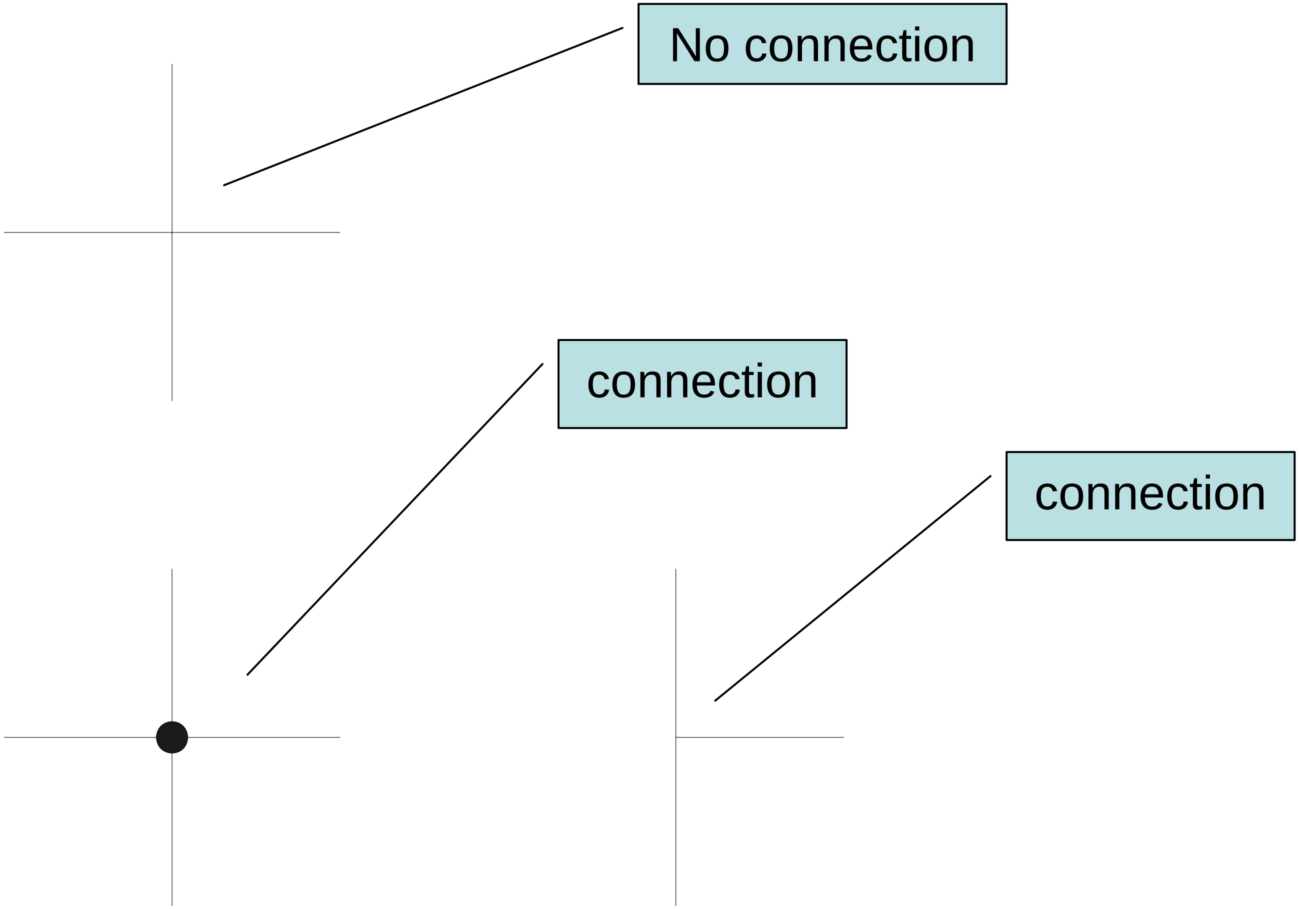
# Stepper Motors

- DC motors with permanent motors and multiple coils around the body.
- Coils are turned on and off in sequence to cause the motor to turn.
- Because the coils are turned on and off they are easy to control with microcomputer and digital circuits. At any given time, the position of the shaft is known.
- Holding torque requires power.

# Servos

- Servos are motors with electronic circuitry that controls the angular position of  the shaft based on a control signal. If the angle is incorrect the motor is turned on until the correct position is reach.
- Angular position controlled by a 0 – 2.0 ms pulse width.

# Schematic Drawing Convention

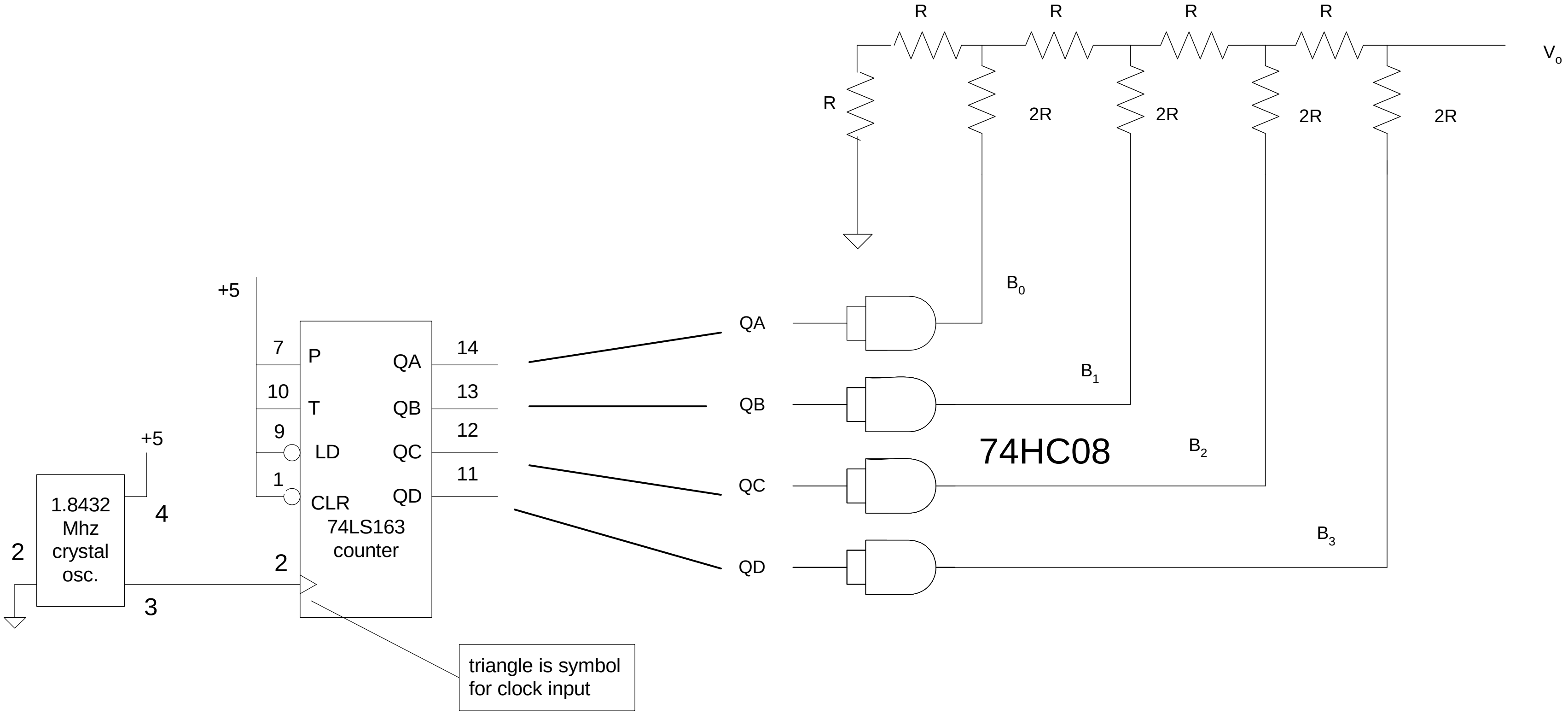


No connection

connection

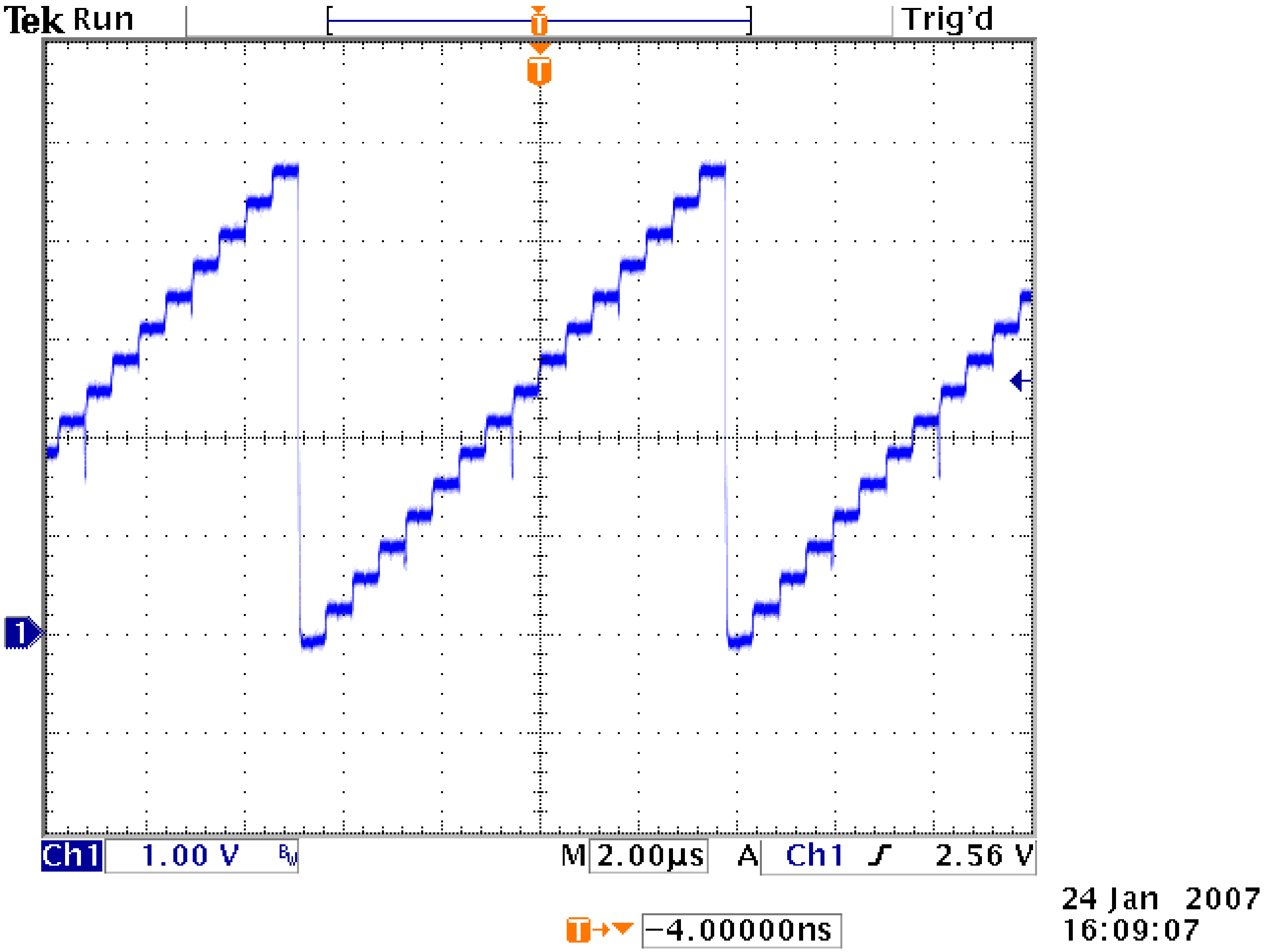
connection

# Lab Exercise - Review
# Ramp Generator



triangle is symbol for clock input

# RAMP Generator Output



# DA Summary

- Output from digital to analog conversion are discrete levels.
- More bits means better resolution.
- An example of DA conversion
  - Current audio CD’s have 16 bit resolution or 65,536 possible output levels
  - New DVD audio samples at 192 khz with 24 bit resolution or $2^{24}$ = 16,777,216

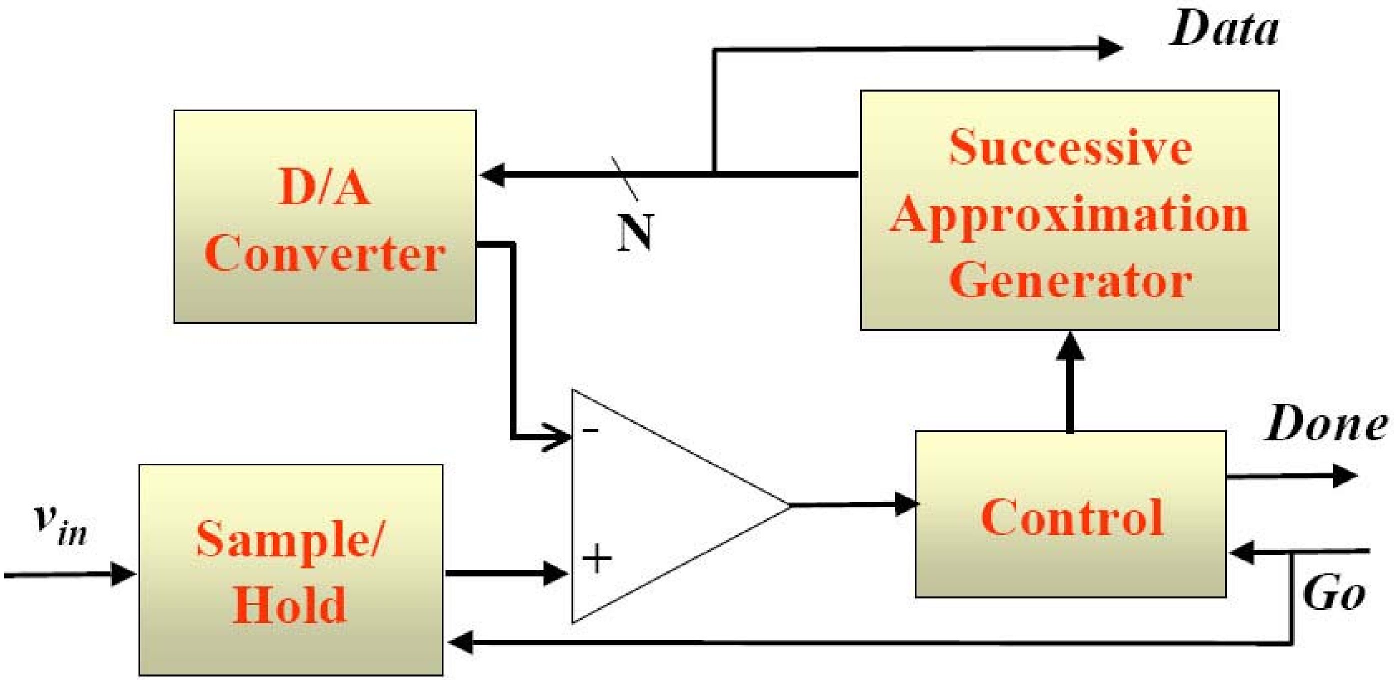
# Analog to Digital Conversion (ADC)

- Successive approximate conversion steps
  - Scale the input to 0-3 volts (example)
  - Sample and hold the input
  - Internally generate and star case ramp and compare

- Flash Compare
  - Compare voltage to one of $2^n$ possible voltage levels. 8 bit ADC would have 255 comparators.

- Note that by definition, ADC have quantizing errors (number of bits resolution)

# Successive Approximation AD

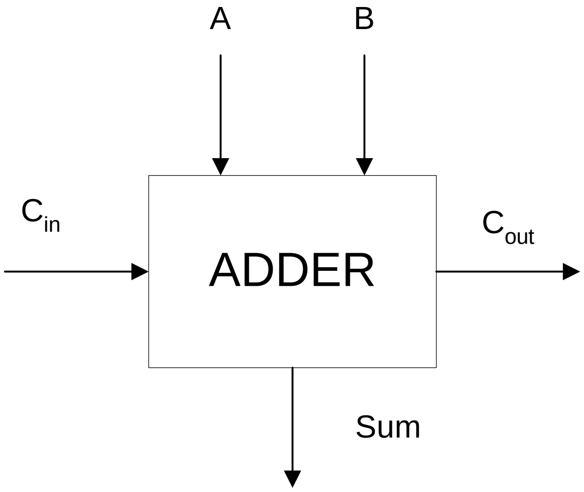


Serial conversion takes a time equal to $N(t_{D/A} + t_{comp})$

(6.111 L14, Fall ’05)

# Binary Adder – m$^{th}$ bit

| $C_{in}$ | A | B | Sum | $C_{out}$ |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 1 |
| 1 | 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 |

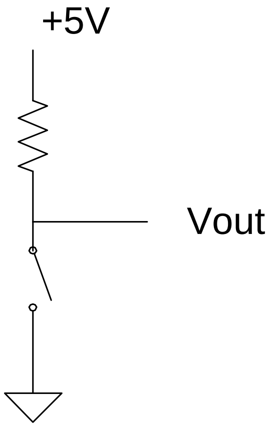


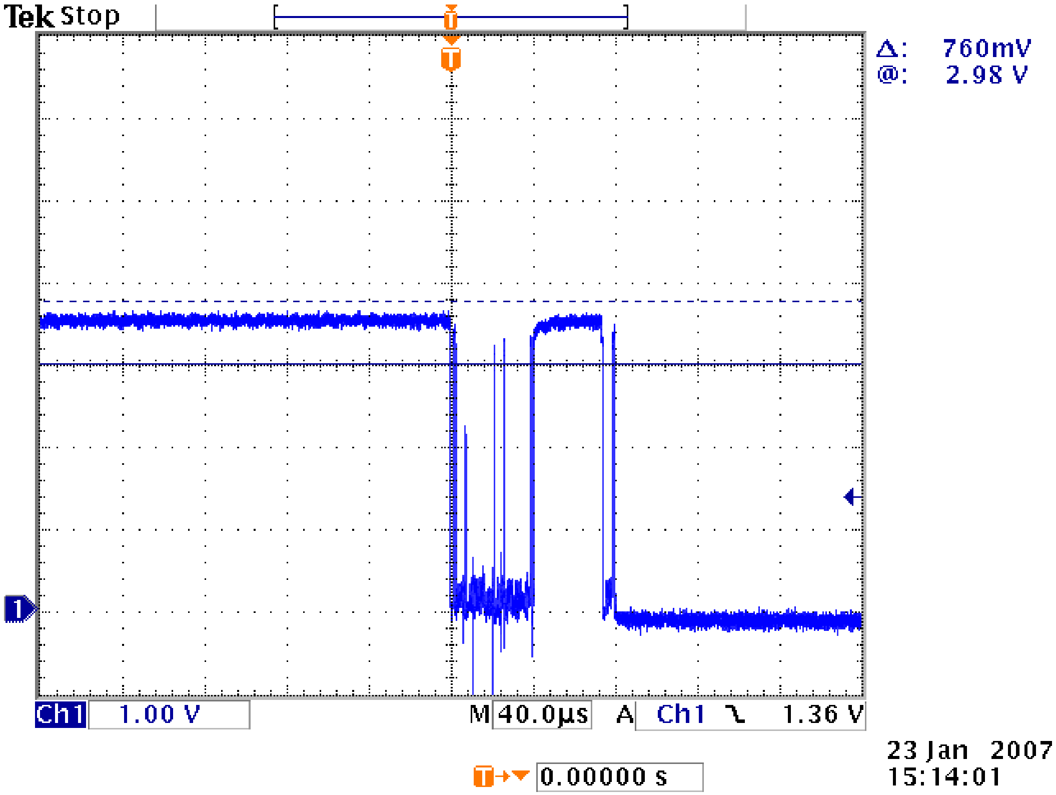
**Sum =**

**$C_{out}$ =**

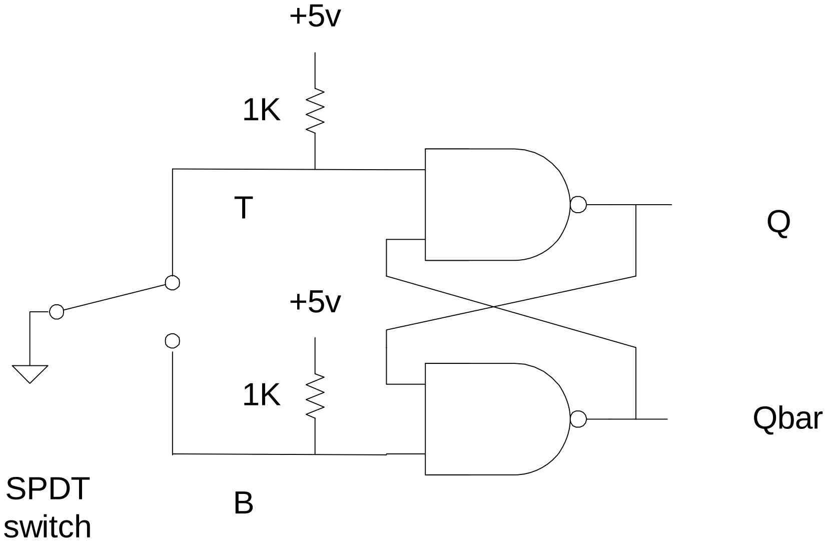
# Switch Bounce

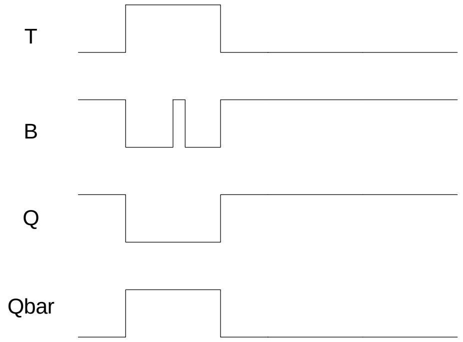
- All mechanical switches have "switch bounce"





# Debounce Circuit





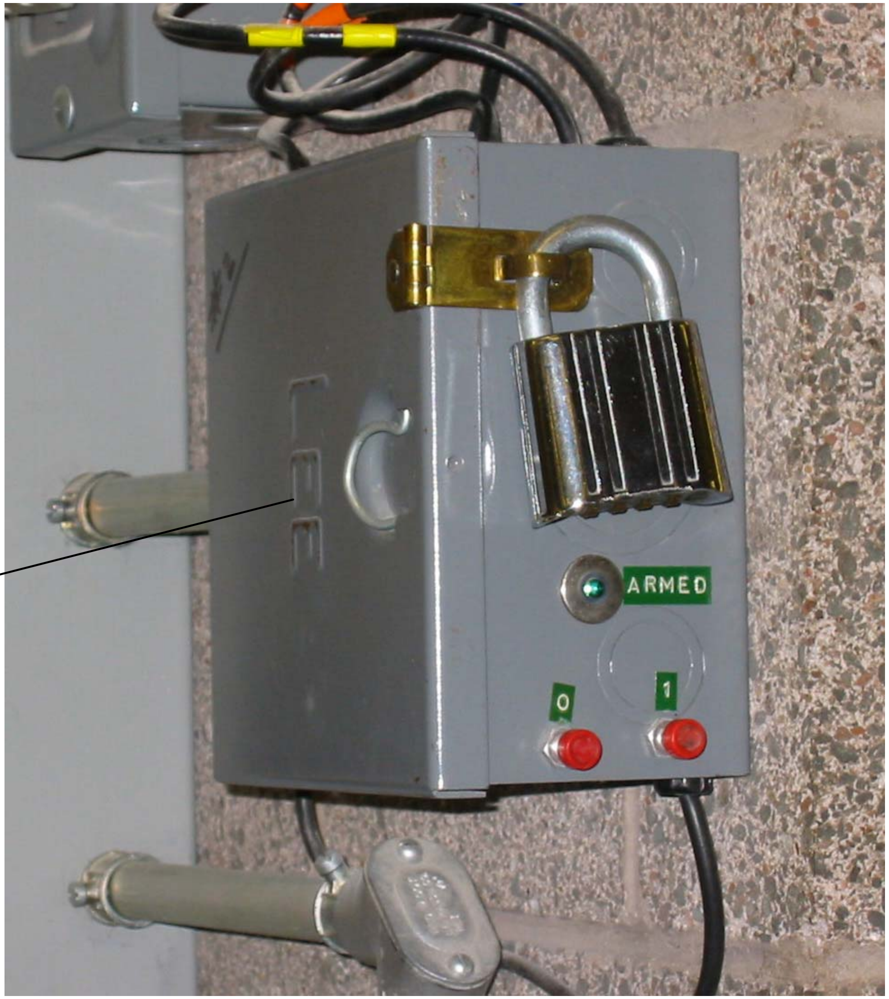
Requires SPDT switch

| T | Qbar | Q |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

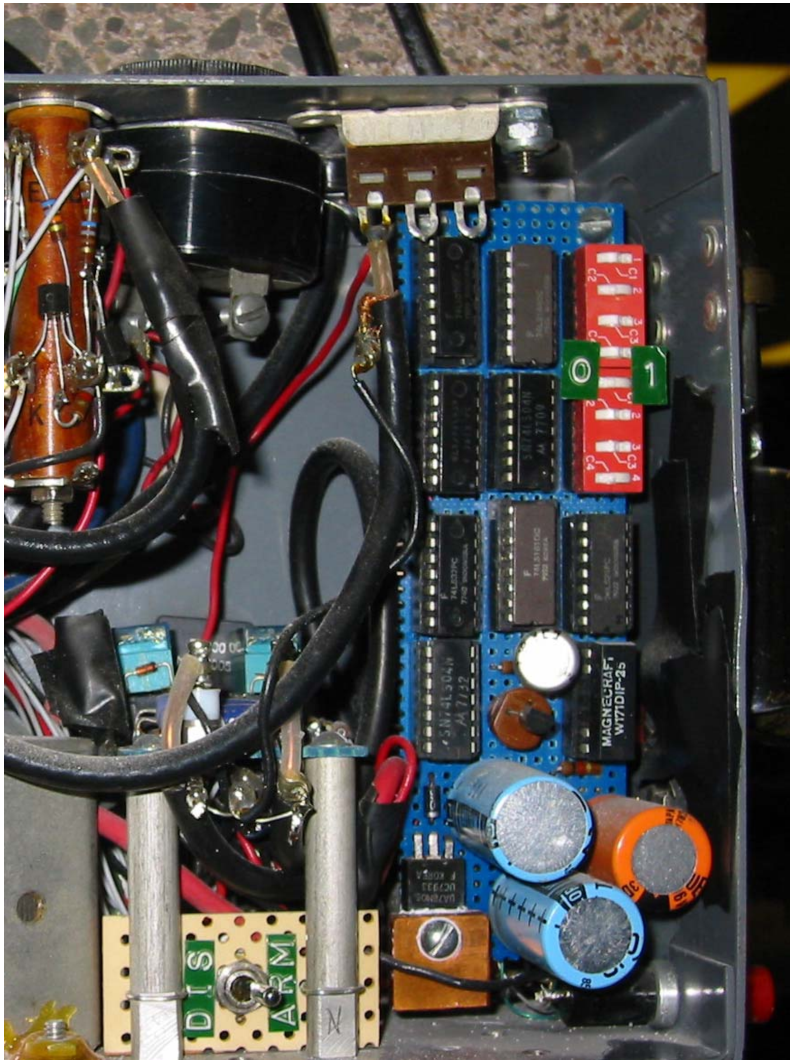
| | | | |
|---|---|---|---|
| **T** | 0 | | |
| **B** | 1 | | |
| **Q** | | | |
| **Qbar** | | | |

# Lab 5

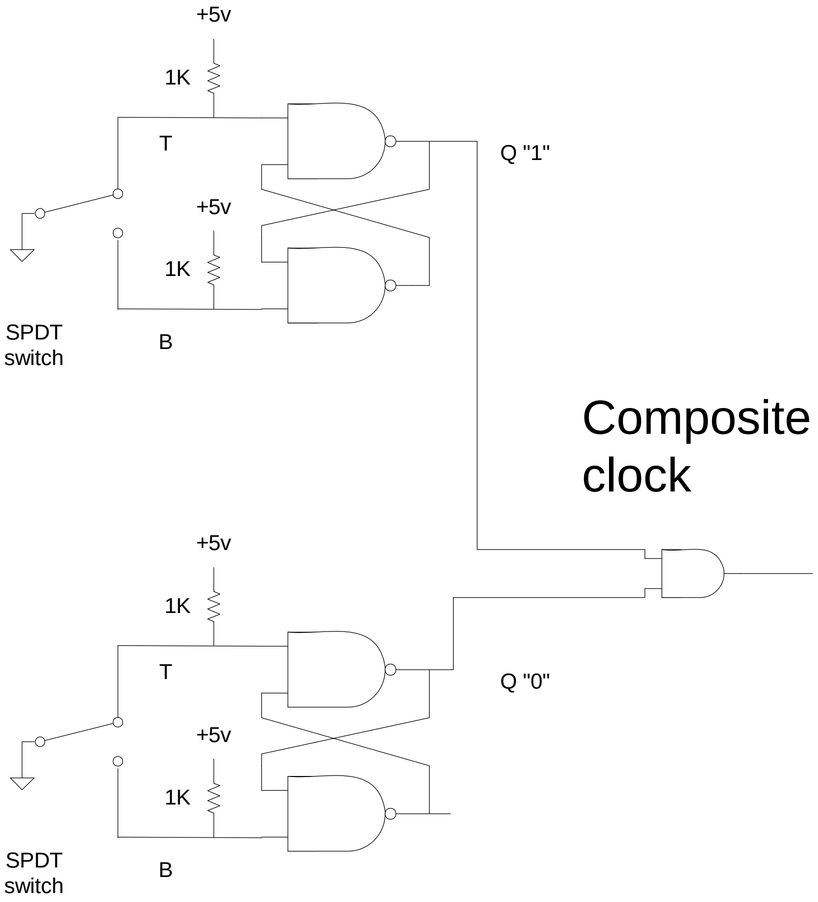
- Design, build and keep the electronics for a digital lock.
- Unlock key based on sequence of 0, 1.



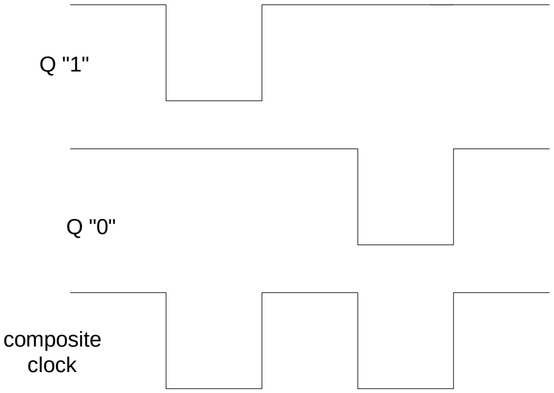
# Digital Lock



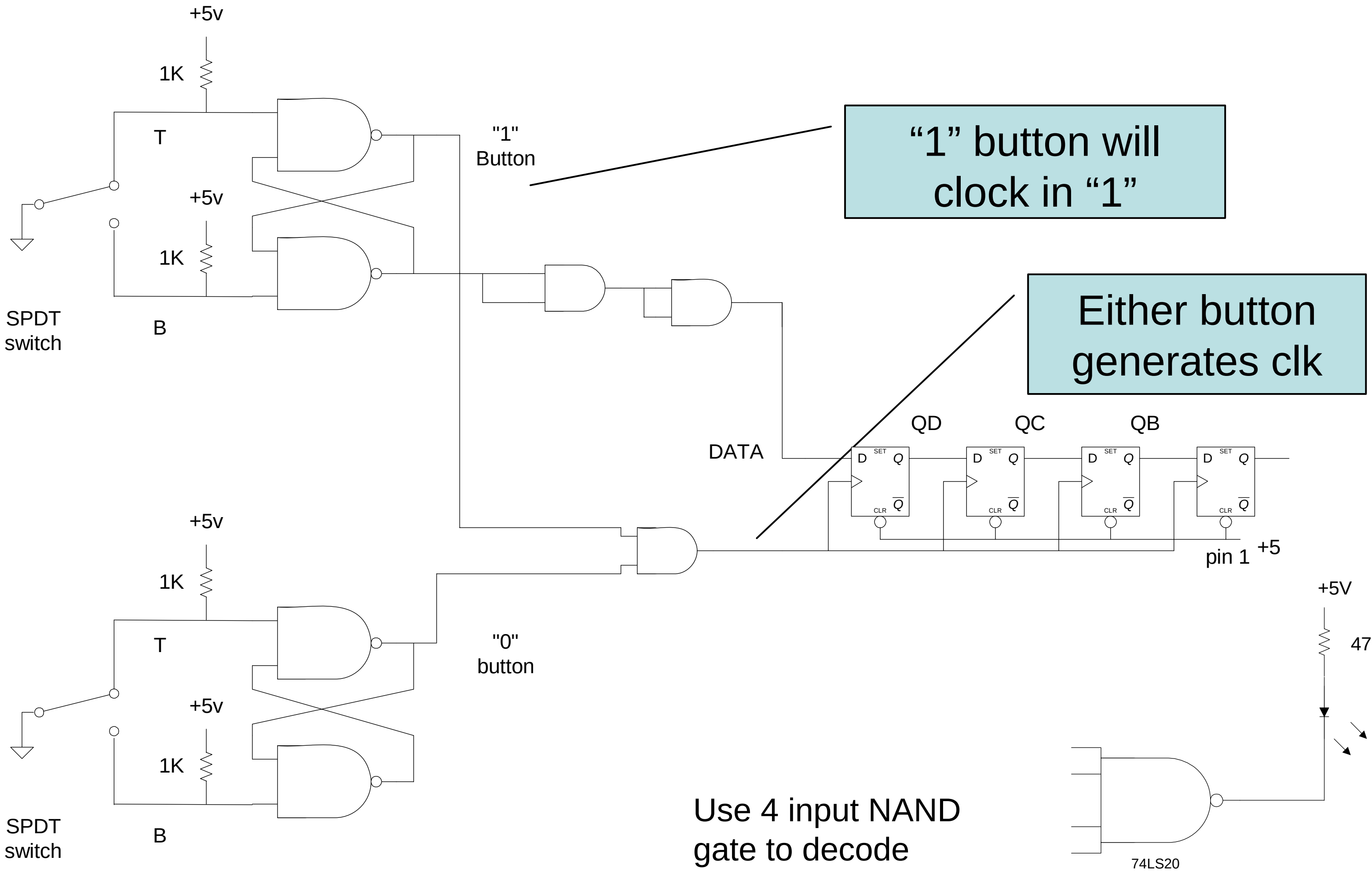
# Pushbutton Clocking



Composite clock



# Digital Lock



“1” button will clock in “1”

Either button generates clk

Use 4 input NAND gate to decode

# Design guidelines

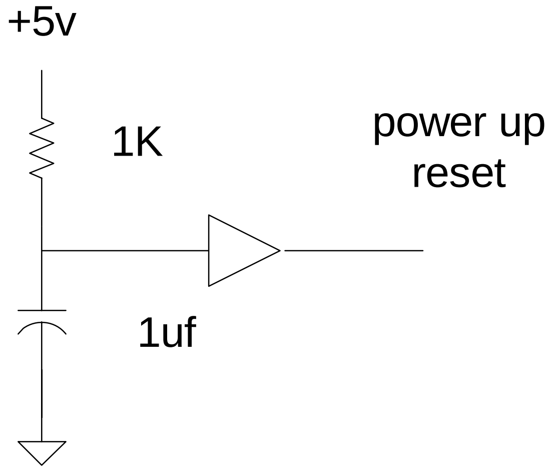
- Apply power and ground to each chip
- Add  10 uf or greater to power bus
- Select and wire up desired code
- For control inputs on all IC's or inputs that matter, tie to a "1" or a "0".  Floating inputs are in an indeterminate state.

# Construction Techniques

- Consider placement of IC’s
- Wire up power and ground to all IC’s
  - Use all four power rails
- Build and debug in stages
  - Debounce circuits
  - Composite clock
  - Shift registers
  - Neat wiring helps!

# Enhancements

- Increase lock code to 8 bits
- Add power up reset



# Lab 5

- Use last three aisle on the left at the end of the 6.111 lab
- Pick up IC’s and tools from LA’s.
- You may keep the completed circuit you build (pushbuttons, IC’s, everything!)
- Return tools.

# Odds and Ends

- FPGA: Field Programmable Gate Array
  - Use high level hardware description language (HDL) to describe behavior
  - Can be re-programmed thousands of times.
  - very inexpensive kits
  - free software tools on web

- Please complete your evaluation of this course