/Output Port peripheral here PORTB as an example. We will focus our attention to the top 4 Most Significant Bits in PORTB for the time being as the Lower 4-bits are used in the Motor Drive "Black Box" Circuit. We will use the nomenclature throughout this presentation by defining each lead of each winding as follows: Winding A leads will be identified by leads A and A', while winding B leads will be identified by leads B and B'. At the top Left-Hand corner of the diagram is our stepping algorithm. Let's step through this algorithm.



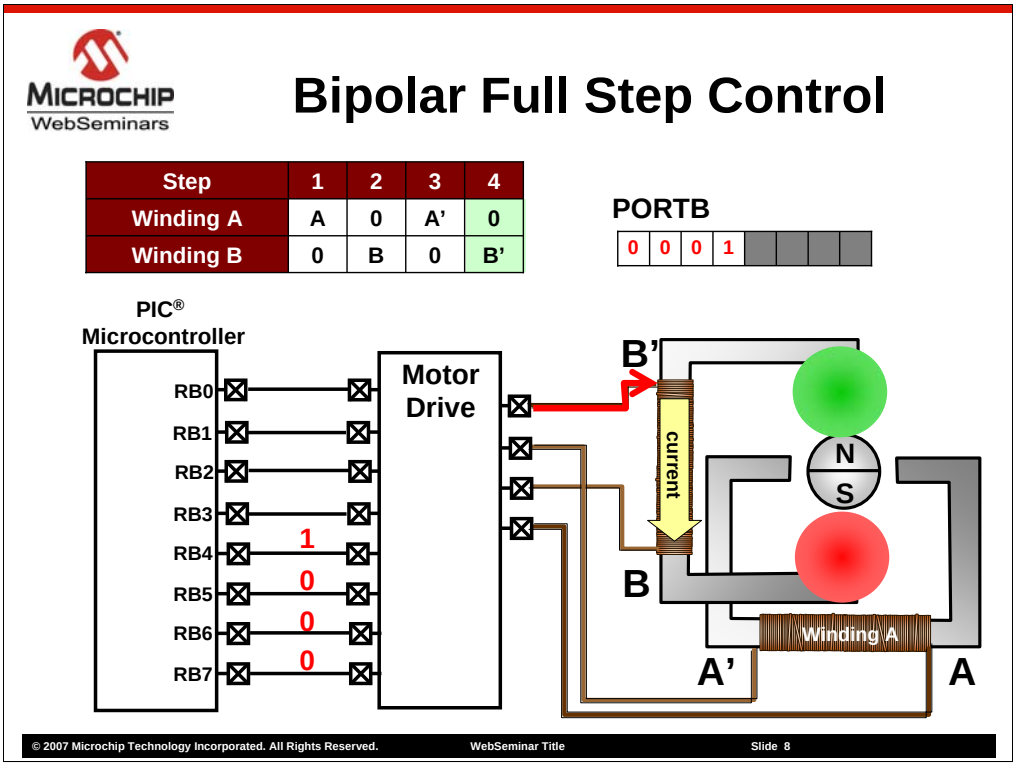
The first step applies a positive voltage or logic HIGH to Winding A's lead A while driving lead A' LOW. Current is generated in the direction shown creating a magnetic flux polarizing the stator poles accordingly. The rotor rotates to minimize the magnetic flux flow reluctance.



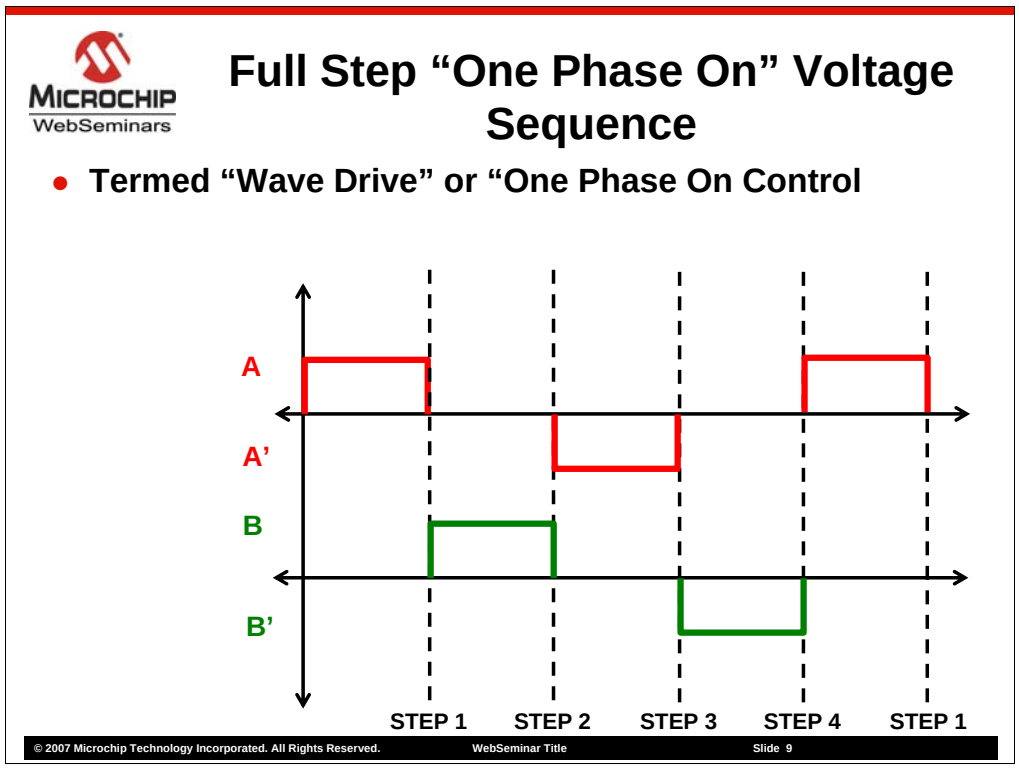
The next step removes the applied voltage to Winding A and drives lead B HIGH initiating current flow towards lead B' which is driven LOW. Again the rotor rotates minimizing the reluctance. Notice that as we step through this full step algorithm we are simply shifting a set bit right each time. Remember though, you will need to connect the motor lead to the appropriate pins to accommodate this algorithm.



Continuing through the algorithm, lead A' is next driven HIGH



Followed finally by driving lead B' HIGH to complete the 360 degree rotation.



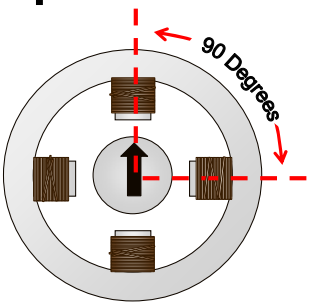
This type of full step algorithm is referred to as One Phase On Voltage sequence. The term “Wave Drive” is sometimes used as the voltage sequence resembles a wave. Each lead is energized one at time for each step.

MICROCHIP
WebSeminars

Determining Speed

- **Number of Steps per Revolution**
  = 360°/Angle for one step

Example
360°/90° = 4 steps/revolution

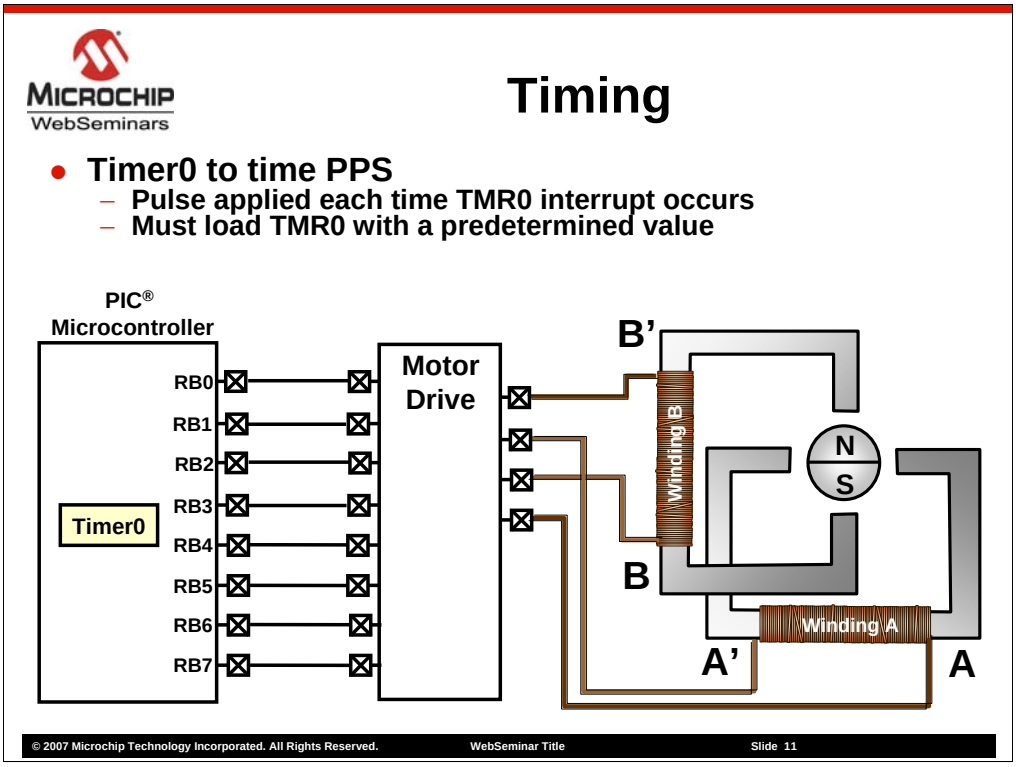


- **Number of Pulses Per Second (PPS)**
  =[(Desired RPM) / 60 seconds] x Number of Steps/revolution

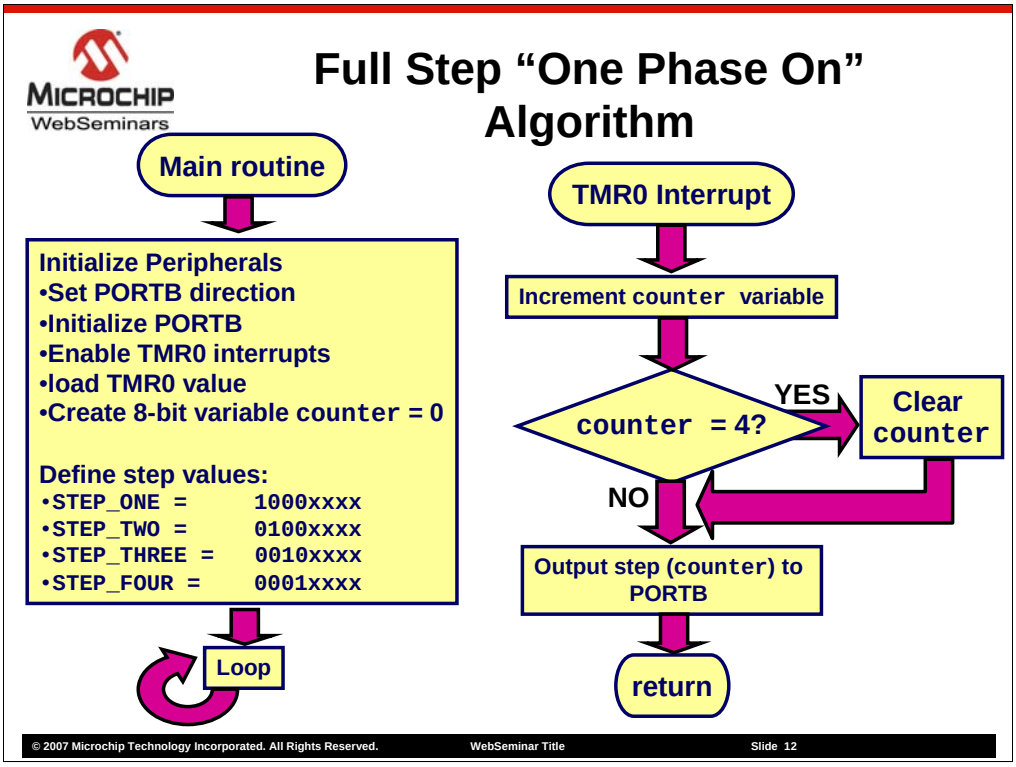
Example
[(120 RPM)/60 seconds] x (4 steps/revolution) = 8 pulses/second

© 2007 Microchip Technology Incorporated. All Rights Reserved.

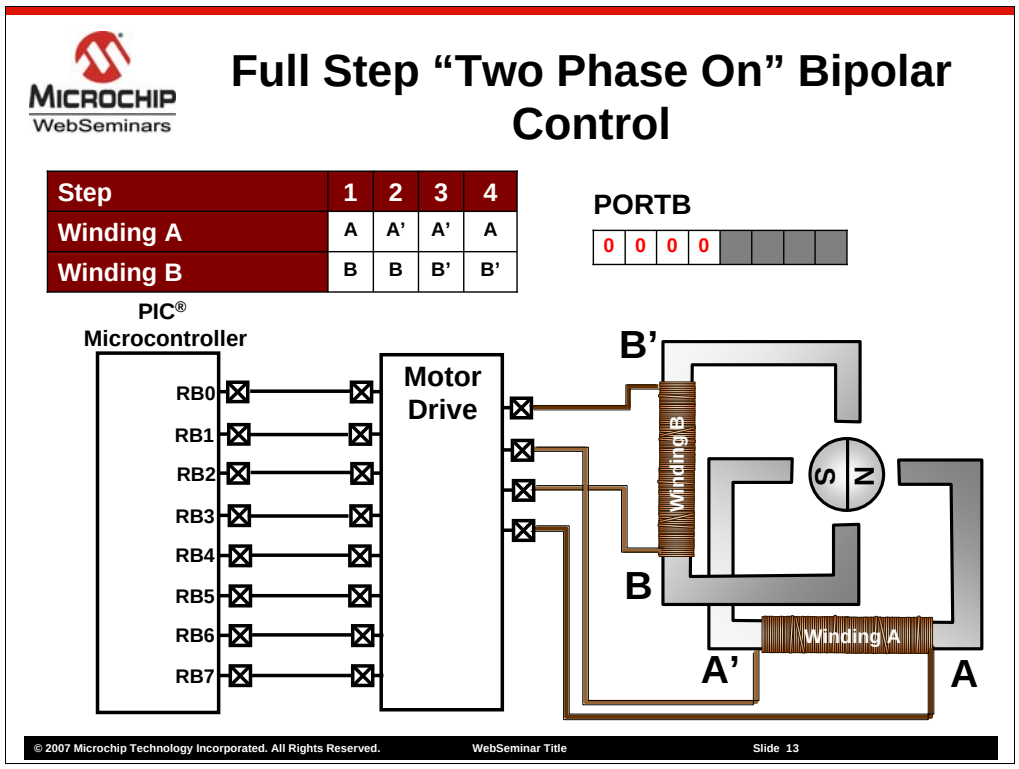
Let's take a moment and talk about speed of revolution or RPM. We can determine how fast to apply the individual steps by following a few simple equations. First we need to determine how many steps actually make up a complete 360 degree revolution. In this case, since we have a 90 degree step angle for each individual step, we can say that it will take four steps for a complete revolution. Next, we need to know how many pulses or steps we will apply per second to achieve the desired revolution. Therefore, we divide our desired RPM by 60 seconds and then multiply the quotient by the number of steps in a complete revolution. The product provides us with the number of steps required per second to obtain the desired RPM.



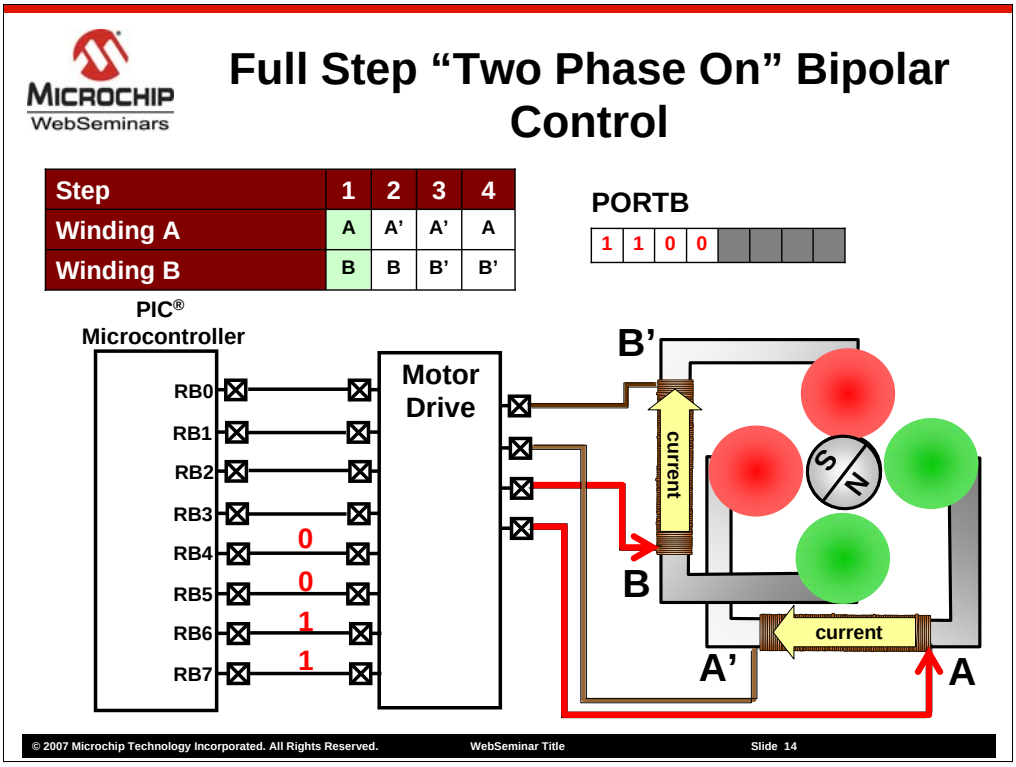
We can easily implement the steps per second using Timer0 interrupts. We must first configure the Timer0 prescaler accordingly and then load a pre-calculated value into the TMR0 register that will interrupt the CPU at the appropriate time intervals to perform subsequent steps.



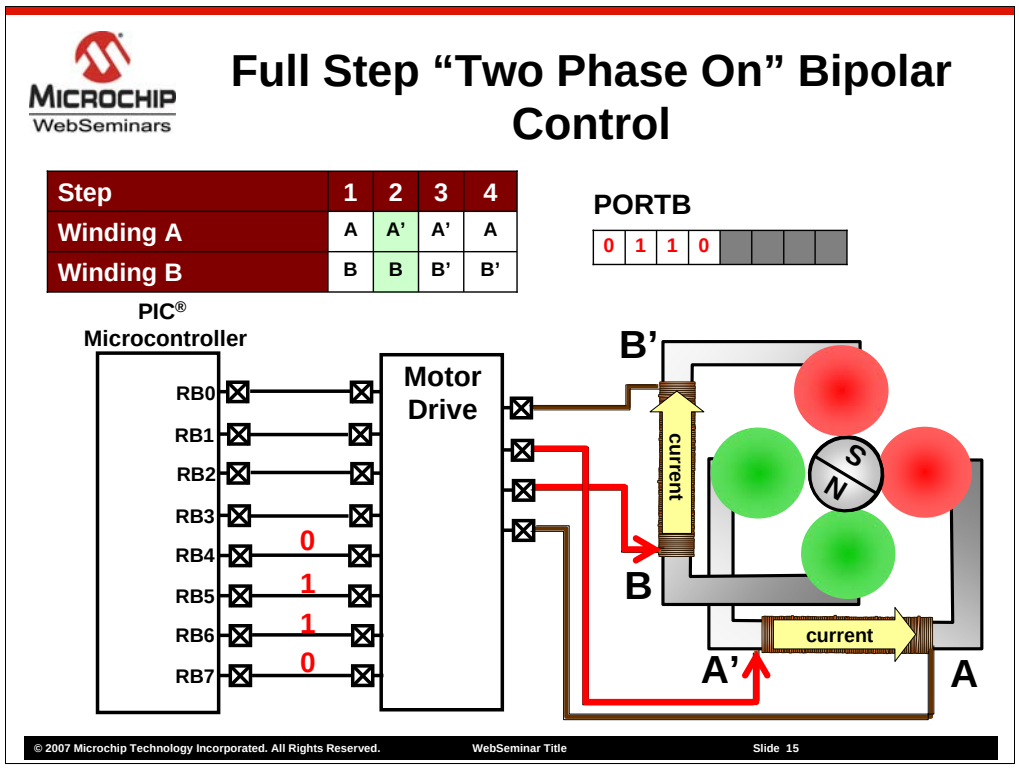
Referring to the above flow chart, to implement this in software, we must first initialize the two peripherals PORTB and Timer0 as discussed. We must also define values to pass to the PORTB register that will produce the desired output sequence as well as define a counter variable. Following peripheral initialization and variable definitions, we simply place the CPU into a loop. On a Timer0 interrupt the counter variable is incremented, checked and then used to determine which step value is outputted to the PORTB register. Remember, if code development is done in C, the counter variable will need to be declared as a global variable.



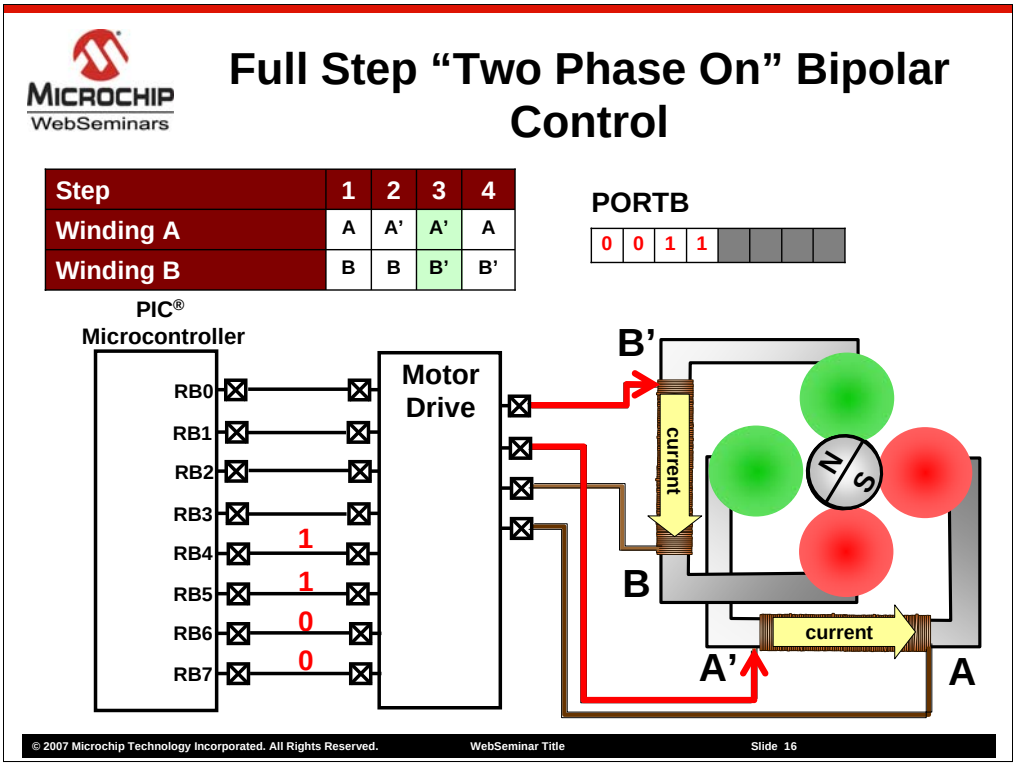
The next full step algorithm is the Two Phase ON Bipolar control sequence. In this algorithm, two phases are energized simultaneously to rotate the rotor. Again, in our diagram the individual lead of Windings A and B are connected to the same "Black Box" motor drive circuit which is connected to PORTB. Note that now our stepping algorithm shown in the upper left corner of the slide has changed from the One Phase On algorithm we have just discussed.



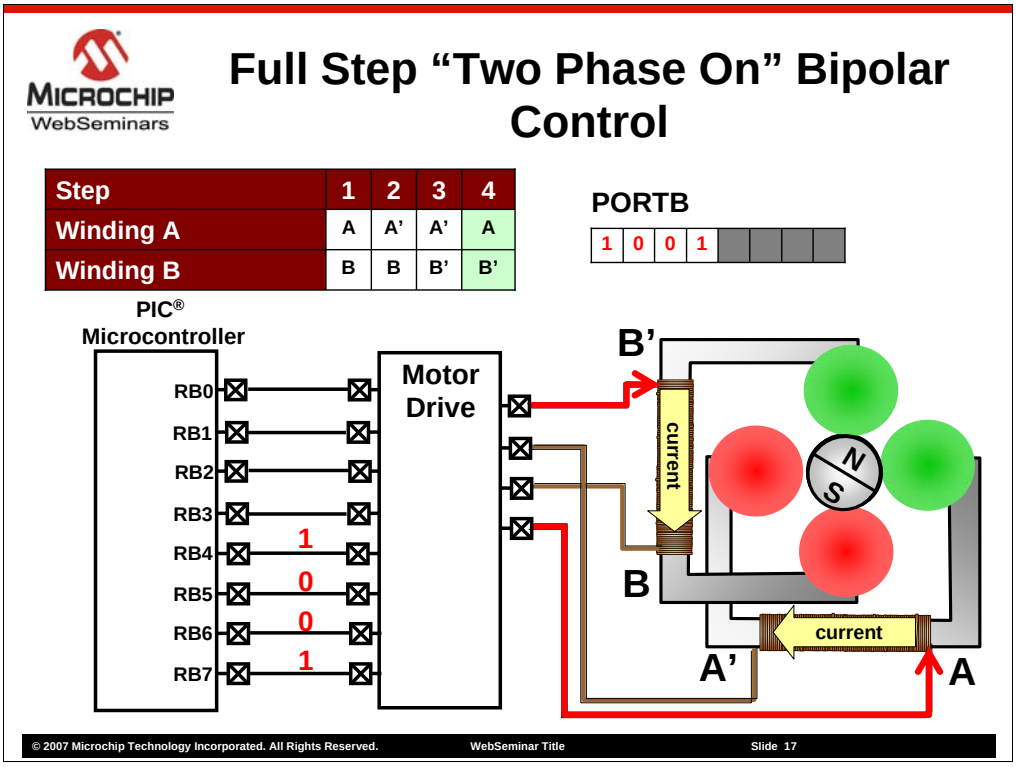
Driving both leads A and B HIGH while keeping A' and B' LOW produces current flow in both windings thereby generating a polarity on all stator poles. Notice also that the rotor's pole pairs are now located between two stator poles as opposed to being lined up with a single stator pole as we saw in the One phase On algorithm.



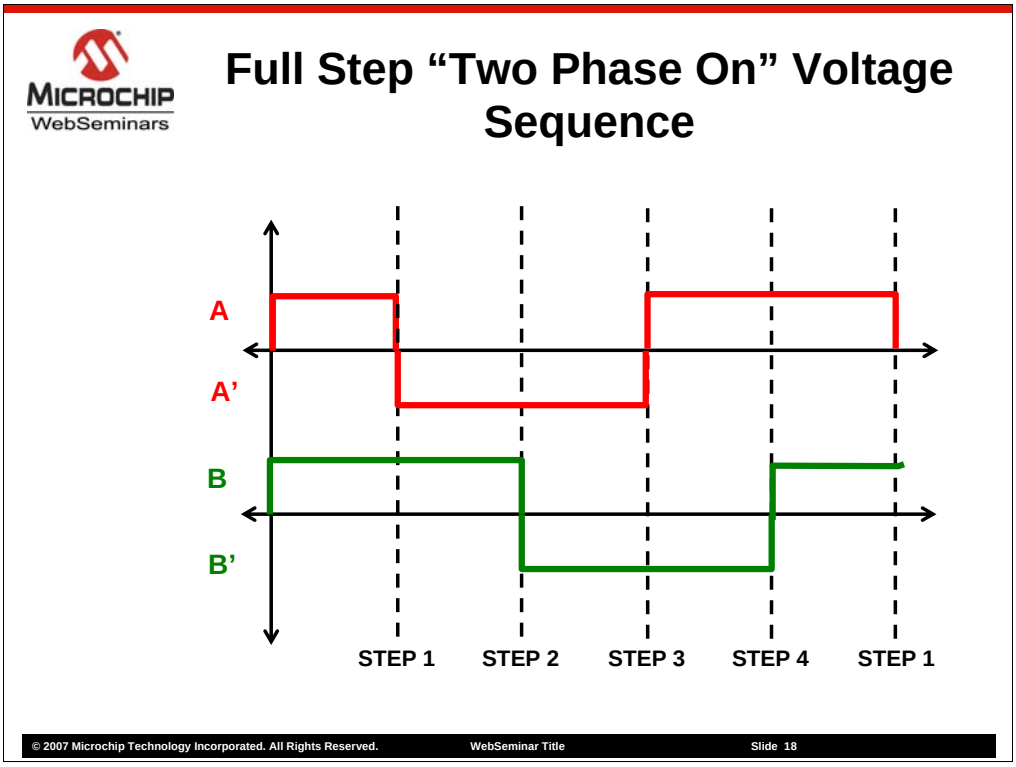
The next step in the algorithm maintains the current flow direction in Winding B while reversing the current direction in Winding A. This causes the rotor rotate 90 degrees so that it lies between the next two stator poles.



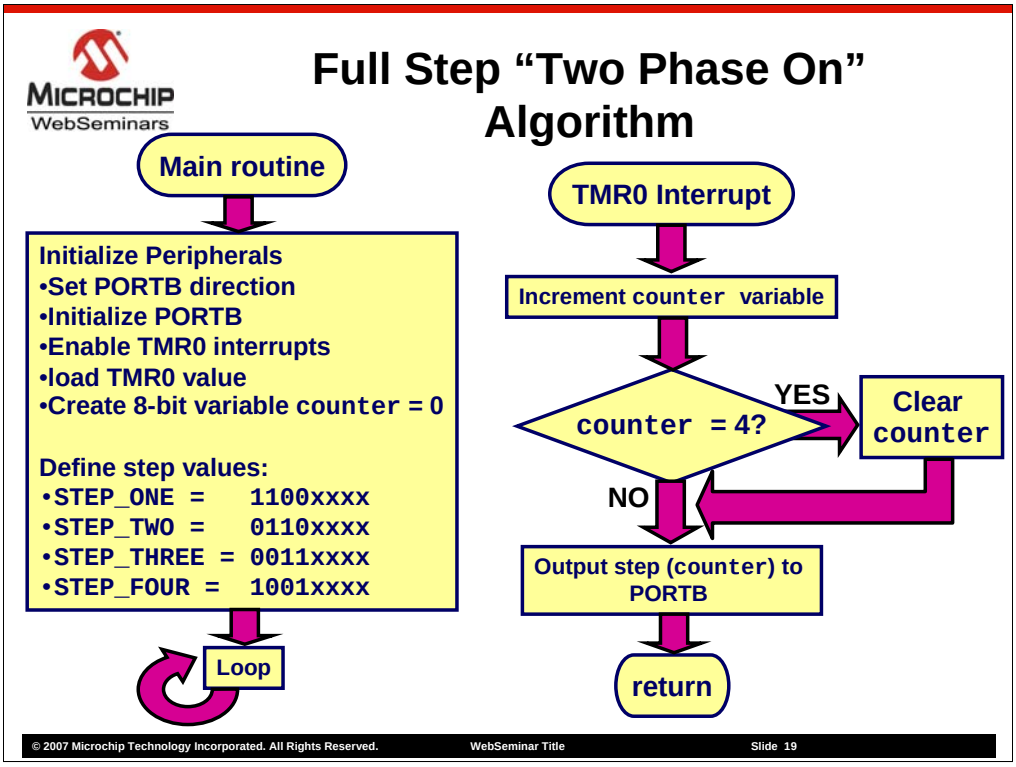
As we continue through the algorithm, current direction through winding A is maintained from the previous step while this time current direction is changed in Winding B.



The final step rotates the rotor to its starting position.

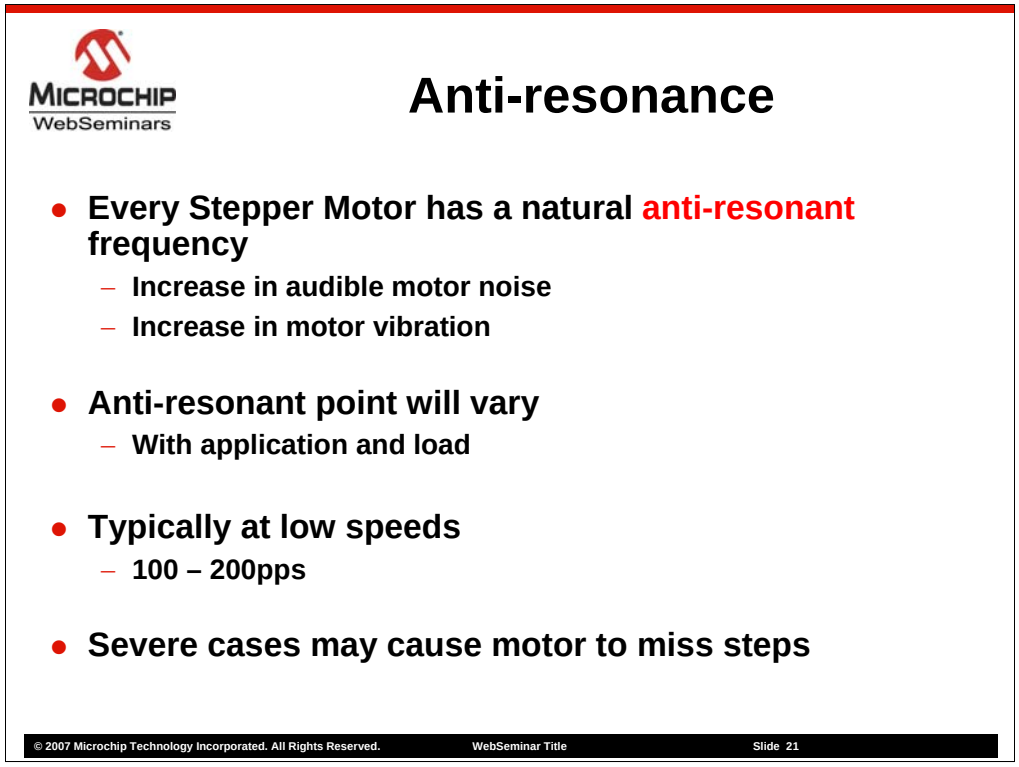


If we look at the voltage sequence for the Two Phase On algorithm we can clearly see that at any given time current is flowing it both windings.

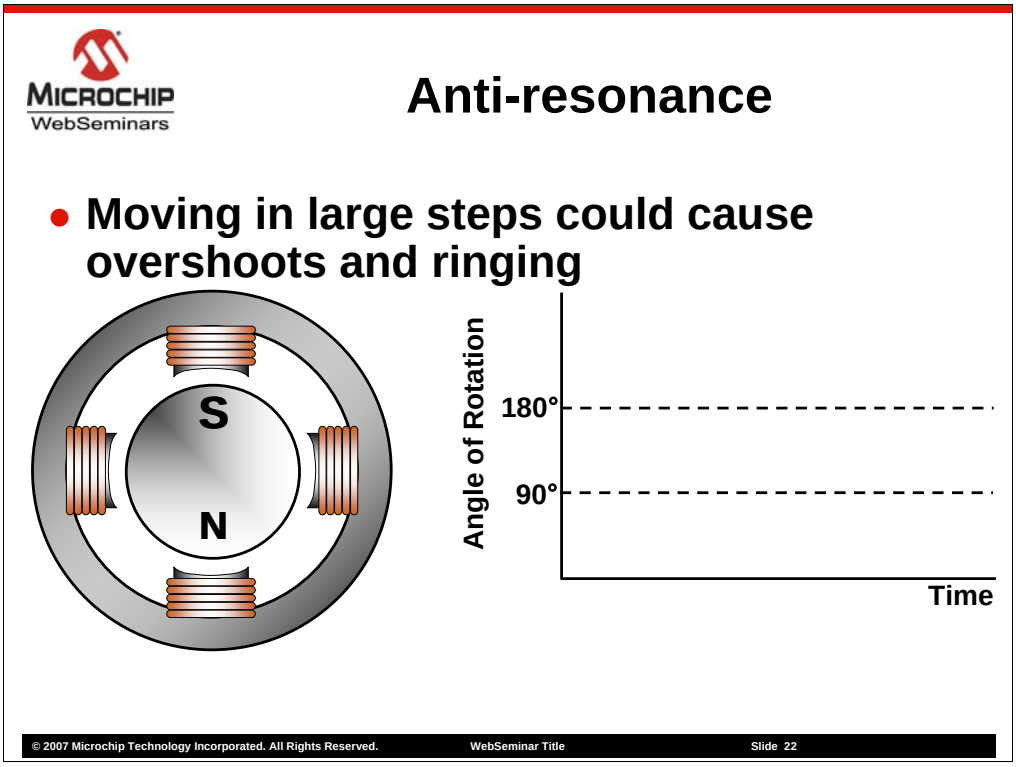


So how does this change our software algorithm? Apart from redefining the step values the rest of the flowchart remains unchanged.

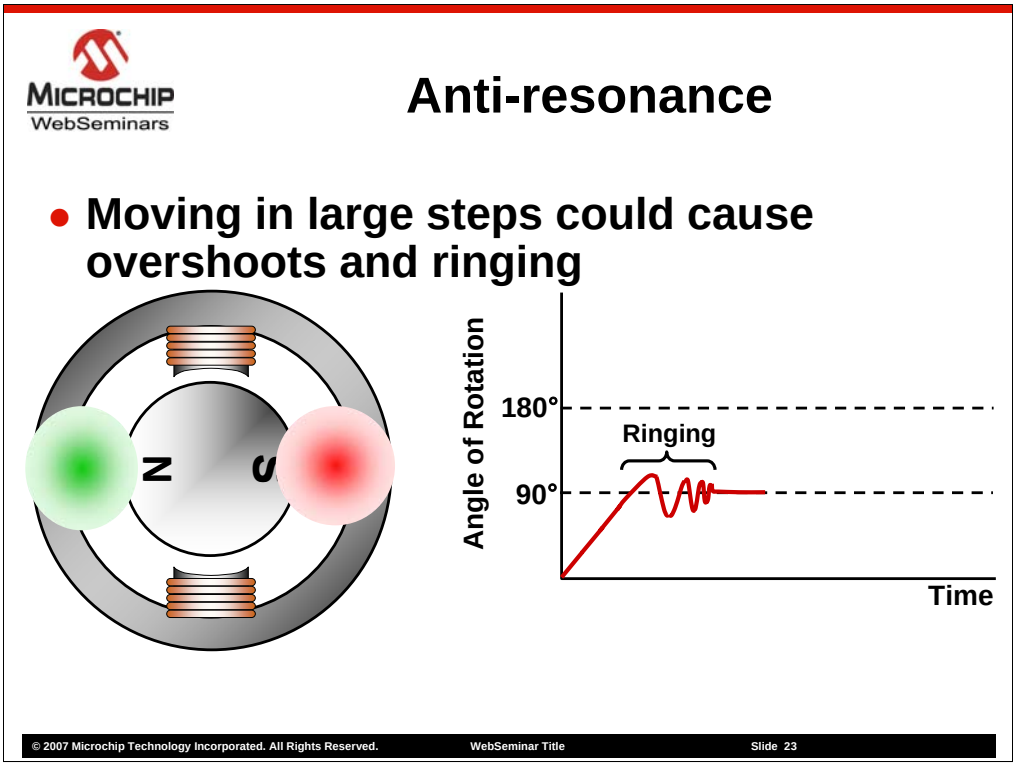
Now that we have looked at both full step algorithms, we need to introduce a condition known as anti-resonance



Every stepper motor will have anti-resonant points typically centered around the stepper motor's resonant frequency. Resonance actually helps rotate the rotor, anti-resonance on the other hand impedes it. Anti-resonant points typically occur at lower speed but are mostly dependant on the application and load on the rotor itself. Anti-resonant points are characterized by increased motor vibration along with audible motor noise. As we will see, in severe cases anti-resonance will interfere with rotor rotation to such an extreme that some steps in the full step algorithm will actually be missed.



Looking at the above diagram, on the left is a simple permanent magnet motor. On the right, a graph that will be used to represent angular rotation of the rotor.

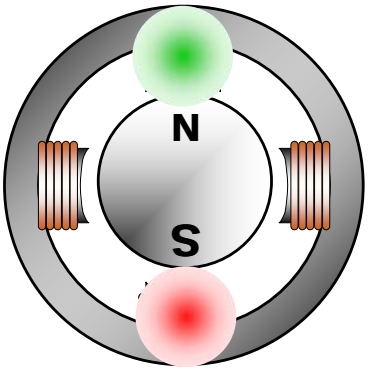


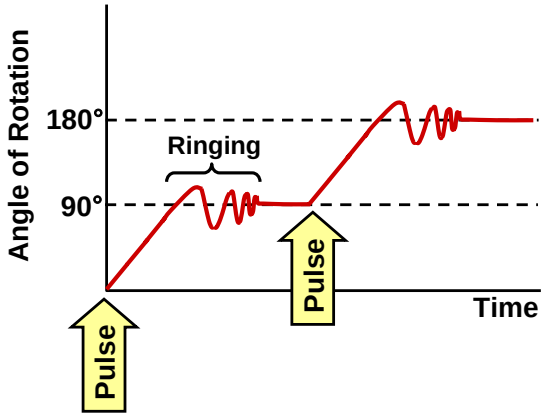
When a step is executed the rotor shaft will undergo a period of time where a ringing occurs before finally settling at the energized stator pole pair.

MICROCHIP
WebSeminars

Anti-resonance

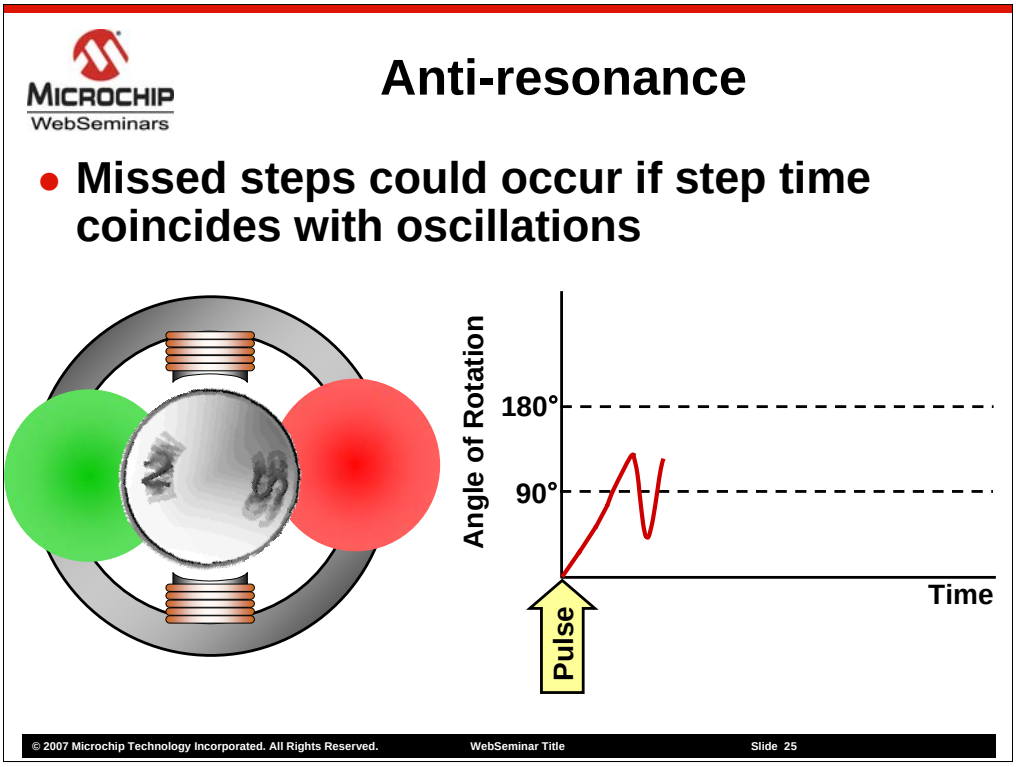
- **Moving in large steps could cause overshoots and ringing**



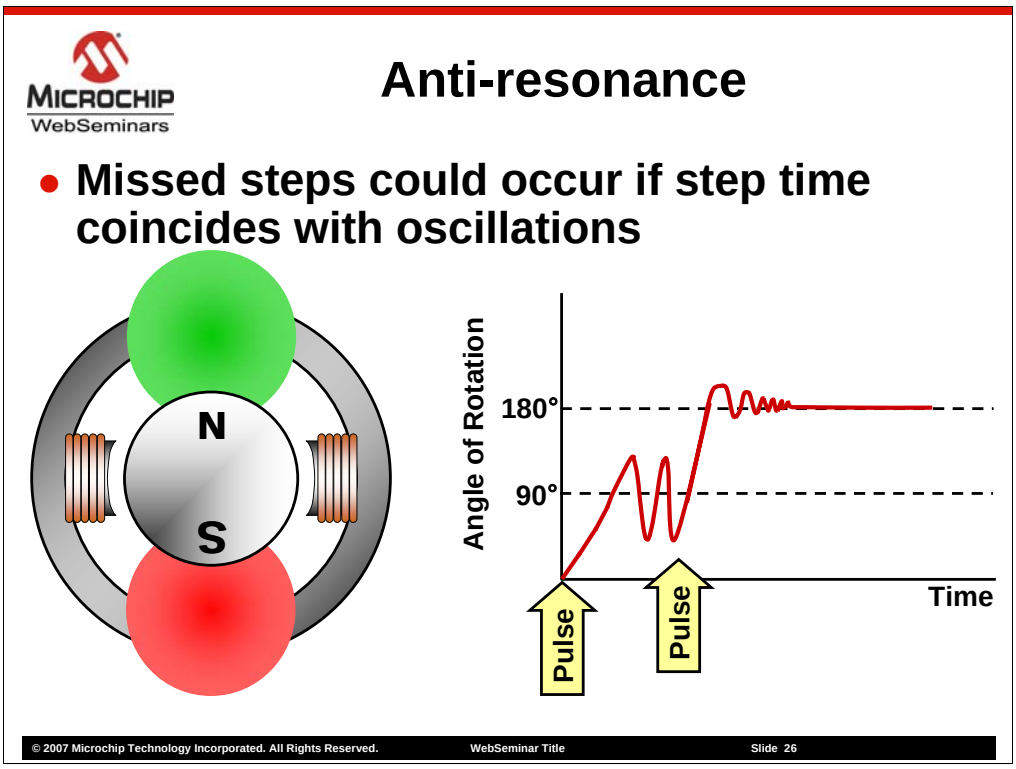


© 2007 Microchip Technology Incorporated. All Rights Reserved.

Subsequent steps will suffer from this same ringing.

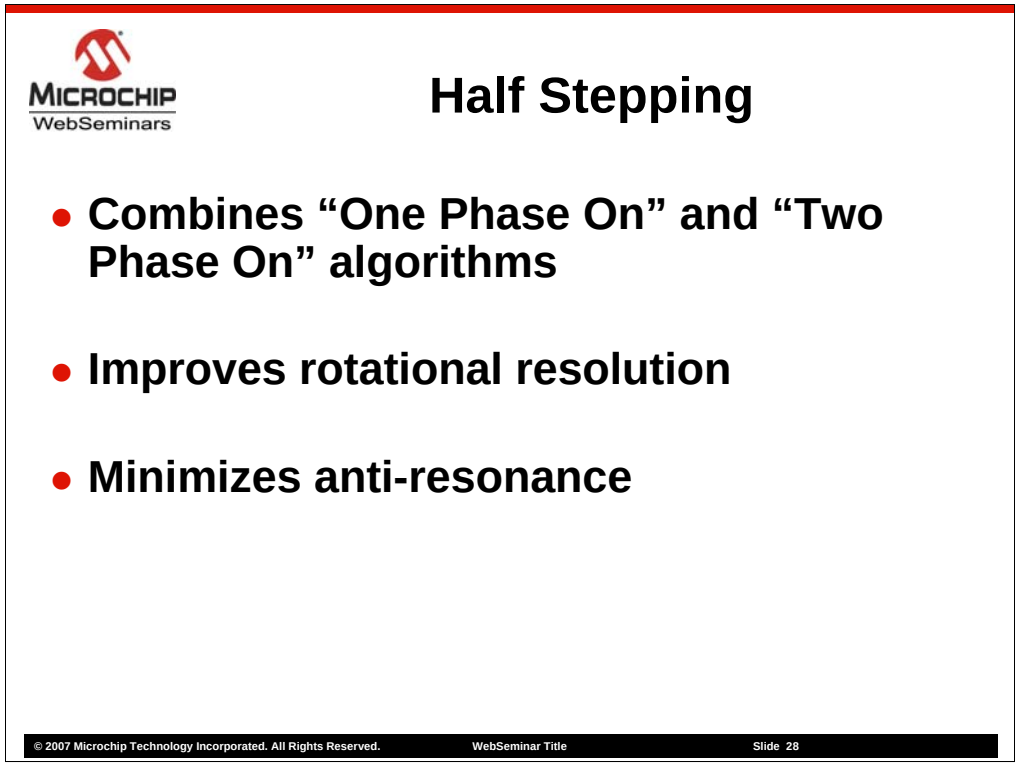


In severe cases, this ringing could be so pronounced that the rotor will not have time to settle before the next step pulse is applied.

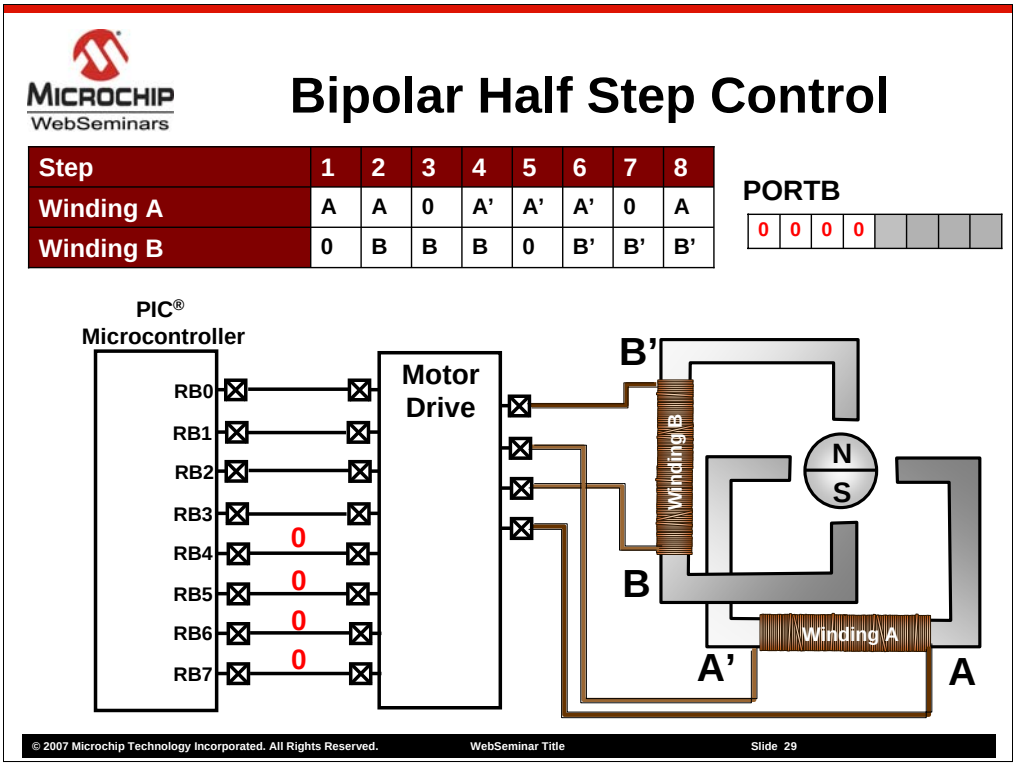


In the above example, the excessive ringing has caused the motor rotation to miss the first step at 90 degrees. In position sensitive applications this could have severe consequences. Not to mention that if you are not using a feedback network of any kind, rotor position will be undetermined.

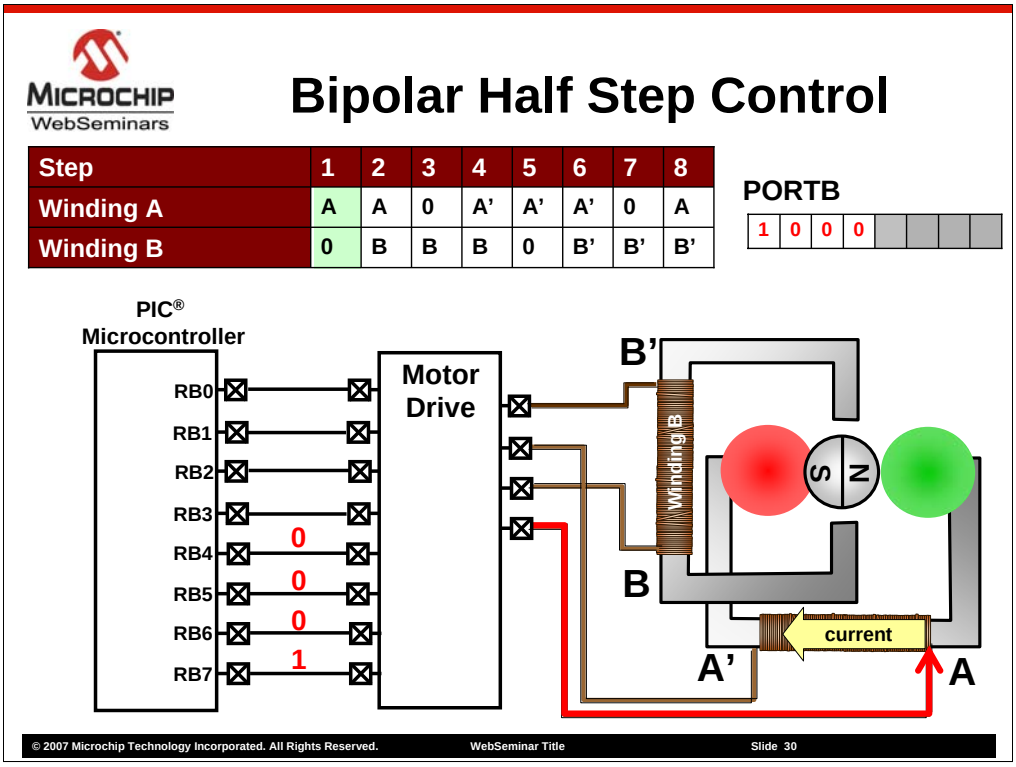
We can overcome anti-resonance in a couple of ways. Here we offer a change to the stepping algorithm as a solution.



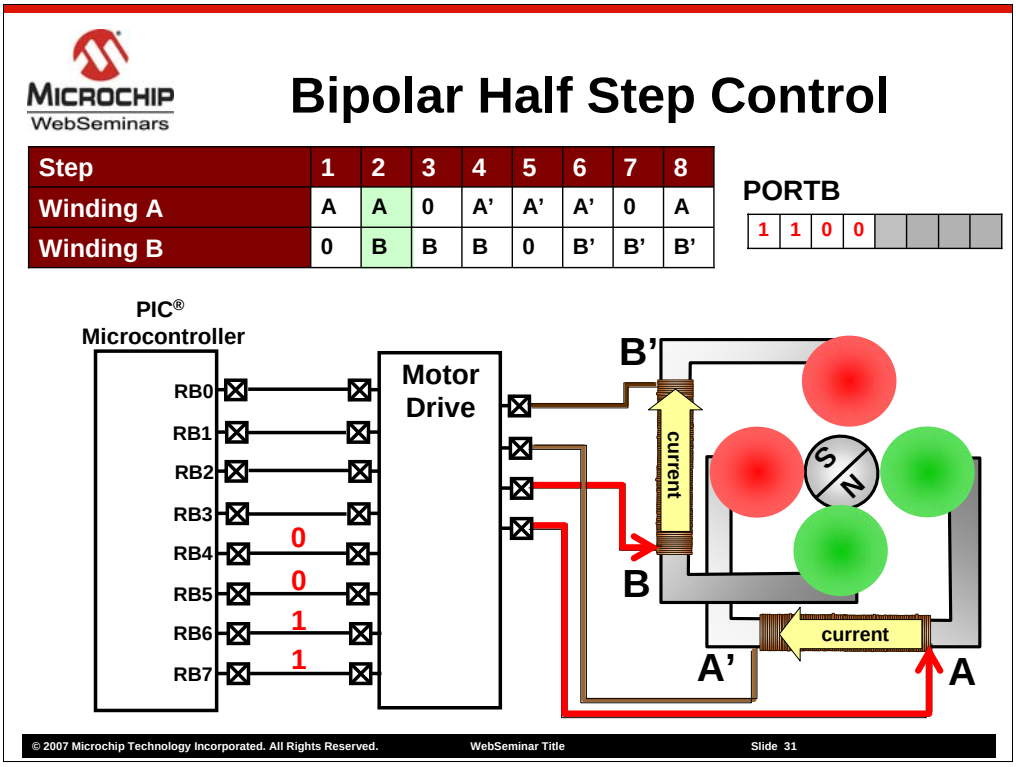
Half-stepping is a method of combining both One Phase On and Two Phase On full step algorithms. In doing so, the step angle is essentially halved. For example our 90 degree per step motor we have been using, will have a new step angle of 45 degrees when using half stepping. Since the rotor shaft doesn’t have as far to travel from one step to the other, the ringing produced at each step is minimized thereby reduce the anti-resonant effects exhibited using full step algorithms.



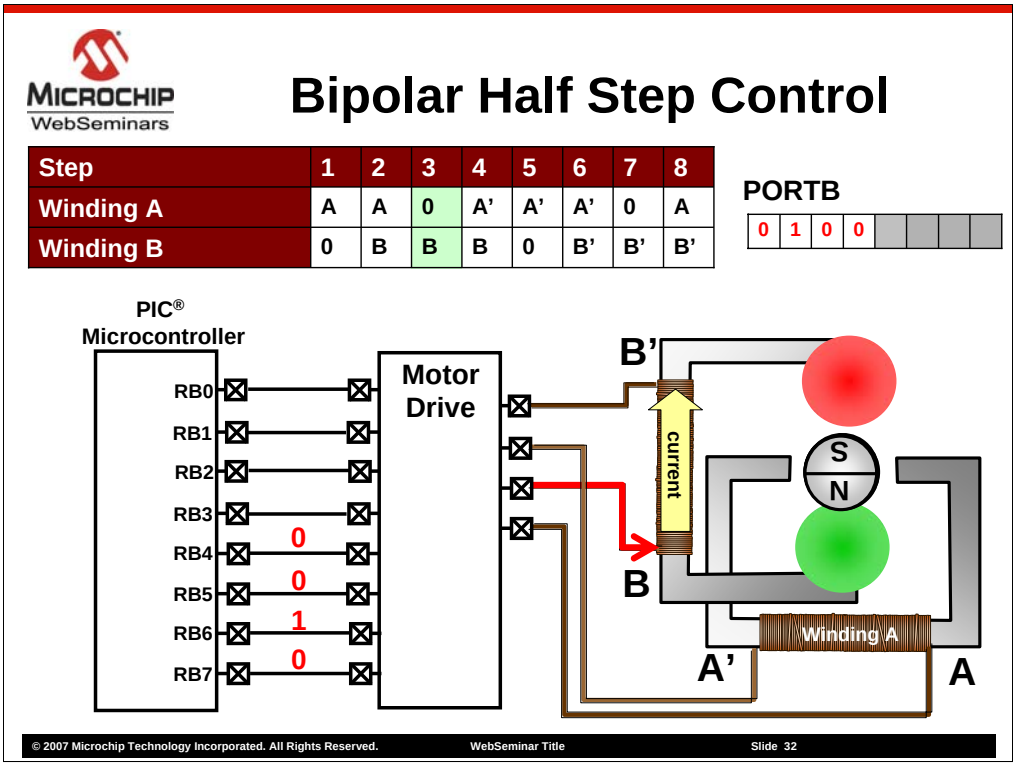
Let's take a look at how half-stepping is accomplished. Again, no change to our block diagram. However, notice that the step algorithm is now twice as long as in full-step examples. This makes sense considering that if we reduce the step angle by half…it will take twice as many steps to complete a 360 degree rotation.



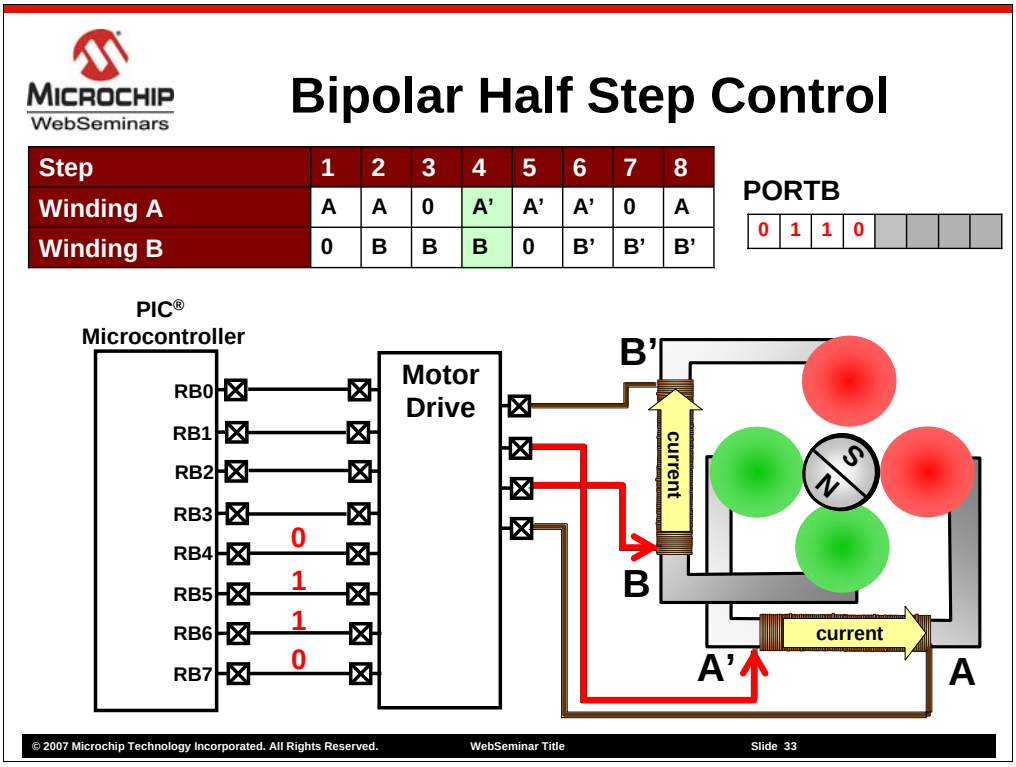
The first step in this new algorithm is actually the first step of the One Phase On algorithm we discussed. Current flow occurs in winding A only and the rotor responds by rotating to align its pole pairs with the stator poles.



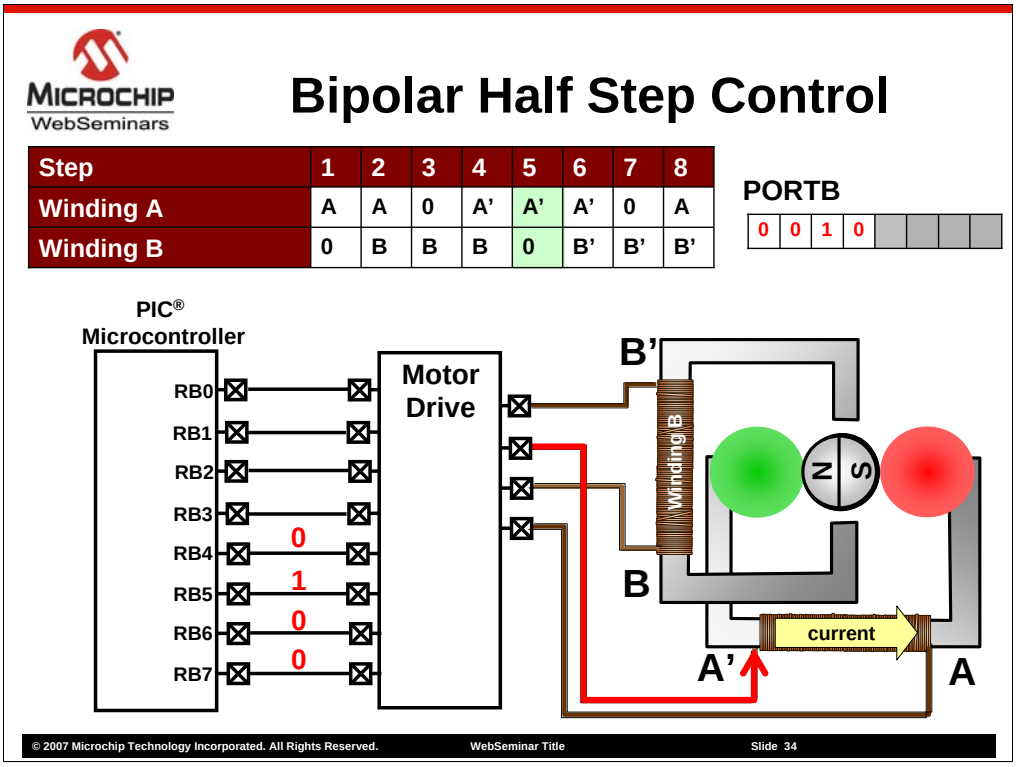
Next, the first step in the Two Phase On full-step algorithm is implemented. Current is maintained in winding A from the previous step only this time winding B is energized to produce current flow. Now the rotor, in the attempt to reduce the reluctance from the two Magnetic Flux produced, positions itself between stator poles.



Next, current is removed from winding A while maintaining current flow in winding B. This is the second step in the One Phase On algorithm.



Moving through the rest of the half-step algorithm….



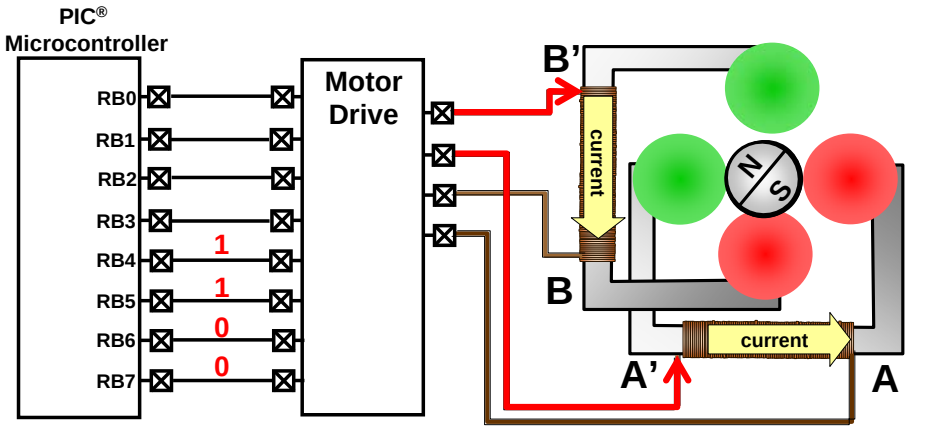
we are simply combining One Phase On and Two Phase On algorithms and executing each step sequentially.

Bipolar Half Step Control

Step	1	2	3	4	5	6	7	8
Winding A	A	A	0	A’	A’	A’	0	A
Winding B	0	B	B	B	0	B’	B’	B’

PORTB

0	0	1	1				



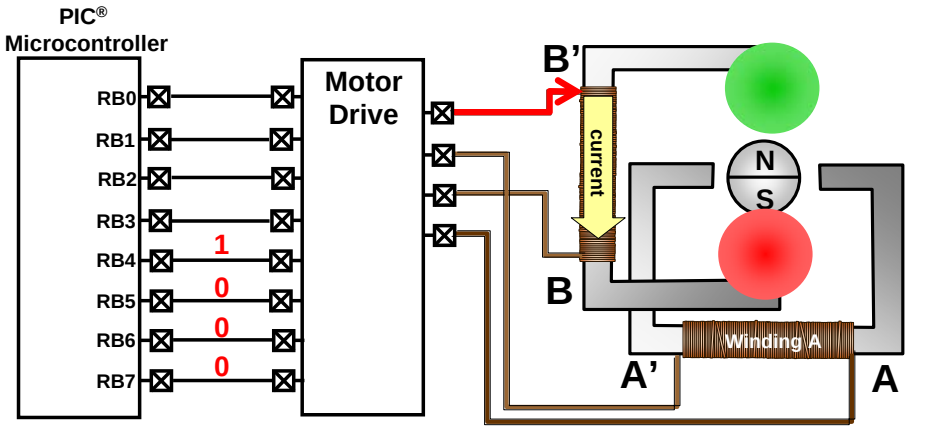
© 2007 Microchip Technology Incorporated. All Rights Reserved.

Bipolar Half Step Control

Step	1	2	3	4	5	6	7	8
Winding A	A	A	0	A’	A’	A’	0	A
Winding B	0	B	B	B	0	B’	B’	B’

PORTB

0	0	0	1				



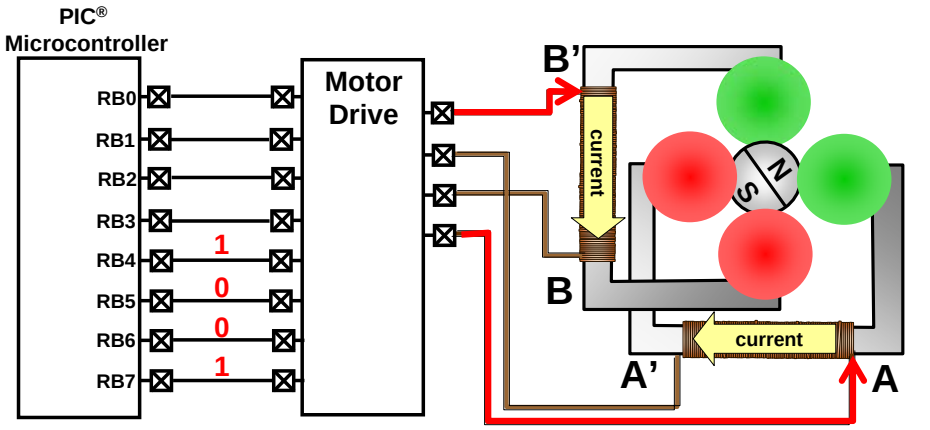
© 2007 Microchip Technology Incorporated. All Rights Reserved.

Bipolar Half Step Control

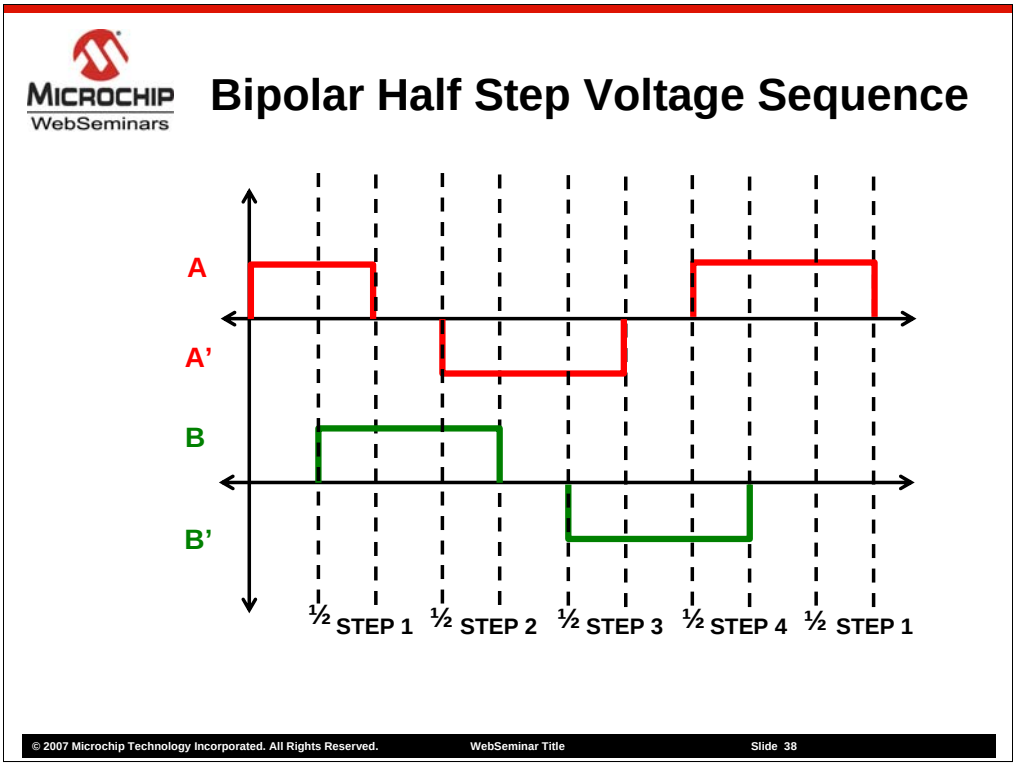
Step	1	2	3	4	5	6	7	8
Winding A	A	A	0	A’	A’	A’	0	A
Winding B	0	B	B	B	0	B’	B’	B’

PORTB

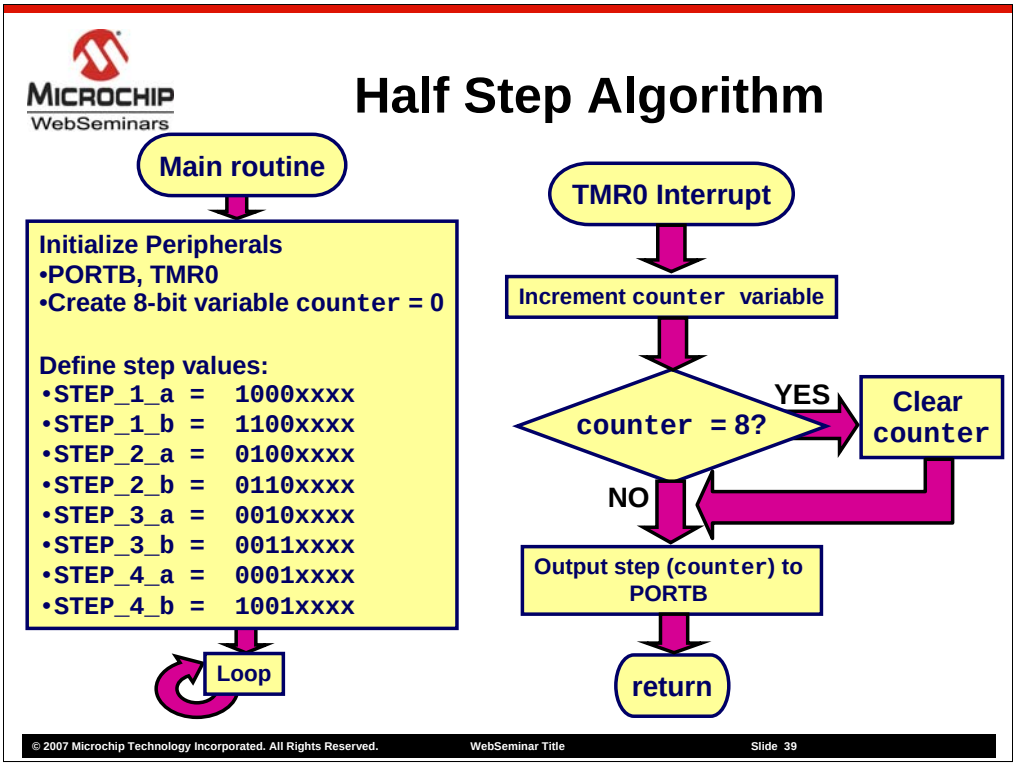
1	0	0	1				



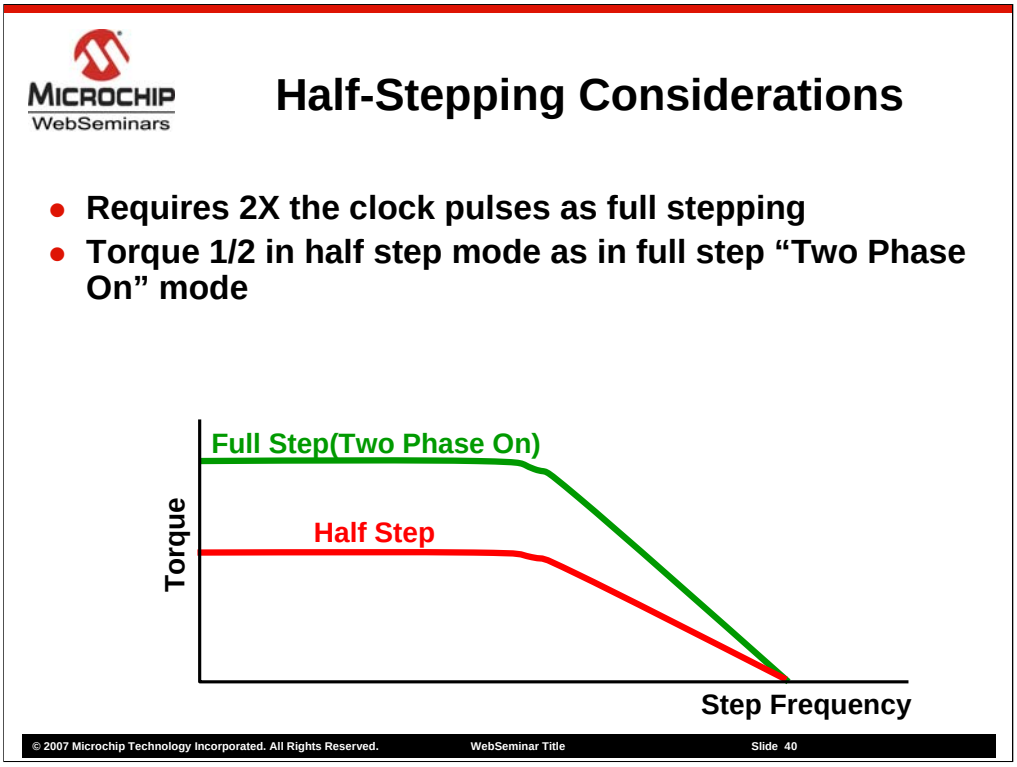
© 2007 Microchip Technology Incorporated. All Rights Reserved.



Referring to the Half-Step Voltage Sequence, current flows in one winding only half the time and in both windings for the other half.



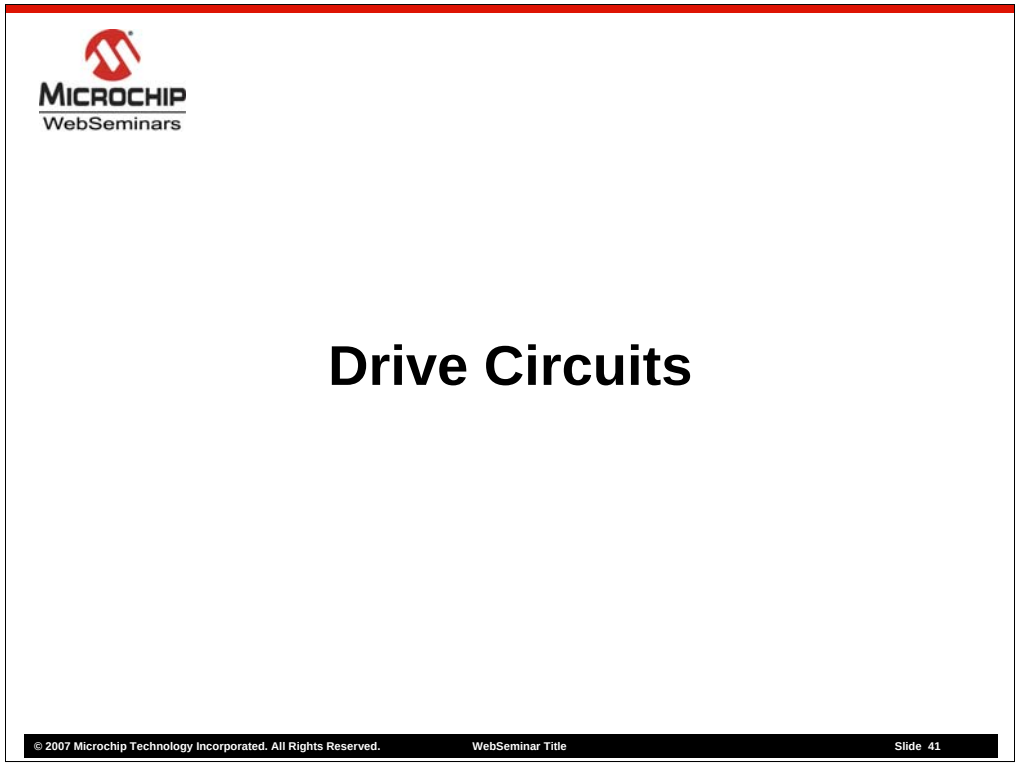
Some changes will be needed to our software flowchart. First, Two Phase ON and One Phase ON steps will need to be combined. Also, since we need twice the number of steps to generate a 360 degree revolution, we now need to increase our counter value to 8 before clearing it.



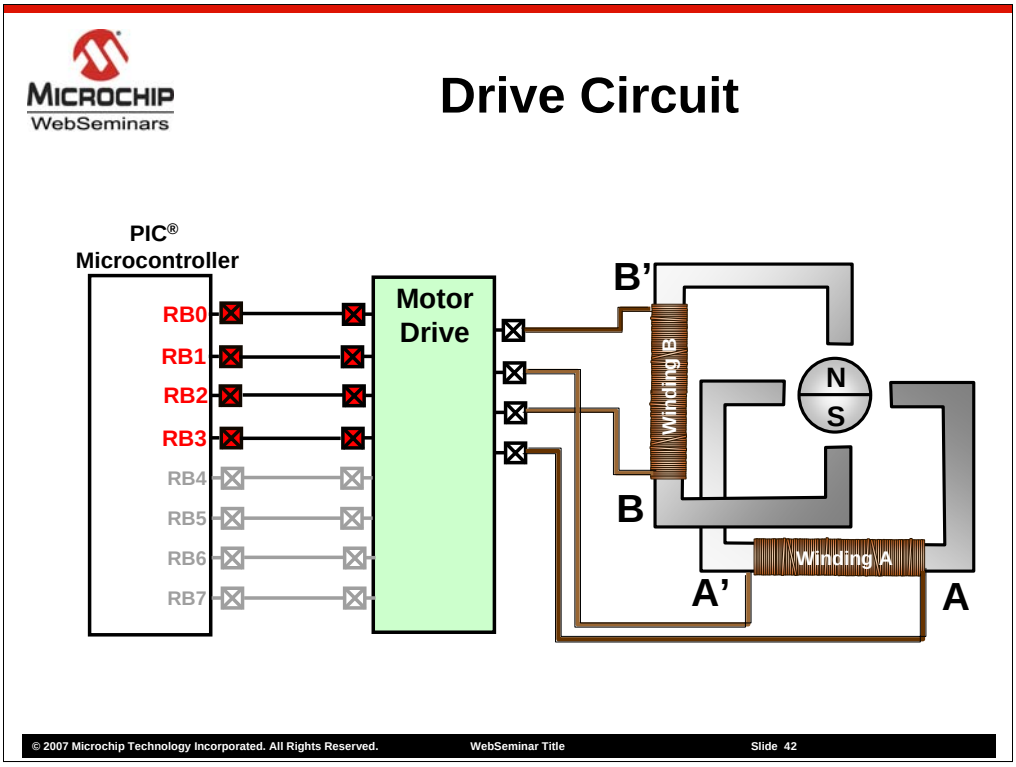
There are some things we will need to keep in mind when utilizing half-stepping algorithms.

RPM will slow to half of what it was in full-stepping algorithms. This means that TMR0 interrupts will need to occur twice as fast. Also, since half the time only one winding is energized, torque will be dramatically reduced in Half-Stepping as compared to Two Phase On full-stepping.

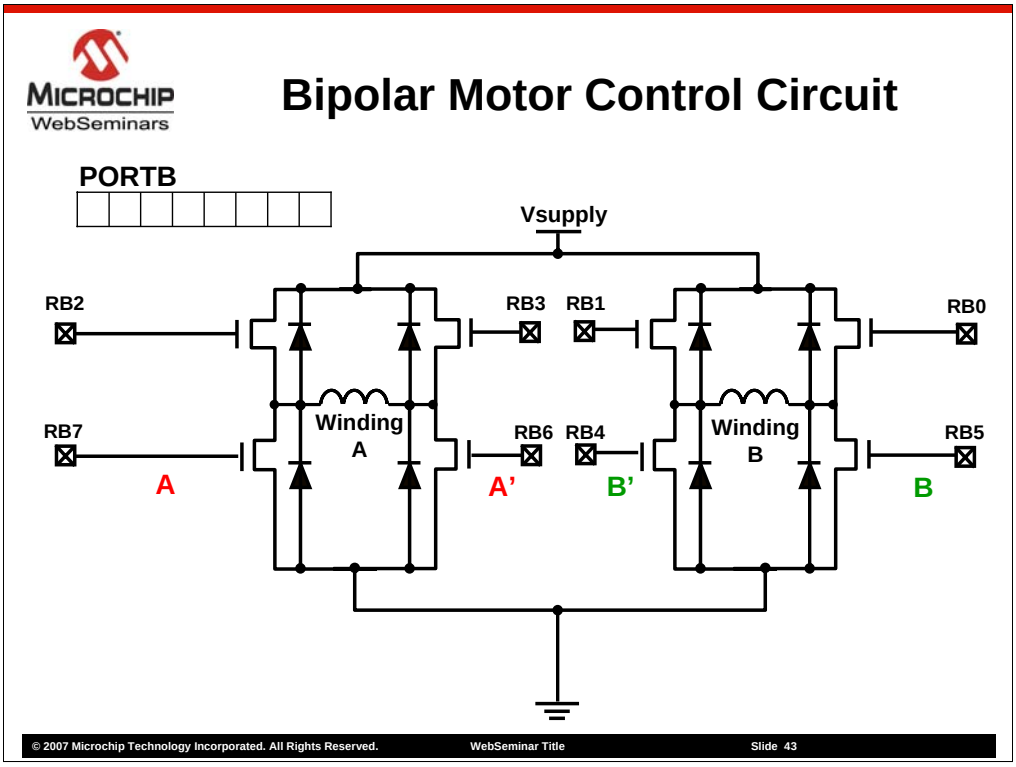
If we can't live with the decreased torque, we will need to move to a larger motor with more full steps and increase the cost of our circuit.



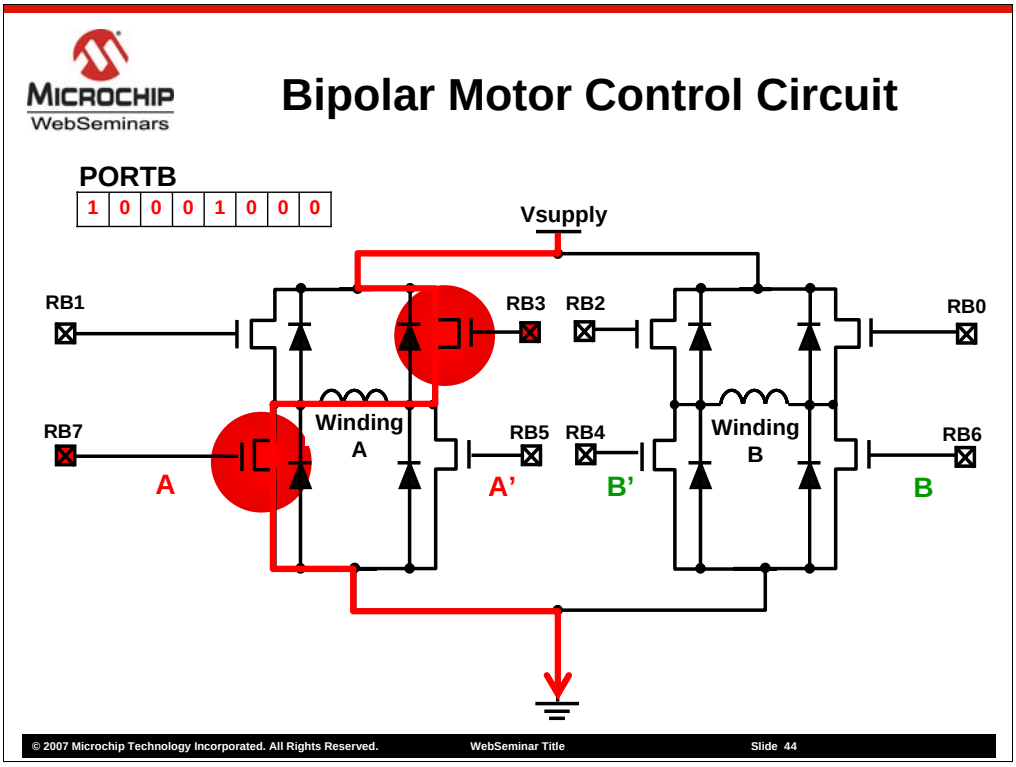
Let’s move away from stepping algorithms and discuss drive circuits.



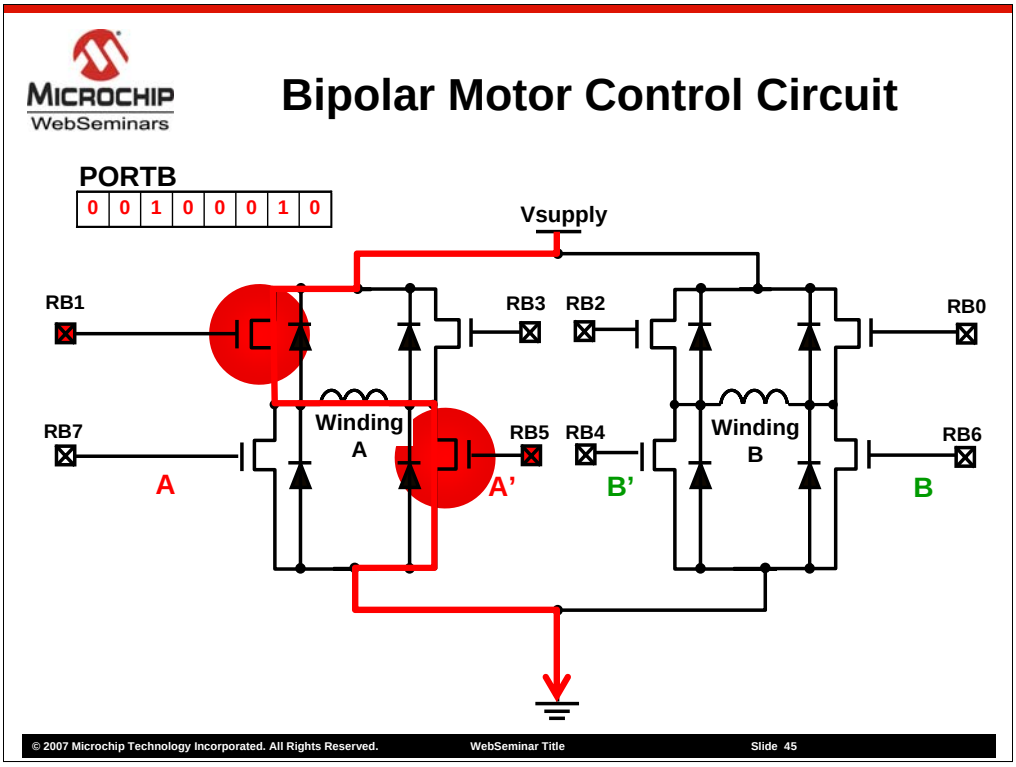
The lower 4-bits of the PORTB peripheral are used in conjunction with the Motor Drive Circuit to control current flow through the windings.



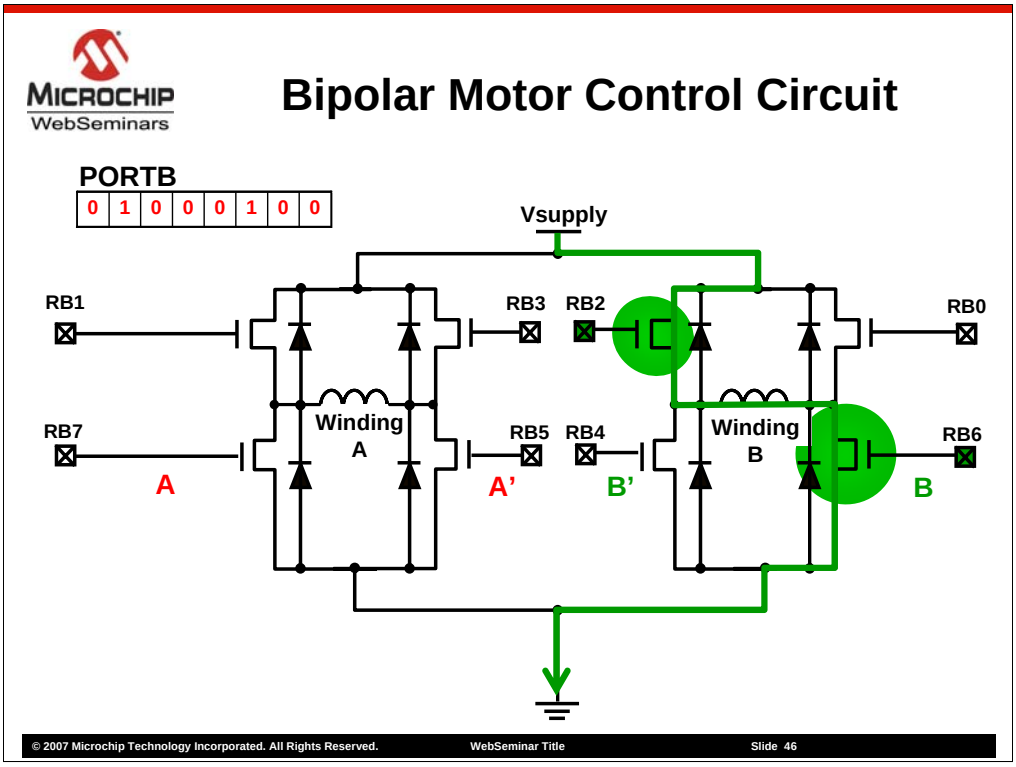
Looking inside the black box, a stepping motor drive circuit is created using two H-Bridges. Each H-bridge consists of four MOSFET transistor that will act as switching mechanisms. Protection diodes are used to avoid damage to MOSFETS as a result of Voltage Spikes produced by the collapse of the Magnetic field around each winding once current is removed. The lower half of each H-bridge MOSFET gates connect to the upper 4-bits of the PORTB register in this example. The upper half of each H-bridge connects to the lower 4-bits. Each winding uses its own H-bridge. Again, winding leads are identified using the nomenclature used throughout this presentations.



To initiate current flow in a particular direction through each winding two MOSFETS will need to be turned on. For example to create a right to left current direction in Winding A, PORTB bits 7 and 3 are driven HIGH turning on the MOSFETS connected to their associated pins. Current now flows through the coil.

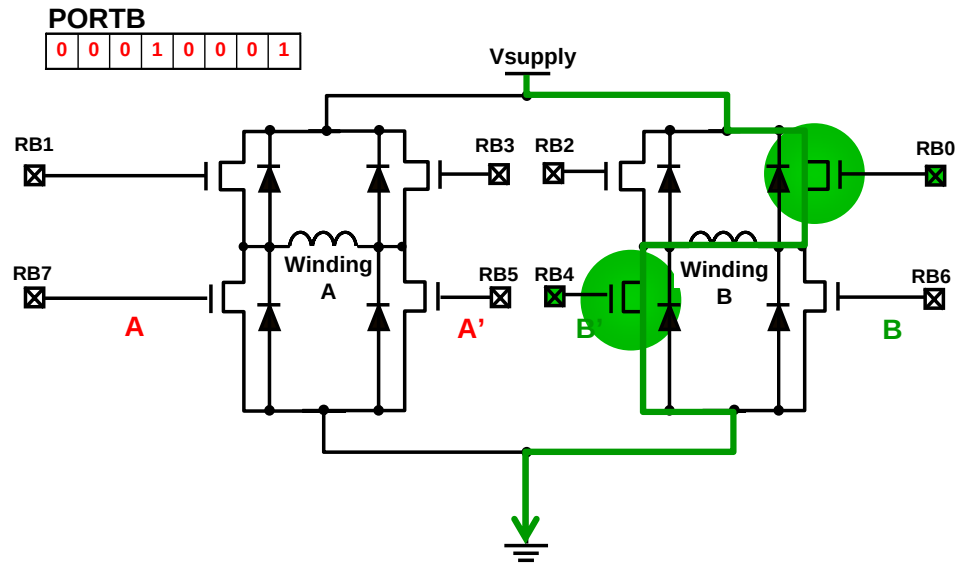


To change direction through winding A we need only change the MOSFETS activated. In this example, MOSFET gates connected to PORTB pins 5 and 1 are driven HIGH and current flows from left to right across the winding.

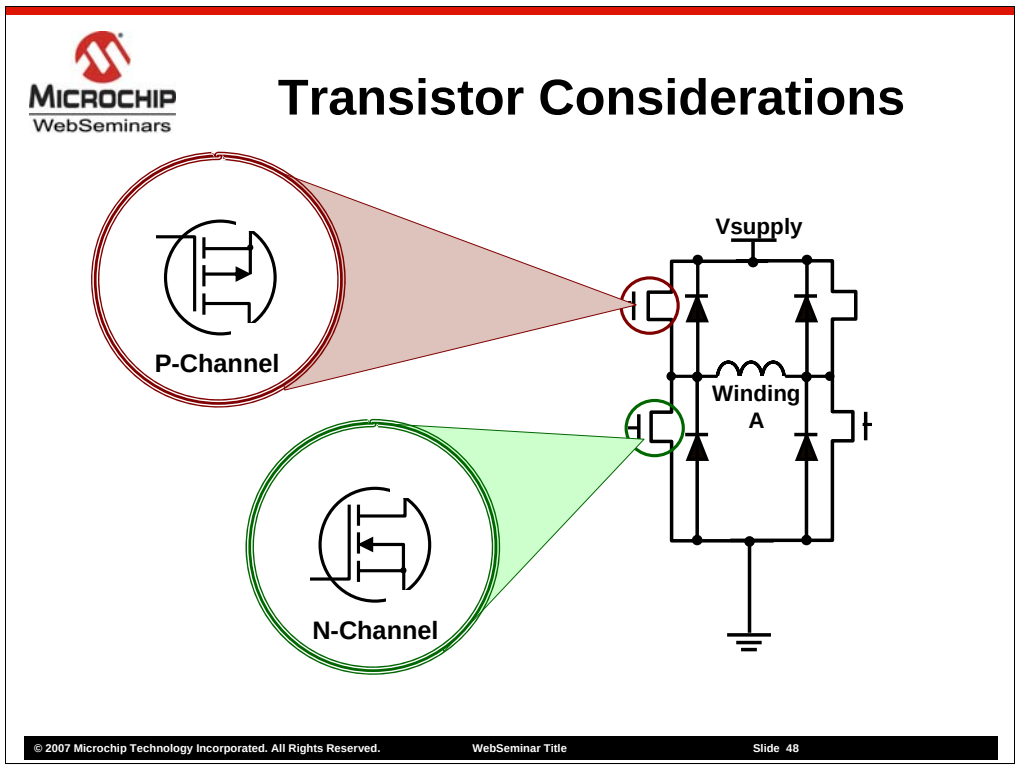


Winding B direction is controlled in the same fashion.

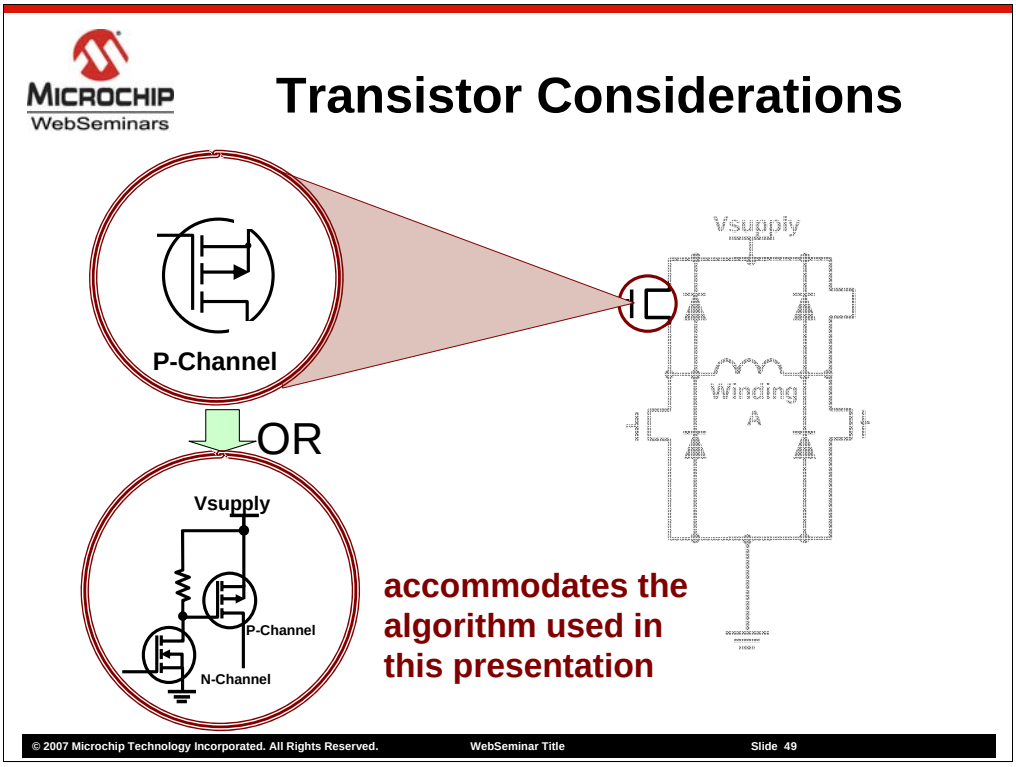
Bipolar Motor Control Circuit



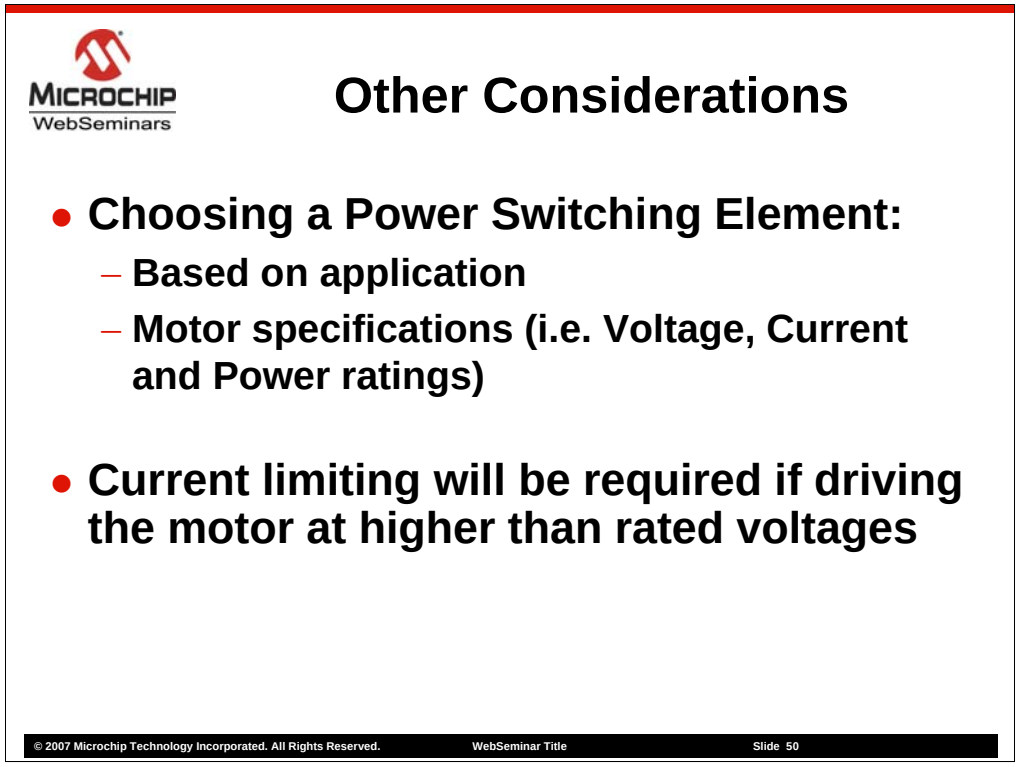
© 2007 Microchip Technology Incorporated. All Rights Reserved.



In the preceding drive circuit example, enhancement type MOSFETS are used. Any transistor could potentially be used such as Bipolar and IGBT transistors. However, MOSFETs are easier to control since they are voltage controlled devices. MOSFETs also offer faster switching times than the IGBT thereby reducing switching power loss. In the above example you'll notice that the MOSFETs in the upper half of the H-Bridge are P-channel type and the lower half of the H-Bridge are N-channel type. The P-channel MOSFET provides the pull-up, or charge current for the gate capacitance and the N-channel MOSFET provides the pull-down or discharge current for the external gate capacitance.



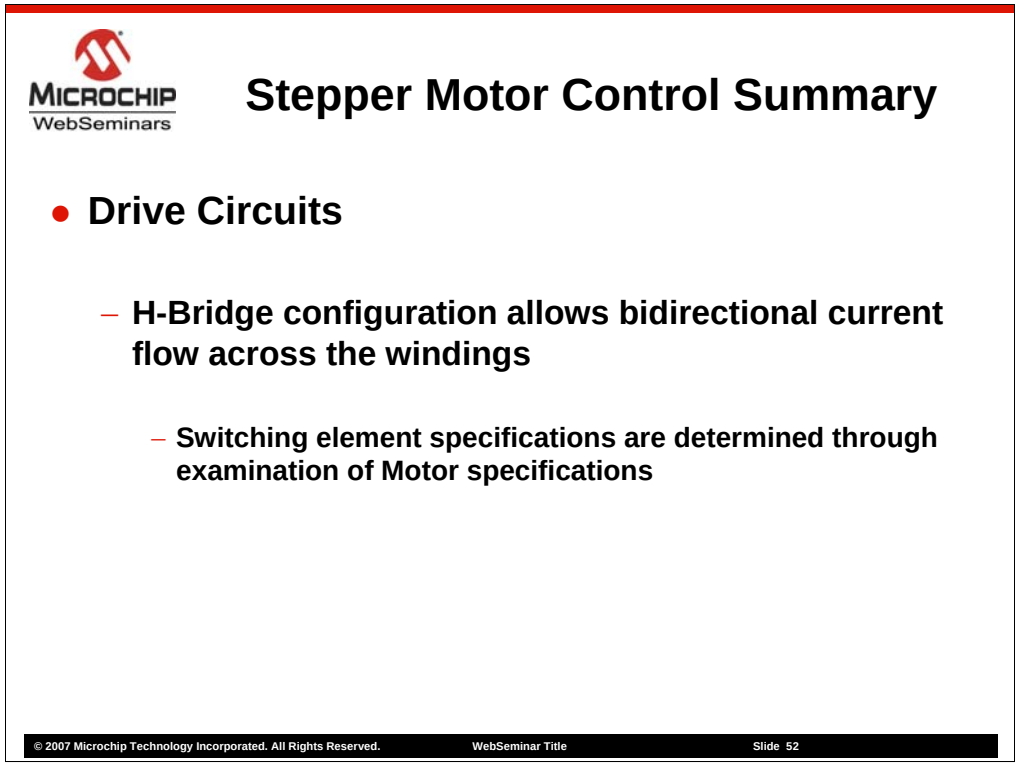
In this presentation however, we have been activating the upper half of the H-Bridge MOSFET gates with a positive voltage or logic HIGH. If you wish to utilize this algorithm, you may consider applying an inverter configuration to the gate of the P-channel type MOSFETs as shown above using an addition N-channel type MOSFET.



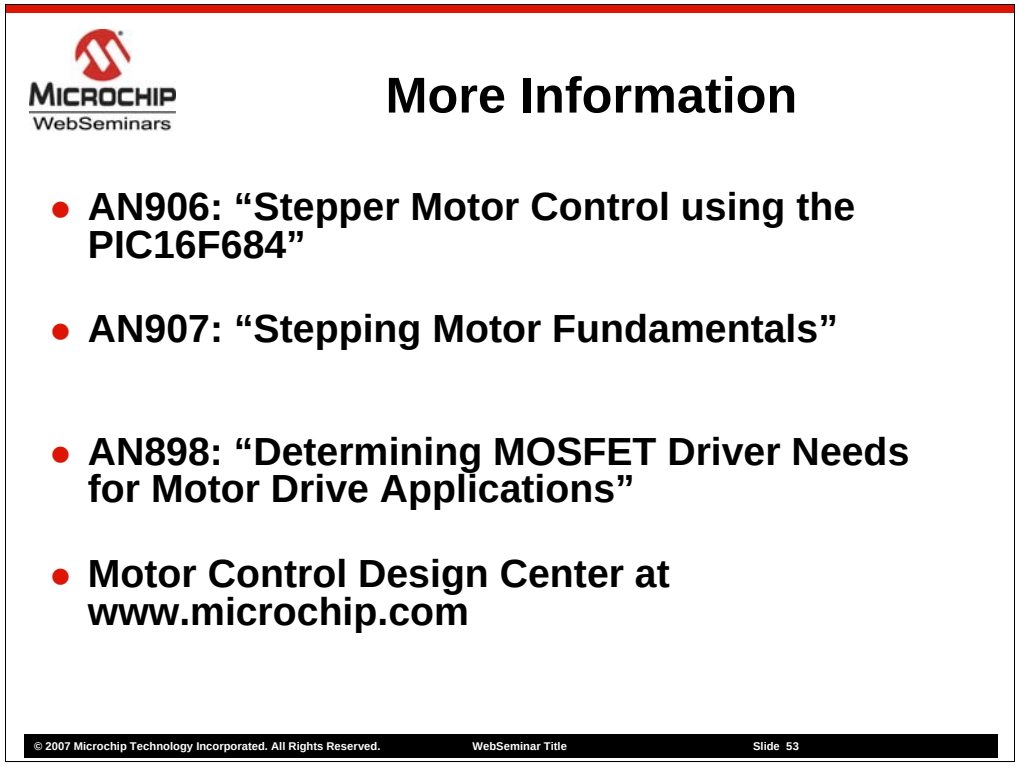
Some other things to consider are the ratings of the MOSFET switching elements used. As always, your application will dictate much of this. Attention to the specifications for the particular motor you are using will help here paying specific attention to Current and Power ratings. Often stepper motors are driven at higher voltages than listed in their specifications in order to reduce current rise slew rates within the coil to allow for higher step rates. However, in driving the motor at these higher voltage levels, current limiting practices will need to be implemented to avoid damaging the motor.

Summarizing full and half stepping modes. We have two options available to us when using Bipolar stepper motors to rotate the motor in full-stepping algorithms. One Phase On energizes one winding at a time while Two Phase On energizes two windings at once. Two Phase On improves the torque of the motor but remember this type of full-stepping will also coil temperature due to power dissipation.

Half-Stepping improves step angle resolution while minimizing the effects of anti-resonance. However, nothing is free, torque is reduced by half of Two Phase On full stepping and steps per second will need to executed twice as fast.



In the examples used in this Web Seminar a traditional H-Bridge configuration is used to drive the stepper motor. You could build your own using some power transistors or utilize on of the many IC packages available on the market. As always when selecting components to interface with your motor always refer to the Stepper Motors specification sheet.



For more information on topics covered in this web seminar or for further information please refer to application notes listed above. AN898 in specific goes into greater detail on various switching components and why you would use one over the other. You may also be interested in visiting the Motor Control Design Center at www.microchip.com for recommended products, application notes and technical briefs related to Motor Control.

My name is Marc McComb and I thank you for downloading this web seminar.

Trademarks

The Microchip name and logo, the Microchip logo, Accuron, dsPIC, KeeLoq, KeeLoq logo, microID, MPLAB, PIC, PICmicro, PICSTART, PRO MATE, rfPIC and SmartShunt are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

AmpLab, FilterLab, Linear Active Thermistor, Migratable Memory, MXDEV, MXLAB, SEEVAL, SmartSensor and The Embedded Control Solutions Company are registered trademarks of Microchip Technology Incorporated in the U.S.A.

Analog-for-the-Digital Age, Application Maestro, CodeGuard, dsPICDEM, dsPICDEM.net, dsPICworks, dsSPEAK, ECAN, ECONOMONITOR, FanSense, FlexROM, fuzzyLAB, In-Circuit Serial Programming, ICSP, ICEPIC, Mindi, MiWi, MPASM, MPLAB Certified logo, MPLIB, MPLINK, PICkit, PICDEM, PICDEM.net, PICLAB, PICtail, PowerCal, PowerInfo, PowerMate, PowerTool, REAL ICE, rfLAB, Select Mode, Smart Serial, SmartTel, Total Endurance, UNI/O, WiperLock and ZENA are trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

SQTP is a service mark of Microchip Technology Incorporated in the U.S.A.

© 2007 Microchip Technology Incorporated. All Rights Reserved.