DC Motors II

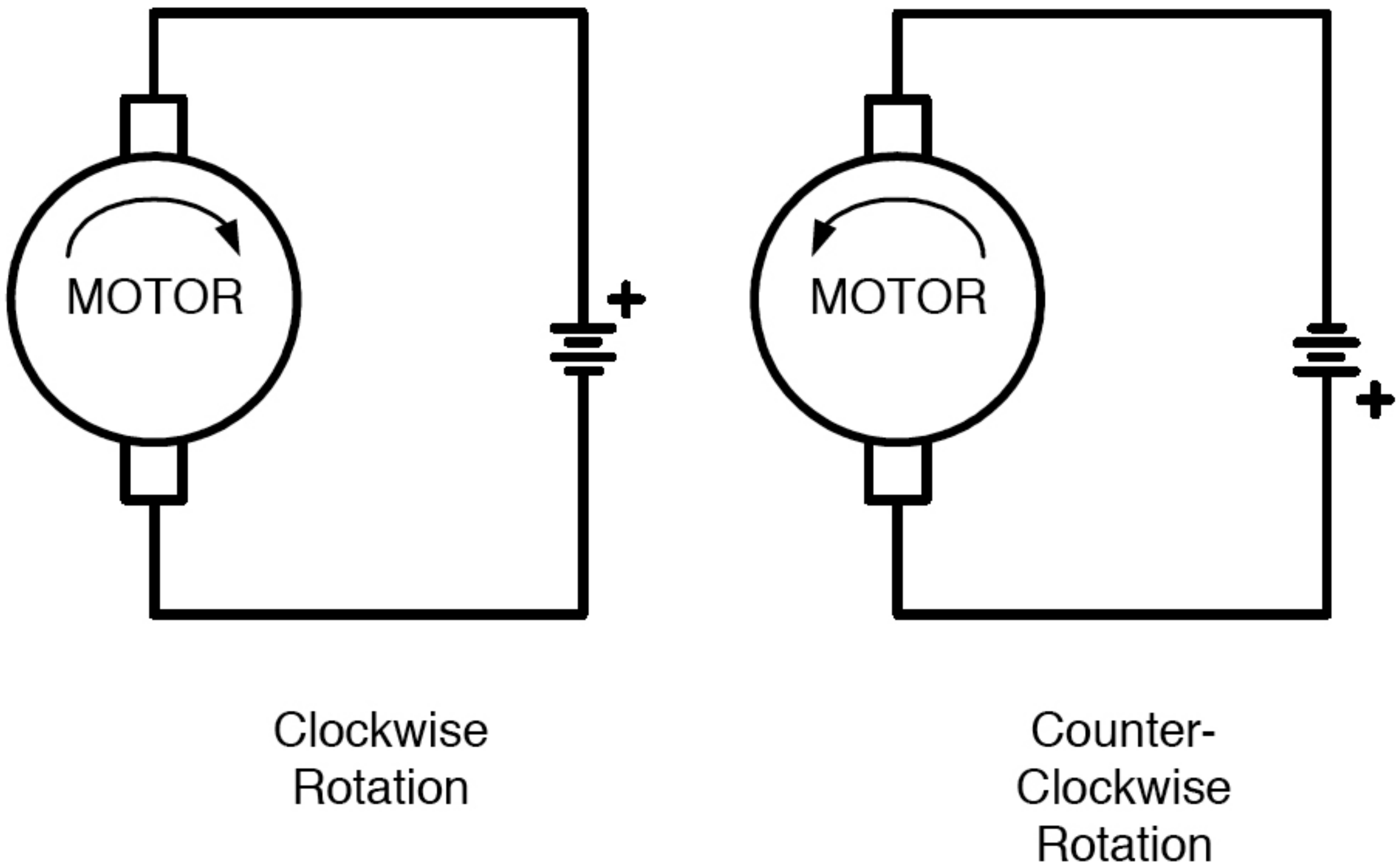
ECE 3710

Most of the arguments to which I am party fall somewhat short of being impressive, owing to the fact that neither I nor my opponent knows what we are talking about.

- Rodney Dangerfield

dc motor

direction:



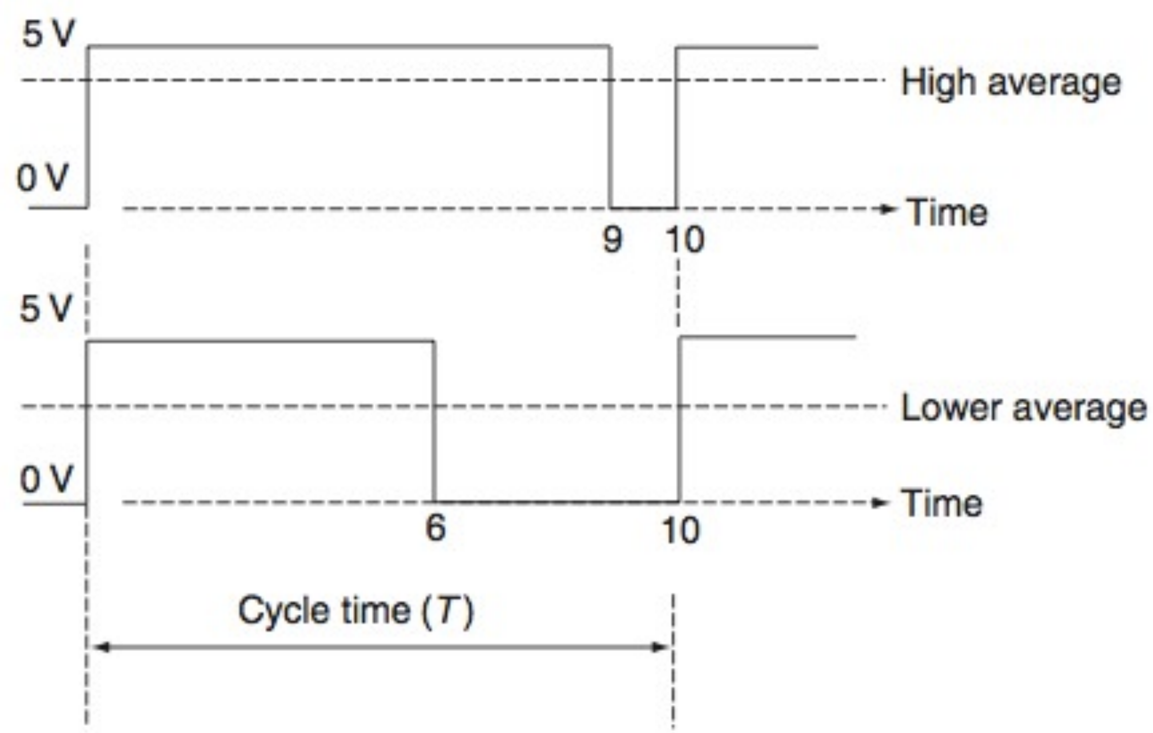
speed:

~ Voltage ← ideal

(current sourced and load)

dc motor: speed

pulse-width modulation:



'on' time

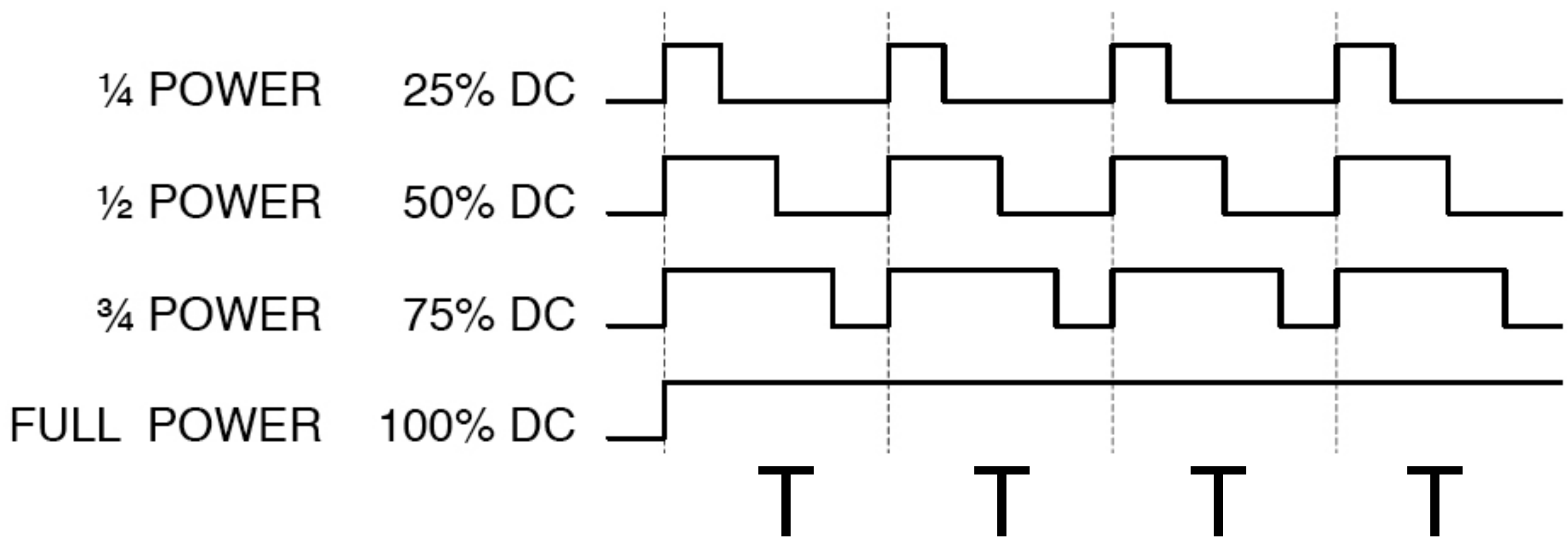
5 V * (9/10) = 4.5 V

5 V * (6/10) = 3.0 V

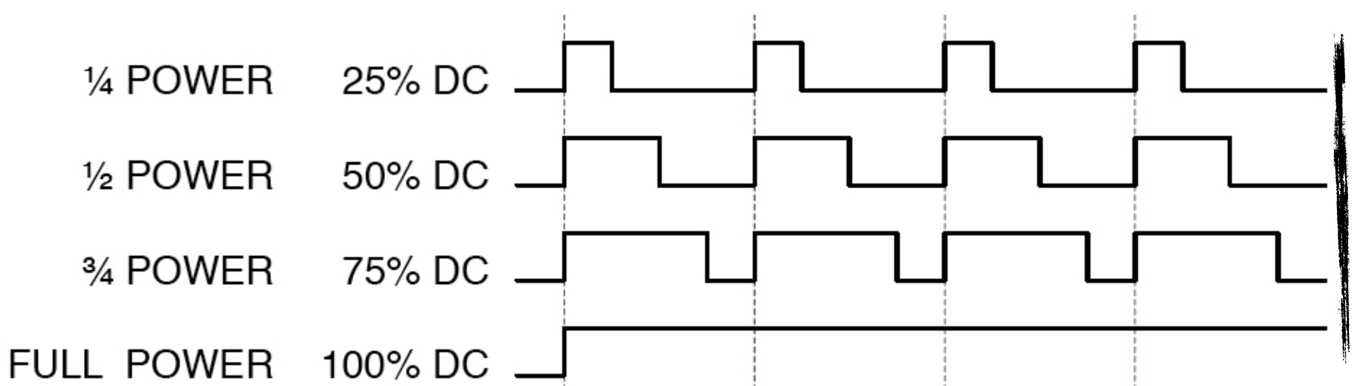
aka duty cycle

dc motor: speed

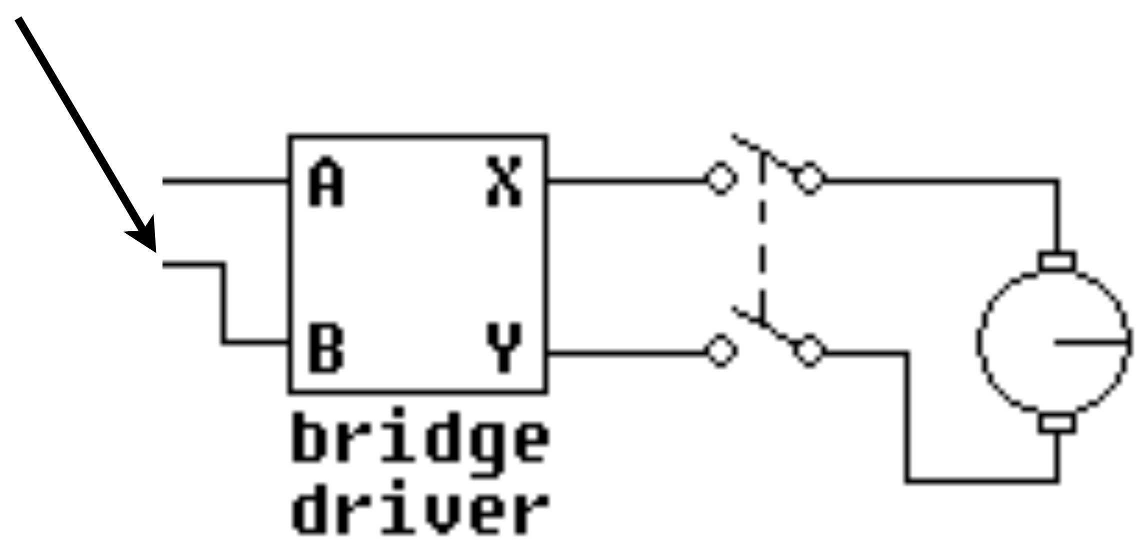
if T << 1 ⟵ motor just sees average



dc motor: speed
(PWM)



PWM signals serve as input to driver



A	B	motor
0	0	stop
0	1	forward
1	0	reverse
1	1	stop

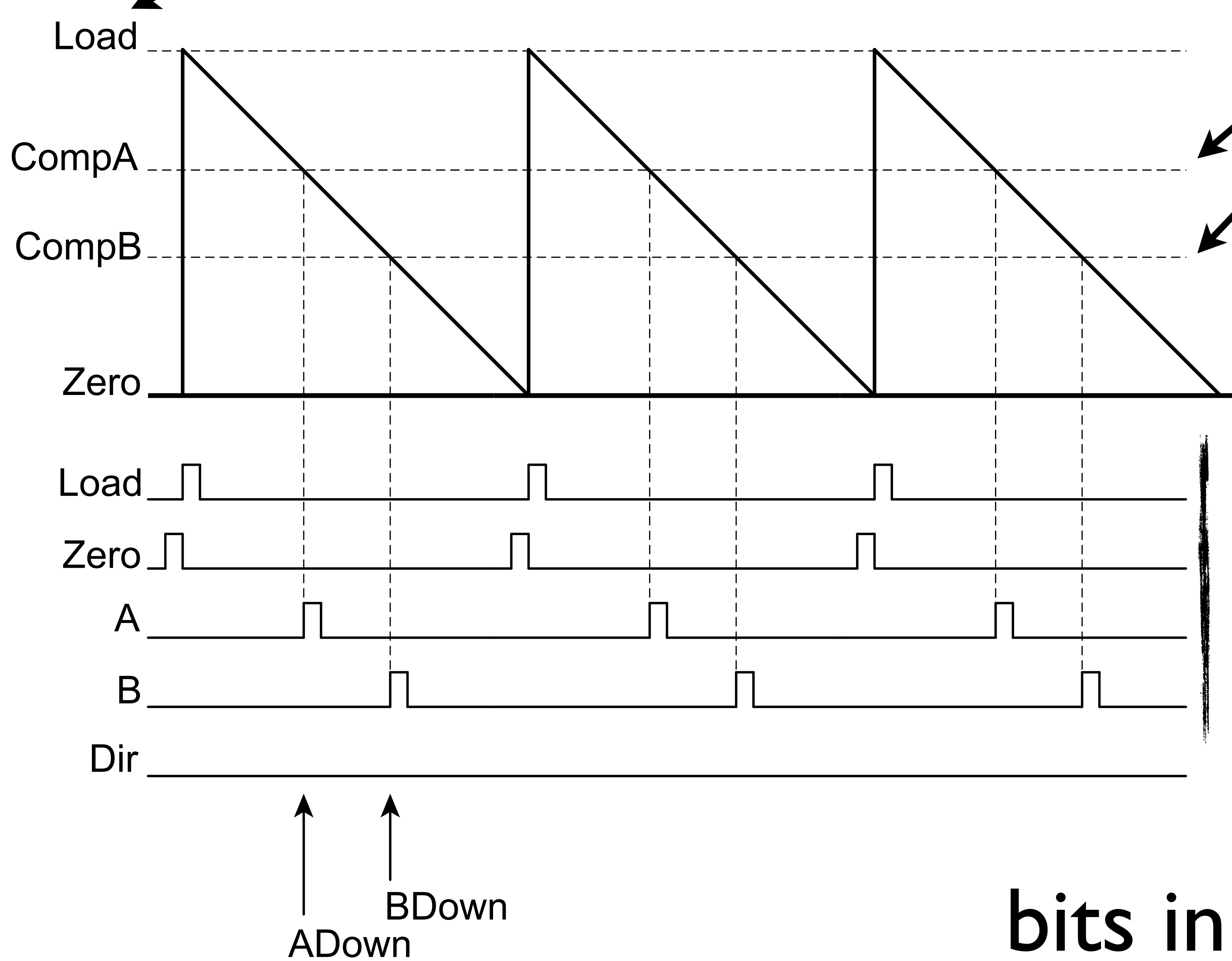
note: signal can't change too fast for driver

(look at driver switching speed)

LM3S1968 PWM countdown mode

initial timer value

comparator values
(also 16-bit)



when COUNT=XX one-clock cycle high
(trigger some action; e.g. interrupt)

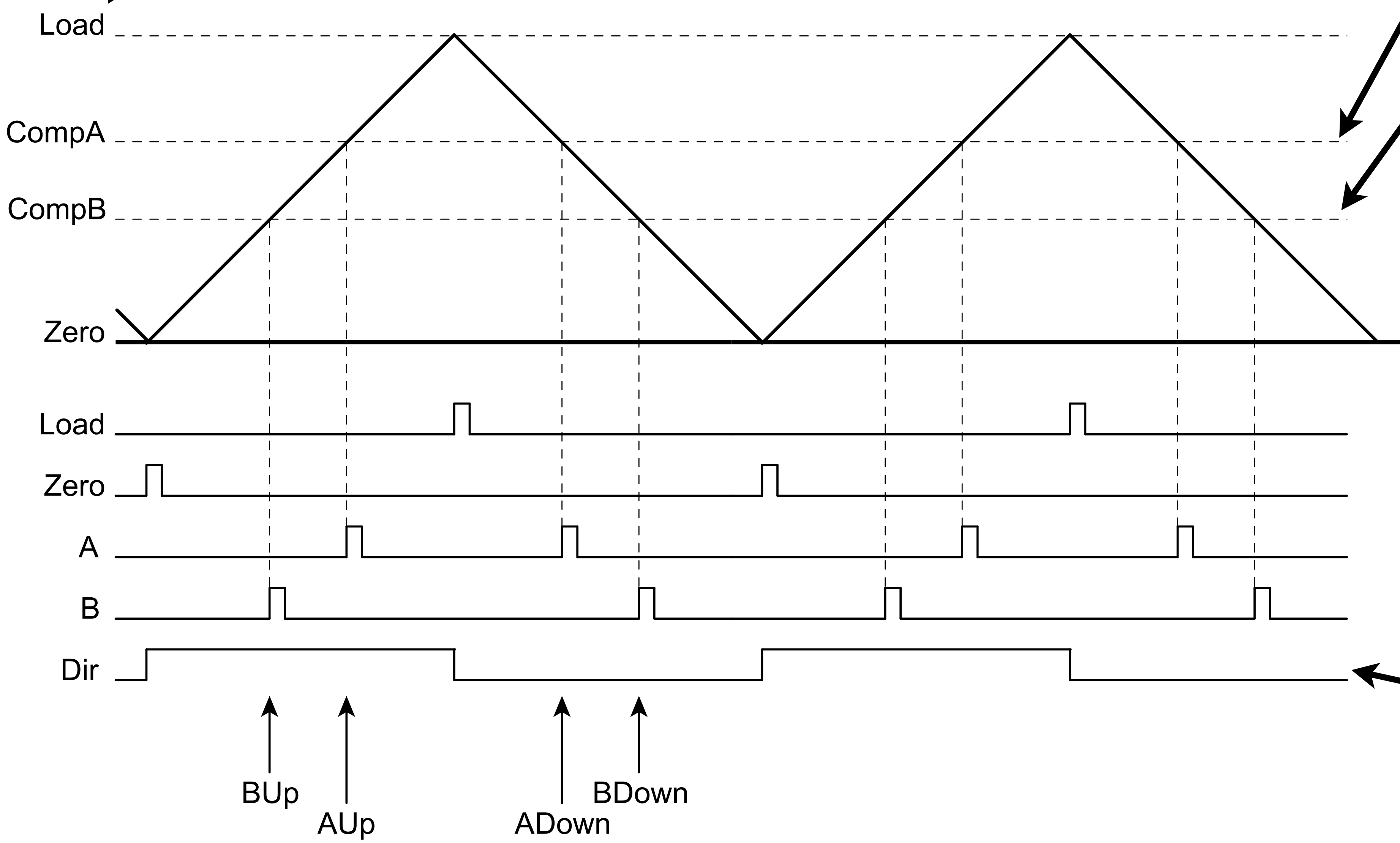
bits in `PWMxRIS`

interrupts: zero,load,match a down, match b down

LM3S1968 PWM count up-then-down mode

initial timer value

comparator values (also 16-bit)



can figure out if pwm timer counting up or down

interrupts: zero,load,match a up/down, match b up/down

LM3S1968 PWM events

what should happen
when COUNT=XX

bits in `PWMxGENy`:

0x0	Do nothing.
0x1	Invert the output signal.
0x2	Set the output signal to 0.
0x3	Set the output signal to 1.

PWM0 Generator A Control (PWM0GENA)

Base 0x4002.8000
Offset 0x060
Type R/W, reset 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	reserved															
Type	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	reserved				ActCmpBD		ActCmpBU		ActCmpAD		ActCmpAU		ActLoad		ActZero	
Type	RO	RO	RO	RO	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

COUNT=

CompB, down

CompB, up

CompA, down

CompA, up

LOAD

ZERO

match while counting

note: each generator (A/B) can use both comparators

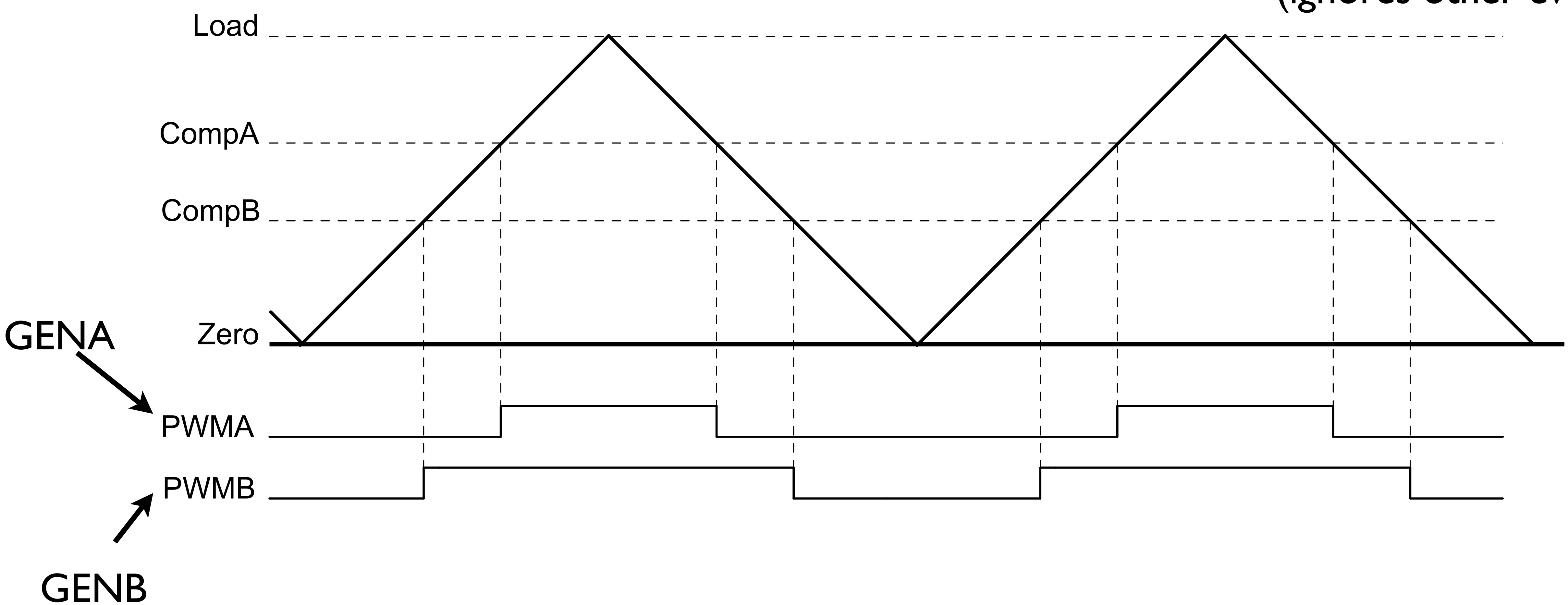
LM3S1968 PWM events

PWMA set to invert outputs when COUNT=CompAU/D

(ignores other events)

PWMB set to invert outputs when COUNT=CompBU/D

(ignores other events)



ex: 25% DC w/25 KHz period

reset every

$$T = \frac{1}{f} = \frac{1}{25 \times 10^3}$$

strategy:

1. trigger event at (re)load periodic timer

(event sets output high)

2. trigger event at 25% of timer period

(event sets output low)

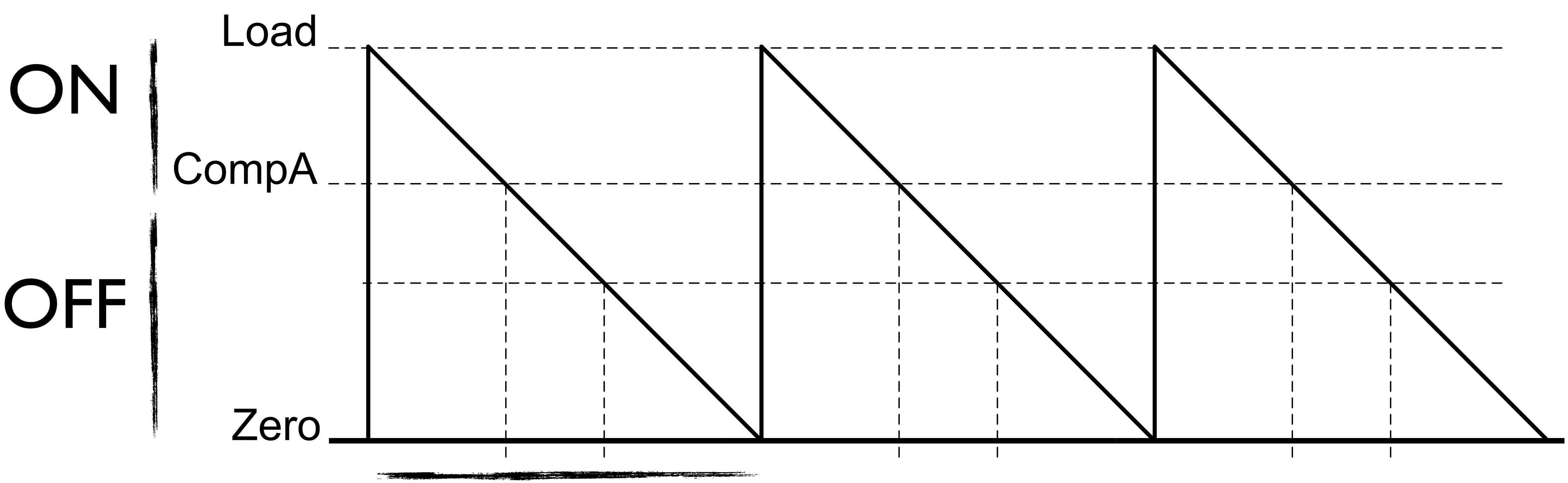
3. goto 1.

$$T_{25\%} = T - 0.25 \times T = T \times (1 - 0.25)$$

LM3S1968 PWM0, GENA configuration

ex: 25% DC w/25 KHz period:

(PWMClk = SysClk = 12 MHz)



T=1/25e3 = 40 us

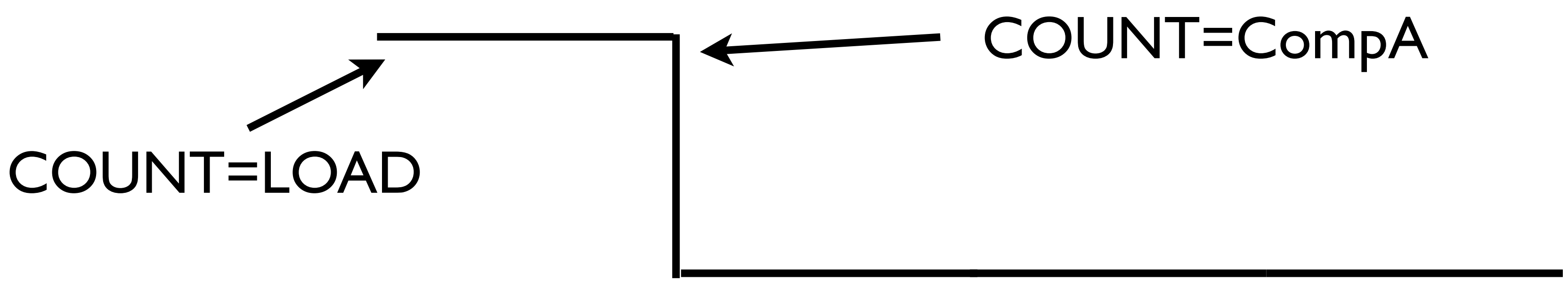
(time for timer to expire)

note: only need a single comparator

LM3S1968 PWM0, GENA configuration

ex: 25% DC w/25 KHz period:

(PWMClk = SysClk = 12 MHz)



T=1/25e3 = 40 us

(time for timer to expire)

bits in `PWMxGENy`:

0x0	Do nothing.
0x1	Invert the output signal.
0x2	Set the output signal to 0.
0x3	Set the output signal to 1.

when:

a. COUNT=LOAD, output =1

b. COUNT= CompA, output = 0

LM3S1968 PWM0, GENA configuration

timer and comparator values:

clock cycles to
get T=40 us

$$\text{TICKS} = \frac{1/f}{\frac{1}{\text{PWMClk}}} = \frac{1/25e3}{1/12e6} = 480$$

remember, just 16-bit timer

LOAD = TICKS - 1 = 479

CompA = TICKS*(1-DC%) - 1
= 480*(1 - 0.25) - 1 = 359

counting down,
want to be off
for 1-DC% of time

stepper motor:

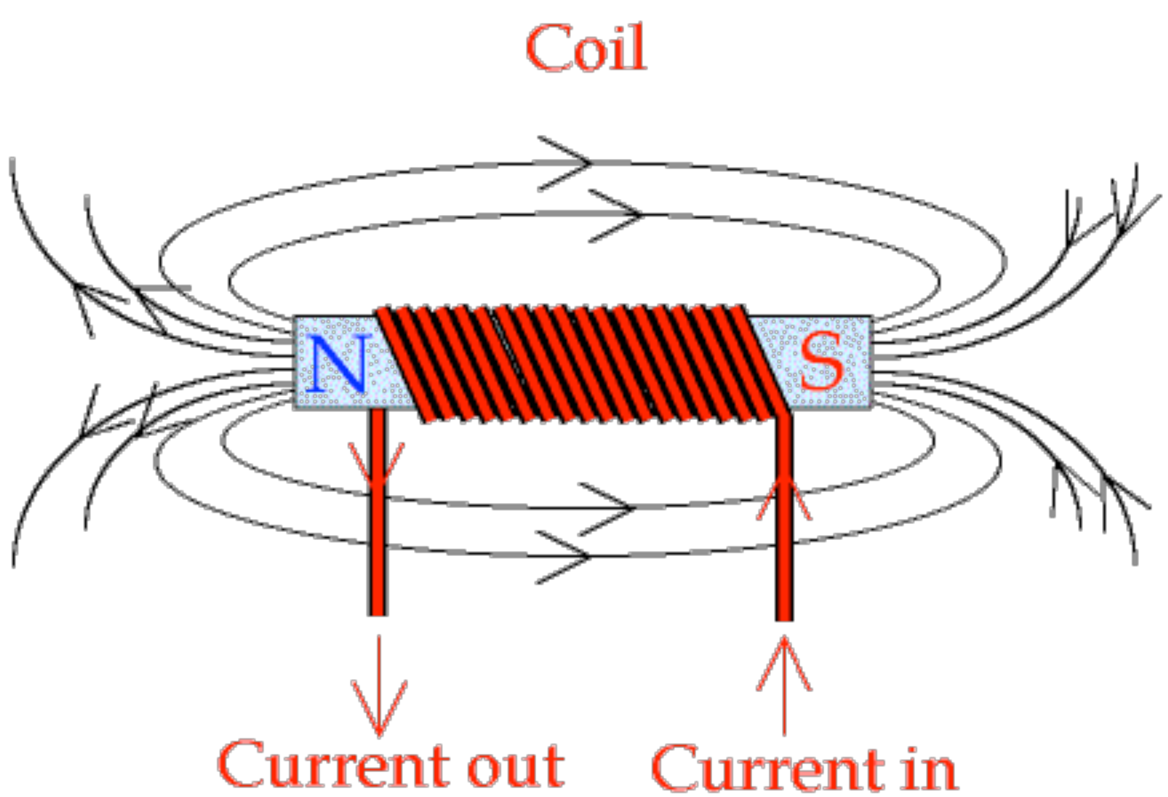
more precise control over rpm

and partial revs.

position control

from Ampere's law

current causes B-field

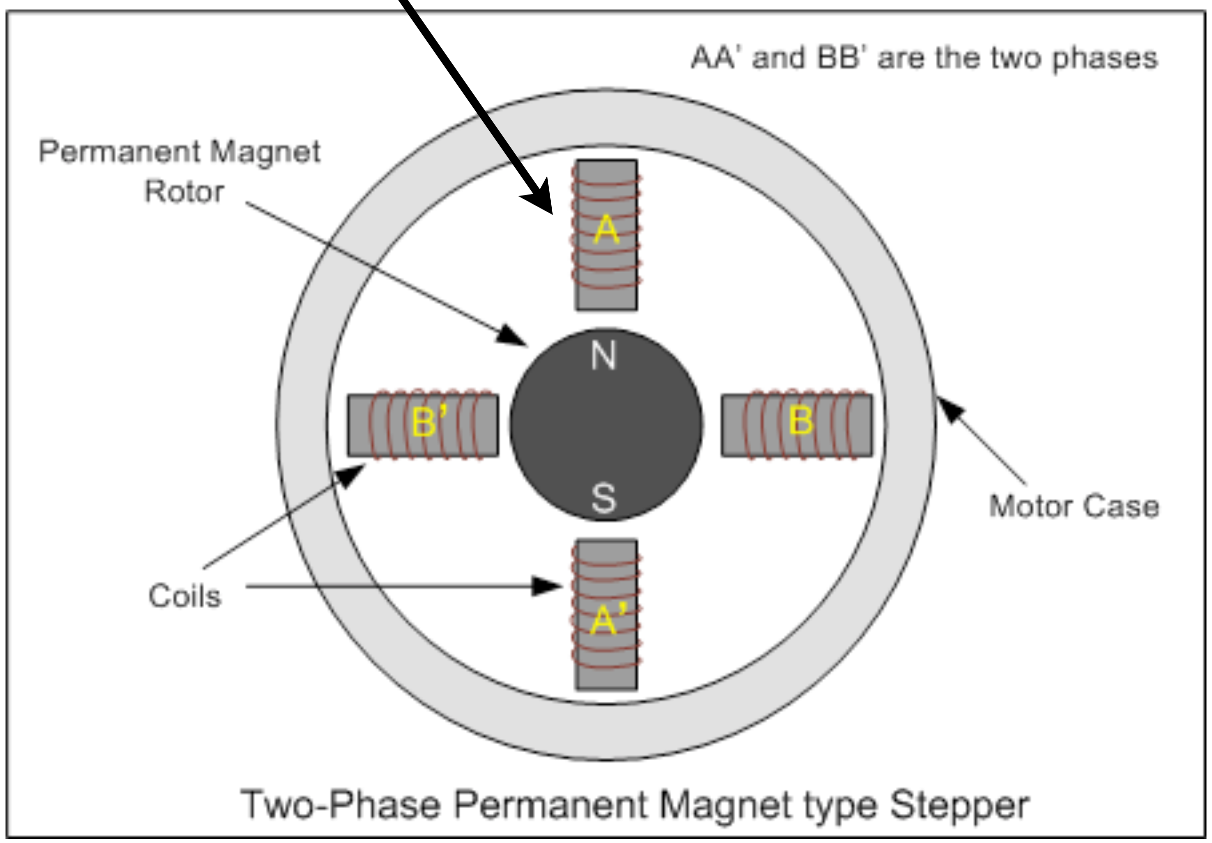


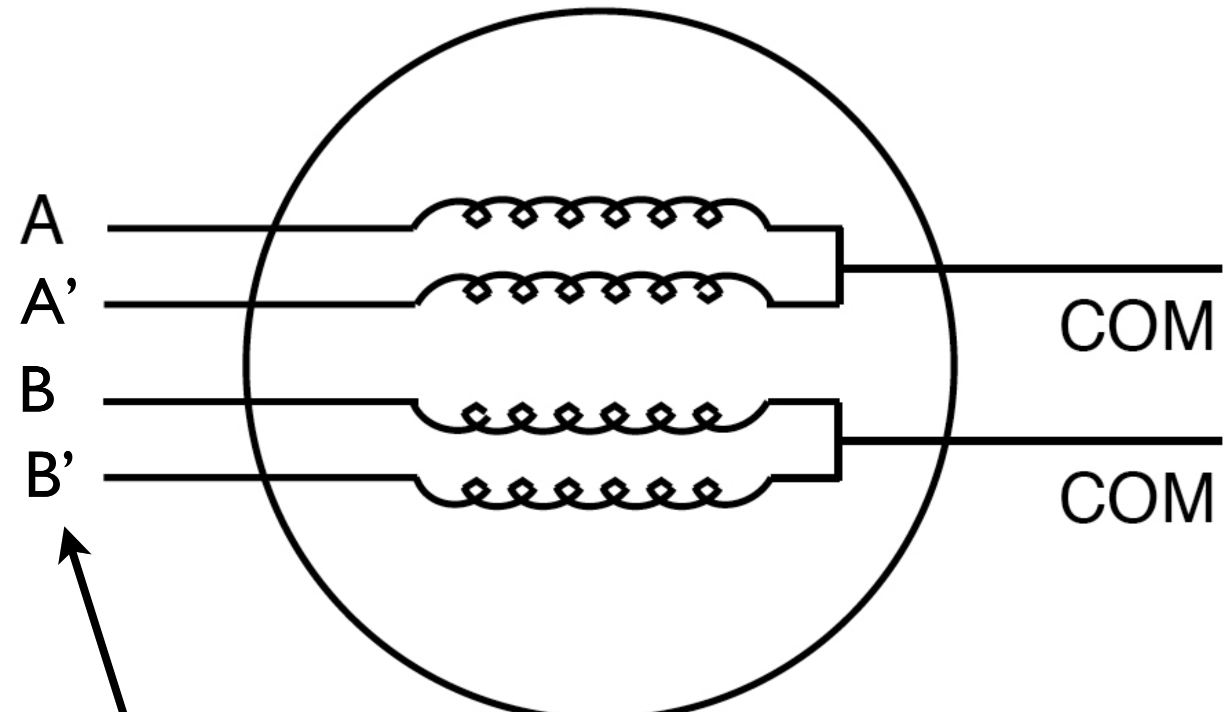
an electromagnet

stepper motor: permanent magnet

stator

both wrapped around





connect switches to coils (ABA'B')

note: labels refer to coil ends, not strators

1=closed
0=open

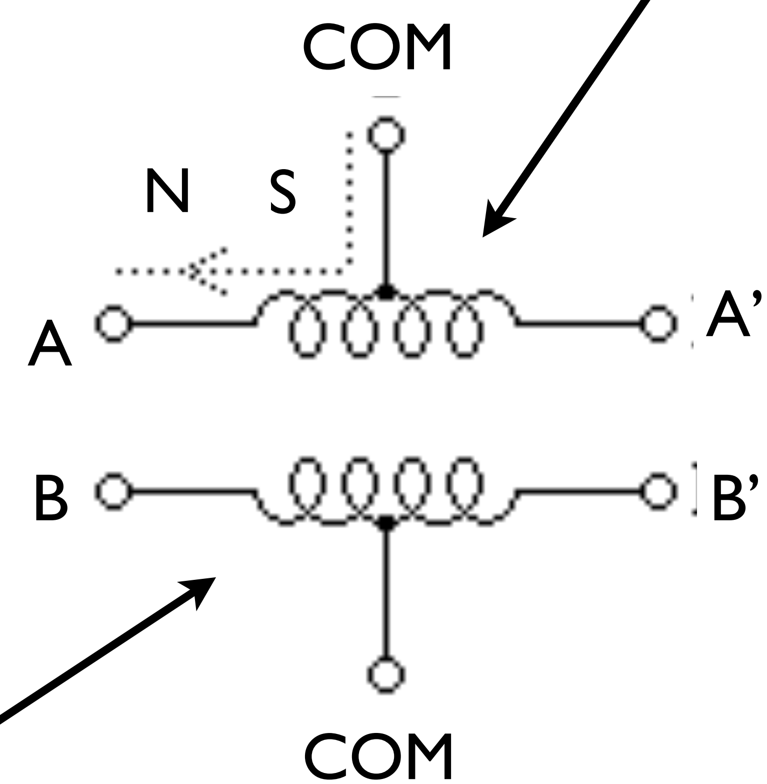
stepper motor: permanent magnet

remember, two coils

switches determines direction of current flow:

1 =

0=



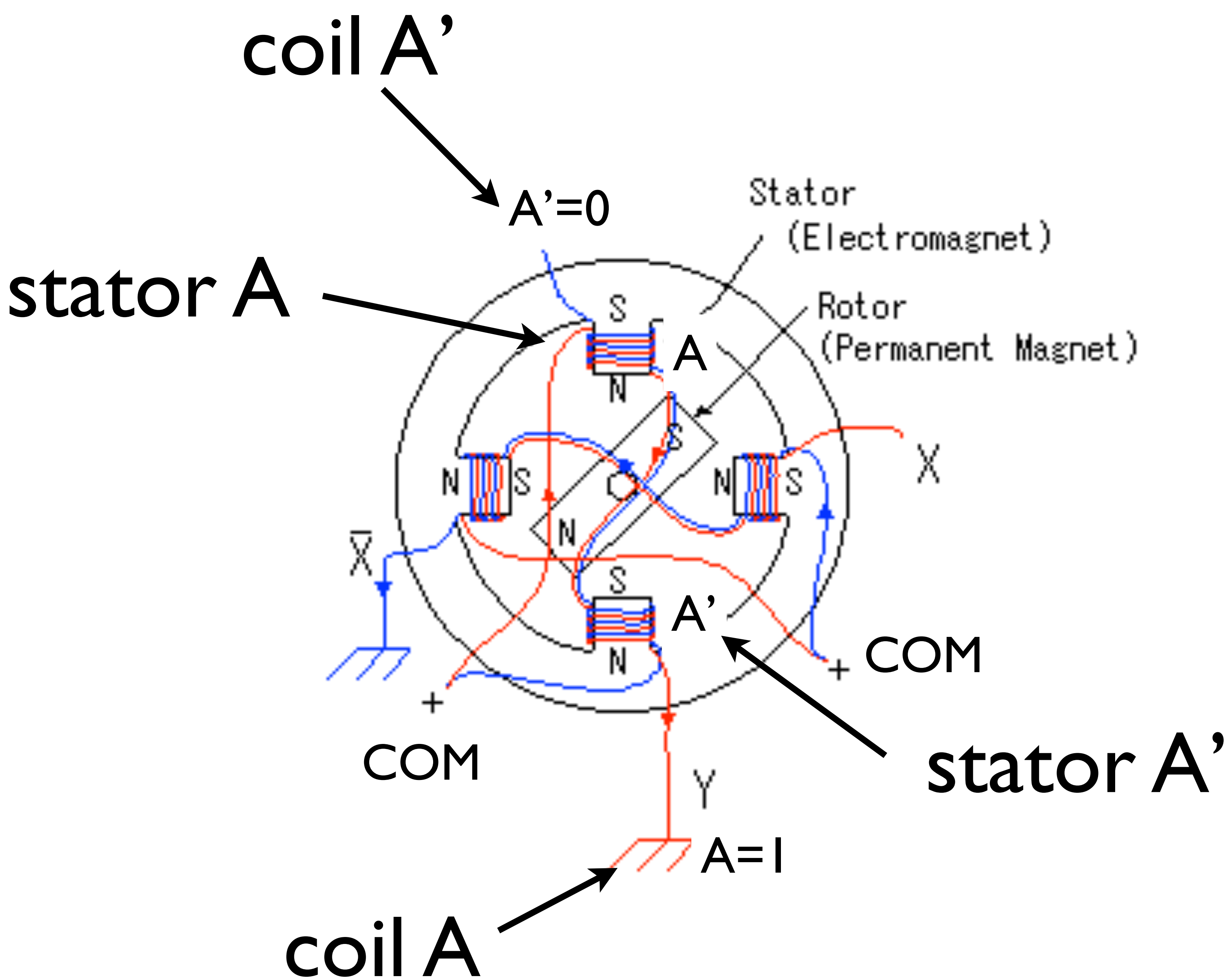
=0

=0

each coil wound around both stators

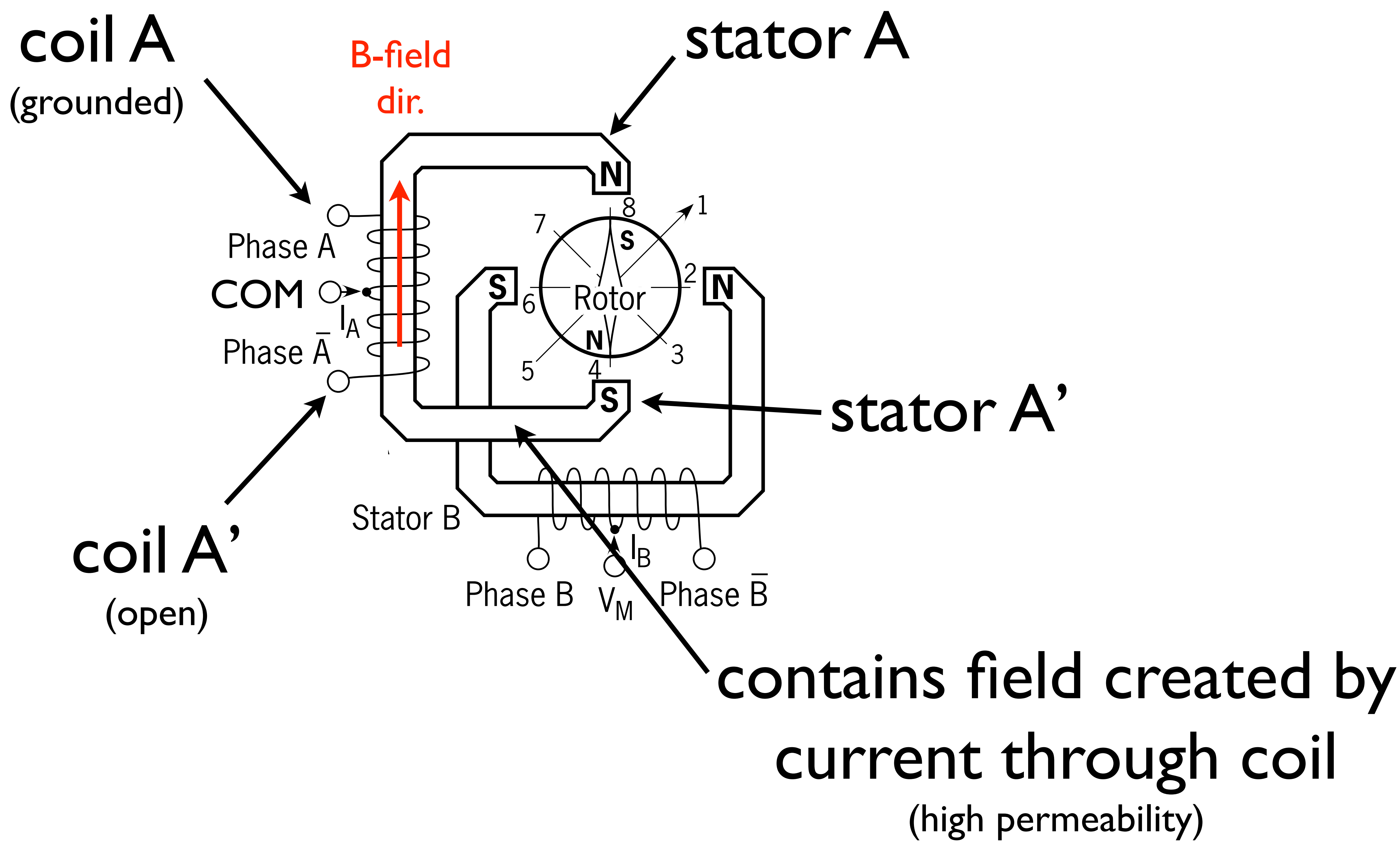
center tap on transformer

each coil wound around both stators



note: this is confusing but necessary to make it work with book

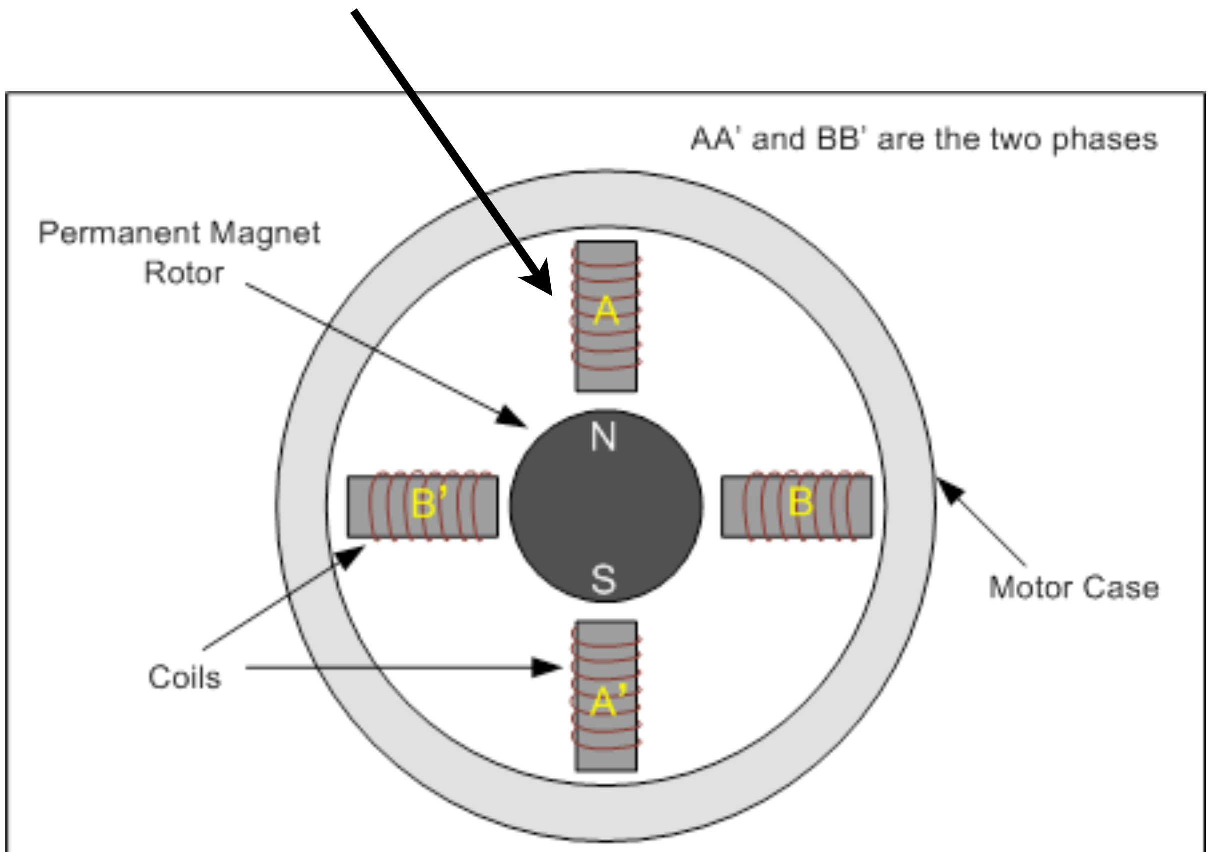
more realistic configuration



note: A=1 and A’=0

stepper motor: permanent magnet

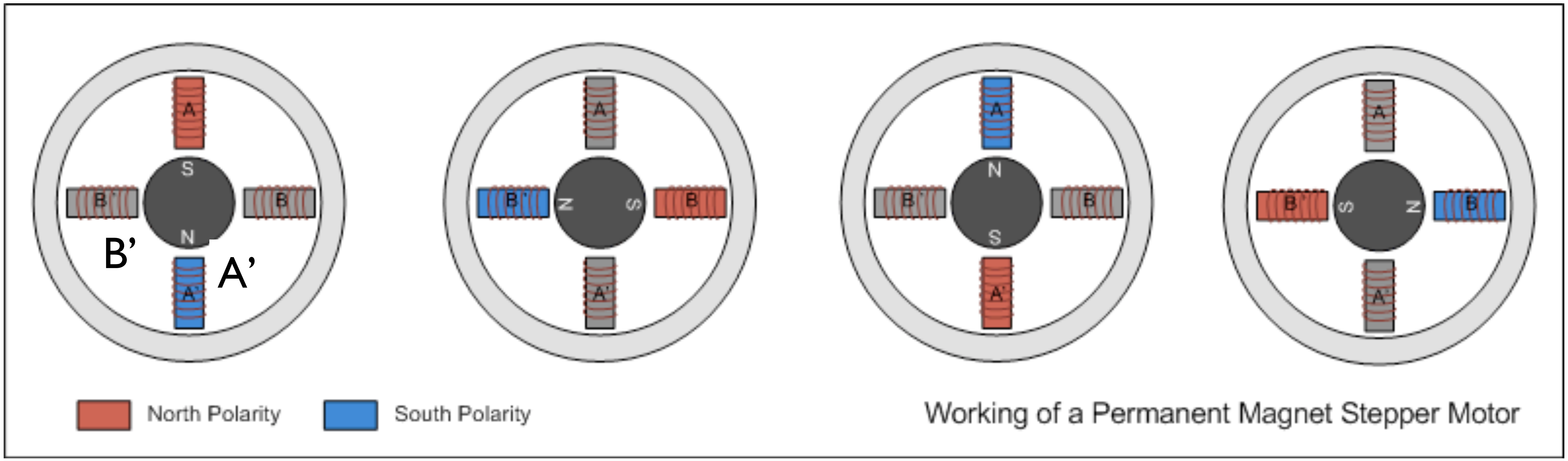
stator



Two-Phase Permanent Magnet type Stepper

Q: how to move rotor by changing magnetic polarity of stators?

stepper motor: abstracted view



A=1 A'=0

B=1 B'=0

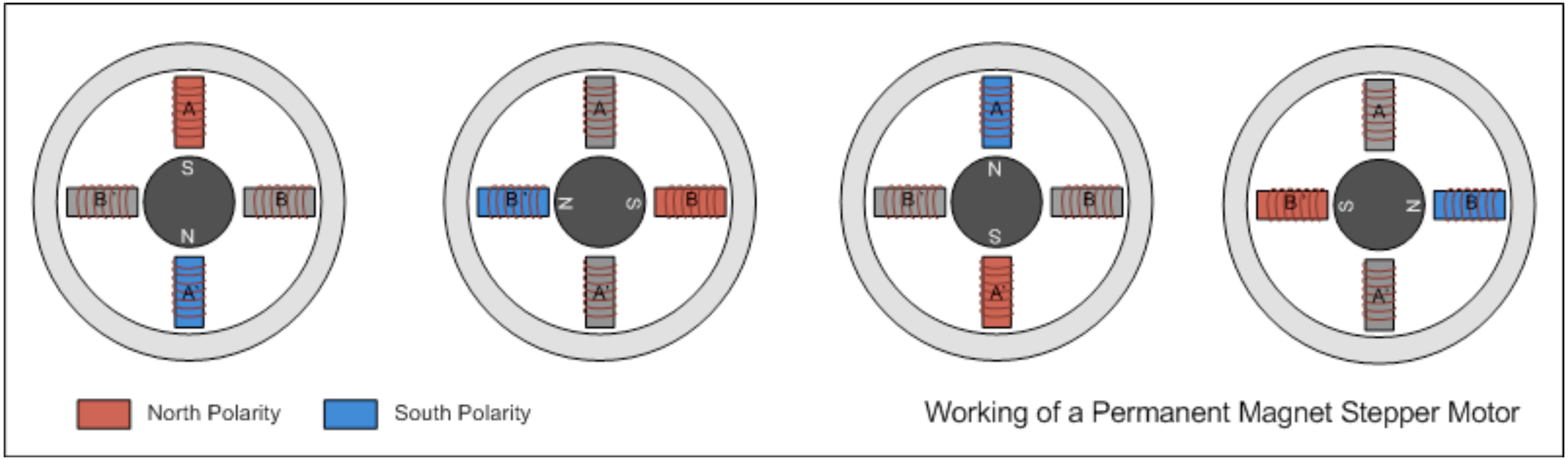
A=0 A'=1

B=0 B'=1

just know:

switch configurations give these states

stepper motor: making it move



CW:

step:	A	B	A'	B'
1	1	0	0	0
2	0	1	0	0
3	0	0	1	0
4	0	0	0	1

CCW:

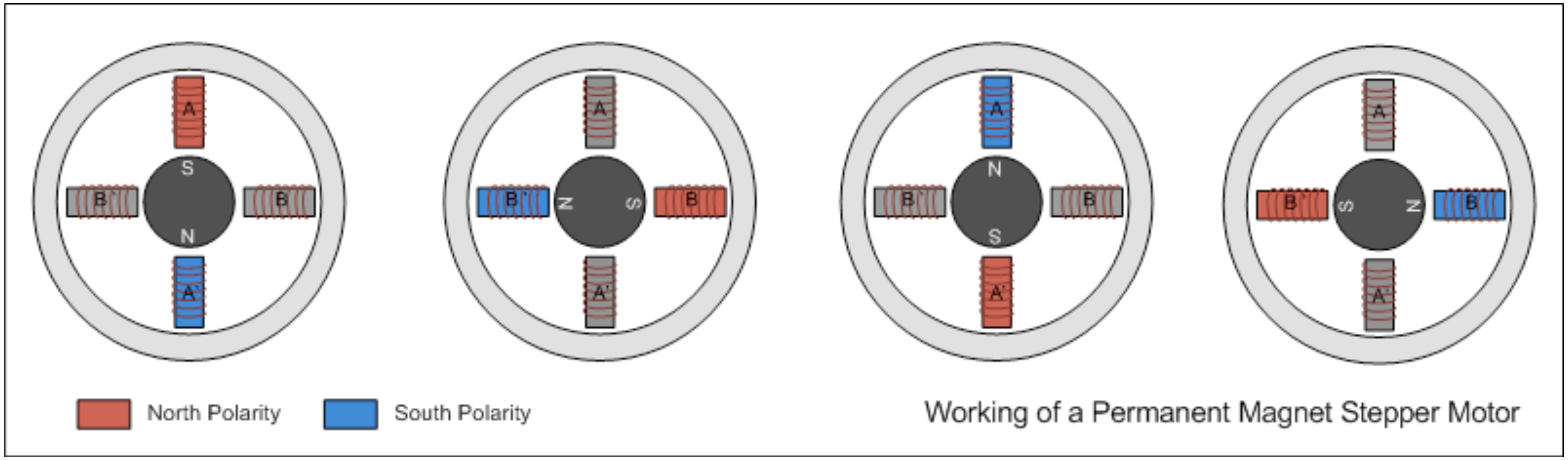
step:	A	B	A'	B'
1	0	0	0	1
2	0	0	1	0
3	0	1	0	0
4	1	0	0	0

repeat, in order, for continuous movement

Q:

1. power
2. resolution
3. speed

stepper motor speed

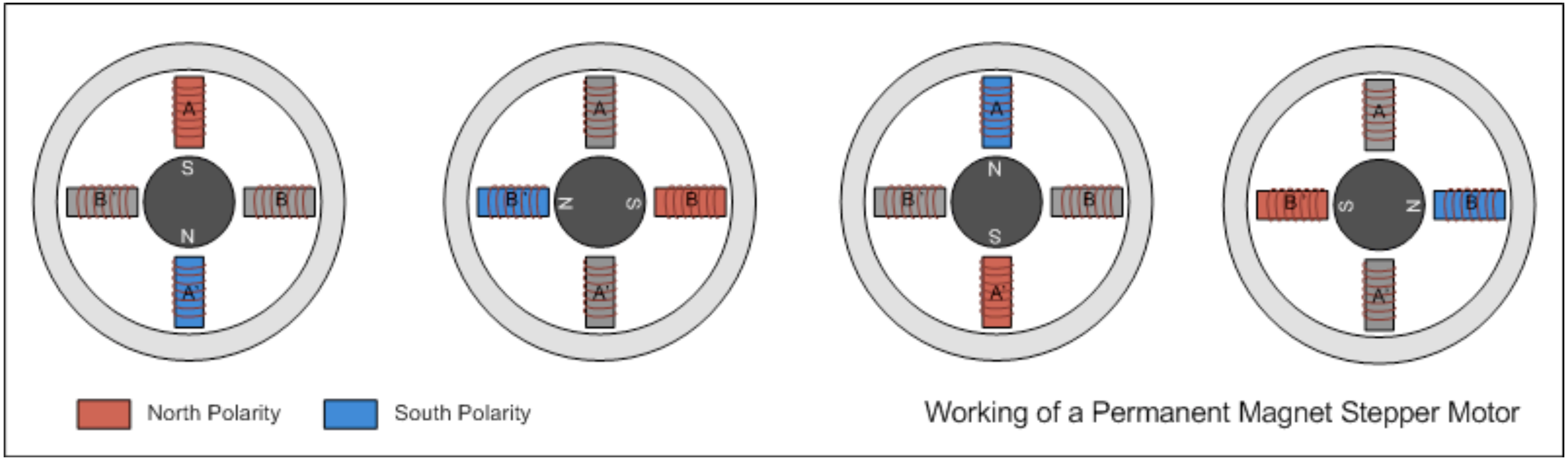


ideal: rotor instantaneously (and continuously) follows stators

step through faster or slower

step:	A	B	A'	B'
1	1	0	0	0
2	0	1	0	0
3	0	0	1	0
4	0	0	0	1

stepper motor speed



steps per revolution = 4 ⟵ depends on stator/rotor (teeth)

depends on motor

$$RPM = \frac{steps}{second}\frac{revolutions}{step}\frac{seconds}{minute}$$

depends on uC

lousy Greeks and their sexagesimal system...

metric time

Remember this time people: 80 past 2 on April 47th.

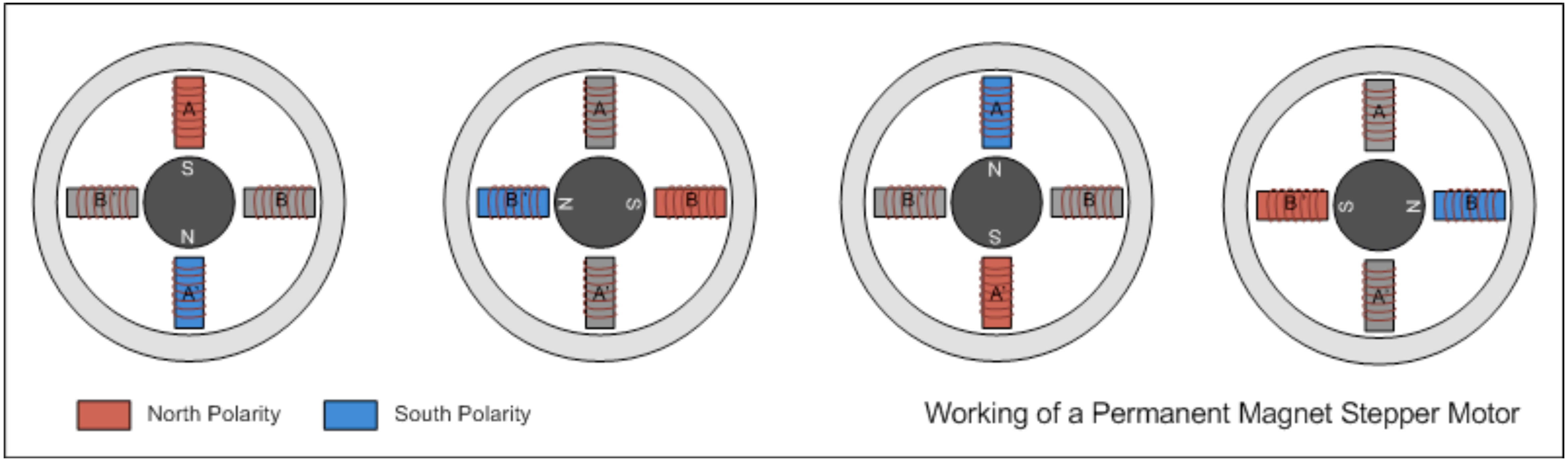
\- Principle Skinner

Q: why do we have the system we have?

couldn't resist...

power

1 2 3 4

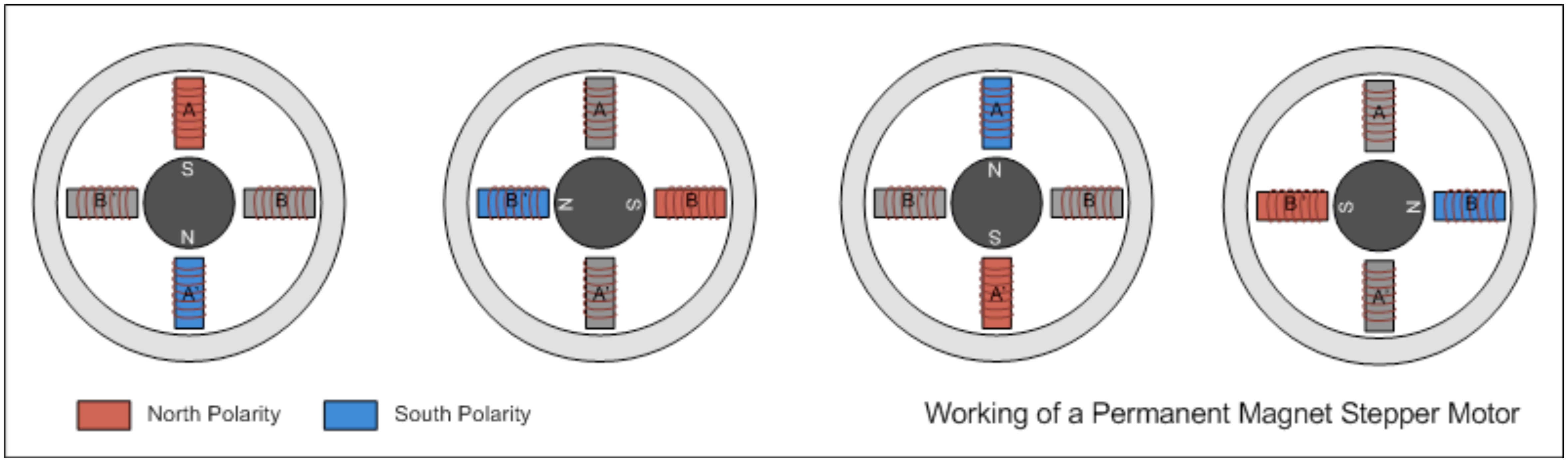


Q: why only turn one set on at a time?

what configuration causes it still to turn?

power

1 2 3 4

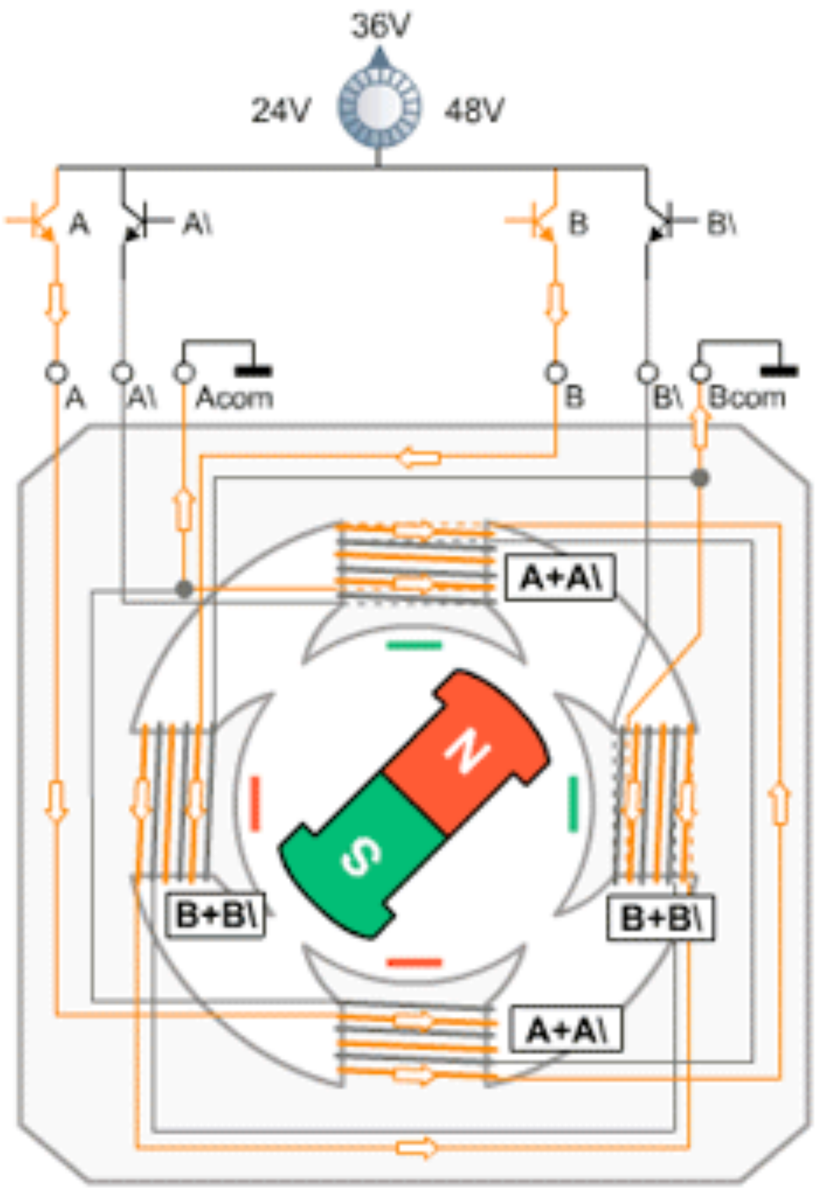


brings stator halfway between A&B

step:	config
1	1+2
2	2+3
3	3+4
4	4+1

	A	B	A'	B'	
CW ↓	1	1	0	0	↑
	0	1	1	0	
	0	0	1	1	
	1	0	0	1	CCW

power



6 Lead Unipolar Driver

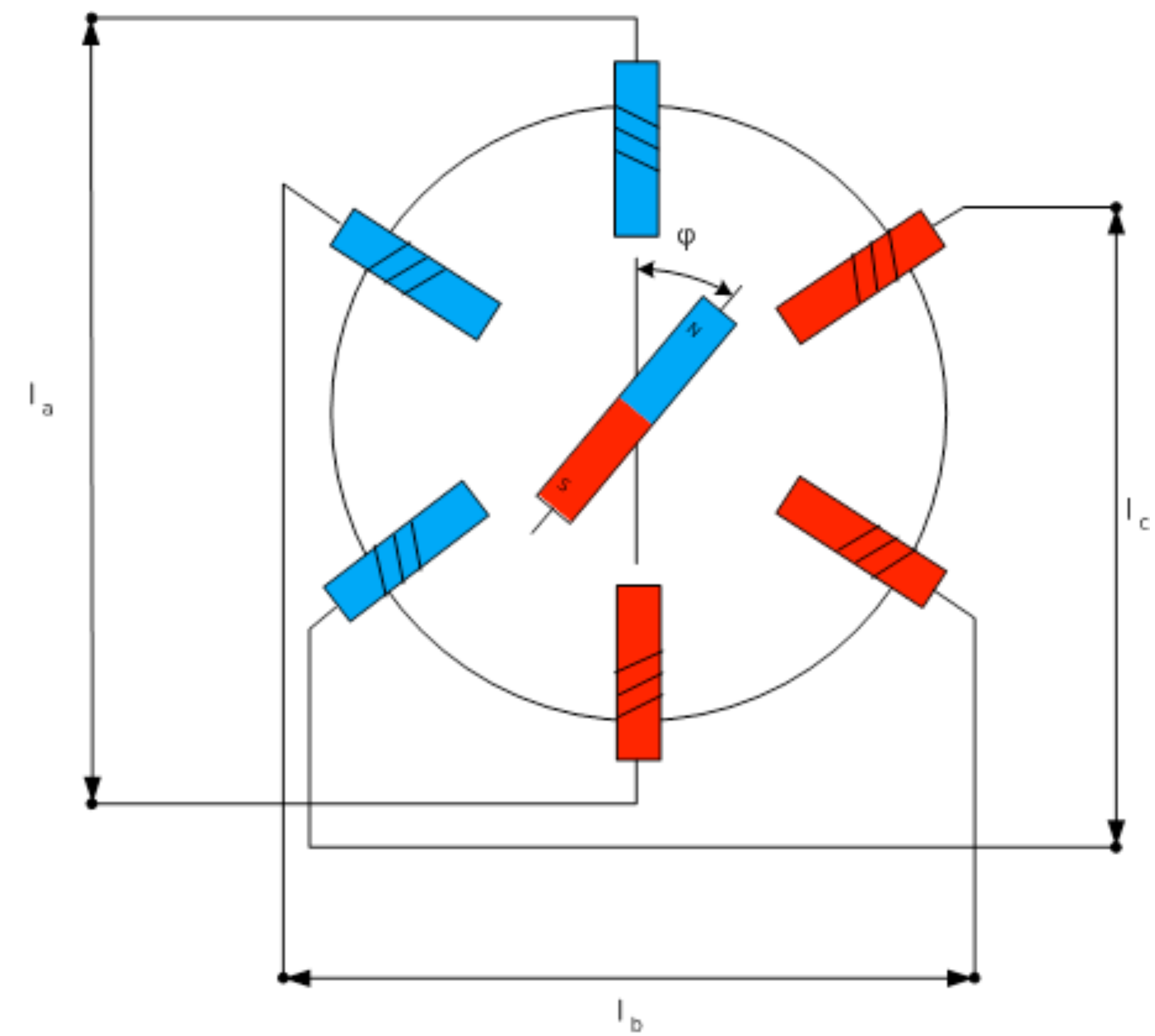
Unipolar control is the most simple and cost-effective way to drive a stepper motor, but results in approximately 30% less torque in comparison to the nowadays widely used bipolar drivers. Since the cost advantage is very small today due to cheap integrated circuits, bipolar drivers are now used in most new applications.

Stepmode

F	0		1		2		3	
H	0	1	2	3	4	5	6	7
A	1	0	0	0	0	0	1	1
B	1	1	1	0	0	0	0	0
A	0	0	1	1	1	0	0	0
B	0	0	0	0	1	1	1	0
dez	12	4	6	2	3	1	9	8

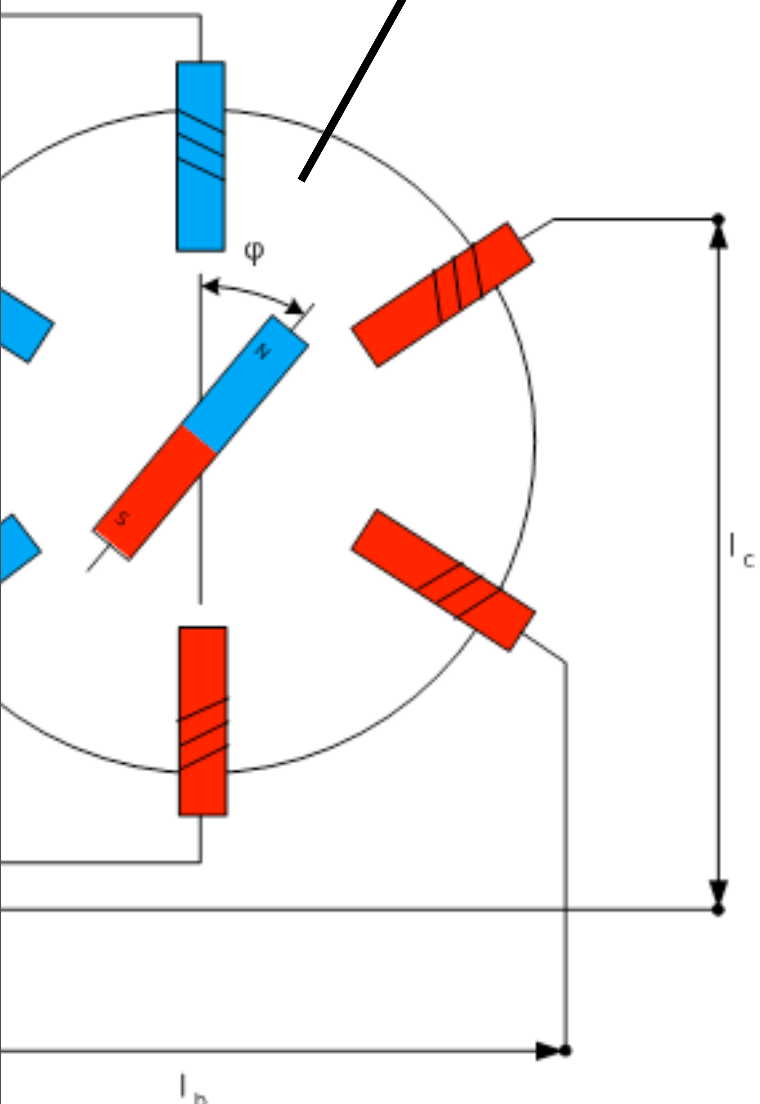
follow
F

even more power: previous + more stators



resolution

angle per step



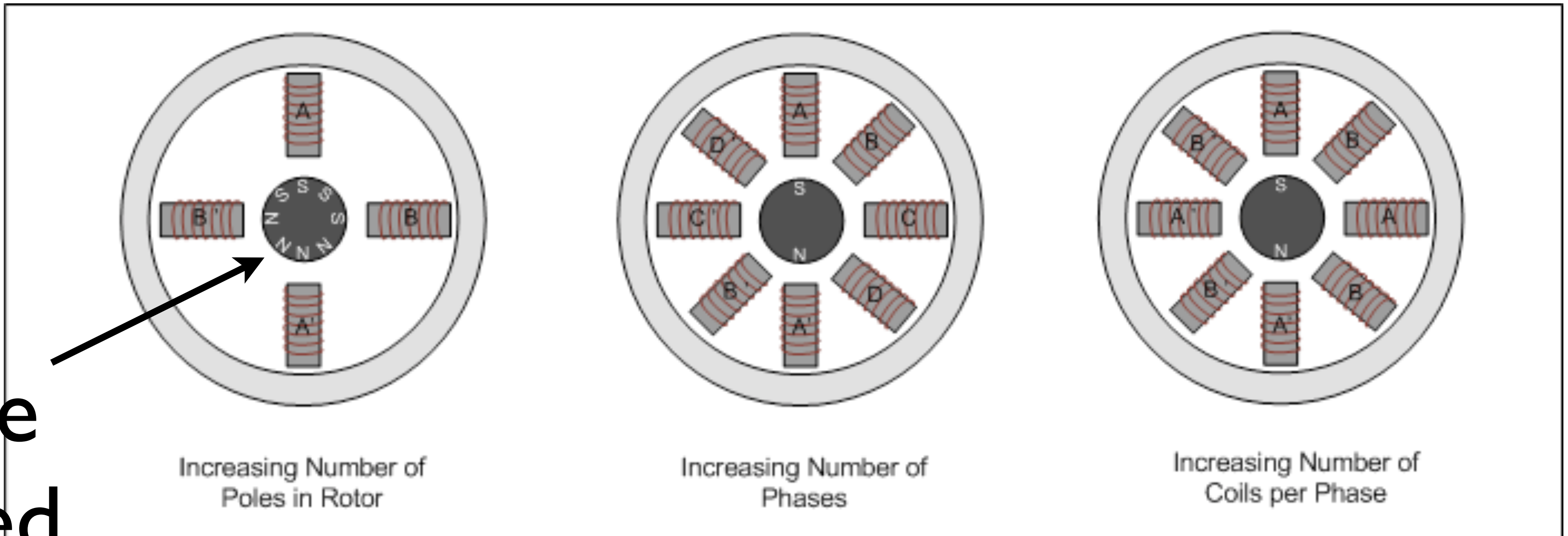
$$\text{step angle} = 360\frac{degrees}{revolution}\frac{revolutions}{step}$$

e.g. steps per revolution = 4

step angle = 360/4 = 90

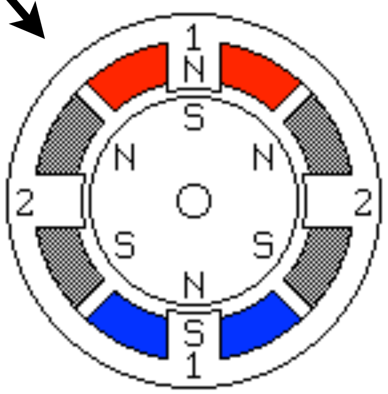
resolution

ways to achieve it:



Methods to increase resolution of a Permanent Magnet Stepper

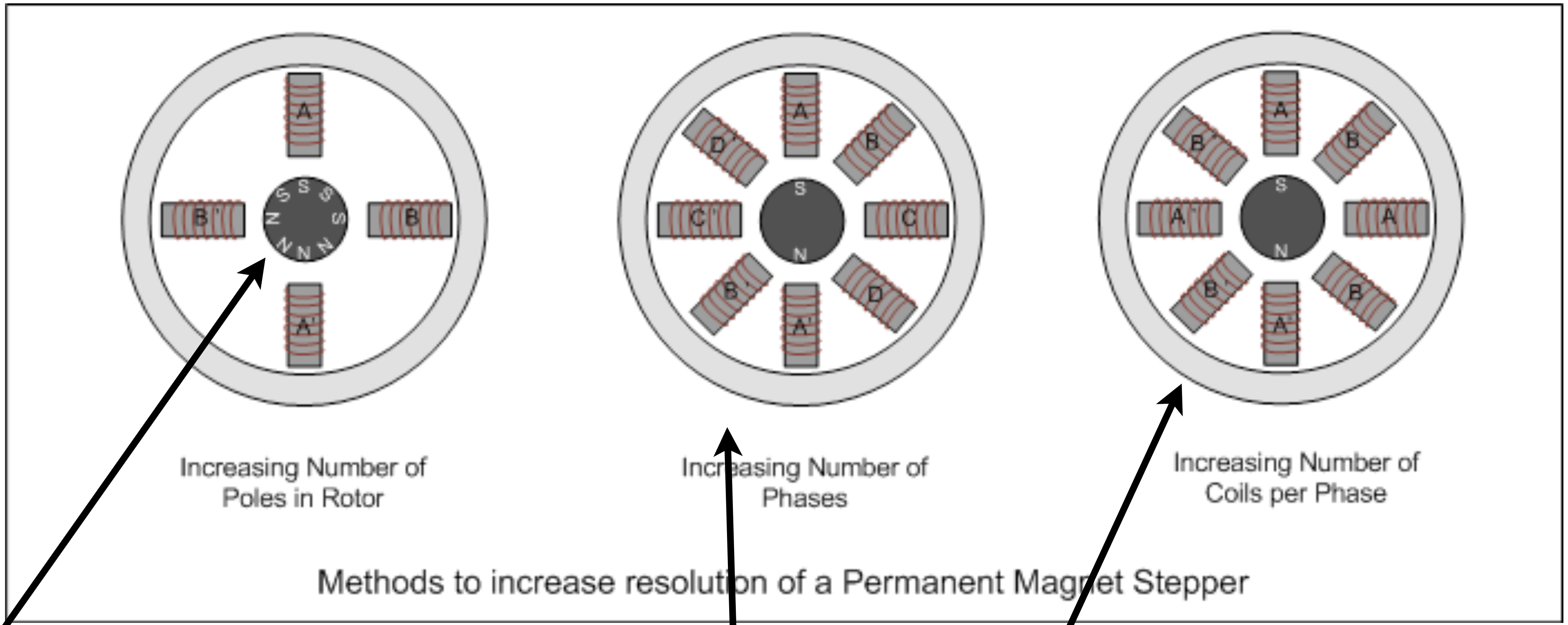
should be interleaved



all require more steps to get one revolution

resolution

ways to achieve it:



basic four-step sequence must repeat many times for one rev

more unique steps

resolution

SA = 45

also, half-stepping:

0,90,180,270

step:	A	B	A’	B’
1	1	0	0	0
2	0	1	0	0
3	0	0	1	0
4	0	0	0	1

+

45,135,225,315

A	B	A’	B’
1	1	0	0
0	1	1	0
0	0	1	1
1	0	0	1

=

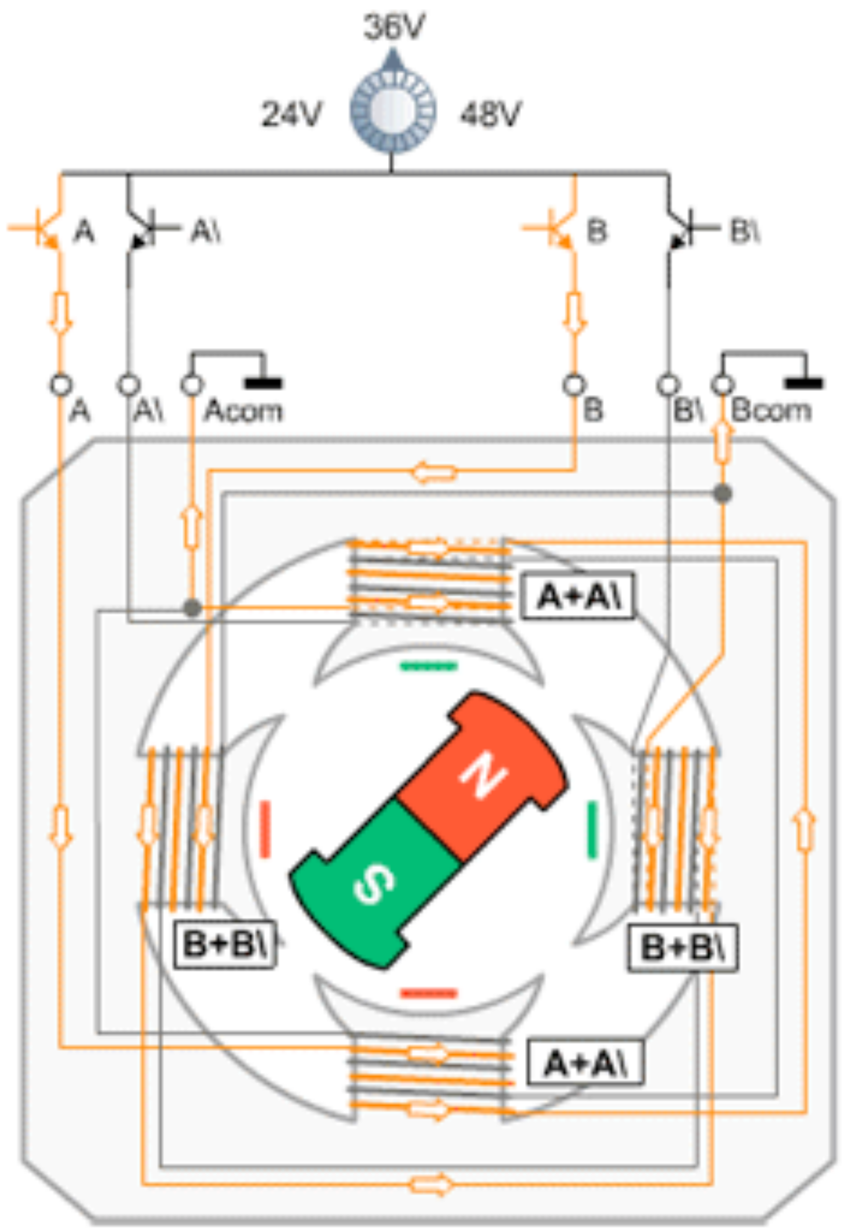
step:	A	B	A’	B’
1	1	0	0	0
2	1	1	0	0
3	0	1	0	0
4	0	1	1	0
5	0	0	1	0
6	0	0	1	1
7	0	0	0	1
8	1	0	0	1

SA = 90

CW

CCW

resolution: half-stepping



6 Lead Unipolar Driver

Unipolar control is the most simple and cost-effective way to drive a stepper motor, but results in approximately 30% less torque in comparison to the nowadays widely used bipolar drivers. Since the cost advantage is very small today due to cheap integrated circuits, bipolar drivers are now used in most new applications.

Stepmode

F	0		1		2		3	
H	0	1	2	3	4	5	6	7
A	1	0	0	0	0	0	1	1
B	1	1	1	0	0	0	0	0
A\	0	0	1	1	1	0	0	0
B\	0	0	0	0	1	1	1	0
dez	12	4	6	2	3	1	9	8

follow H

con: uneven torque