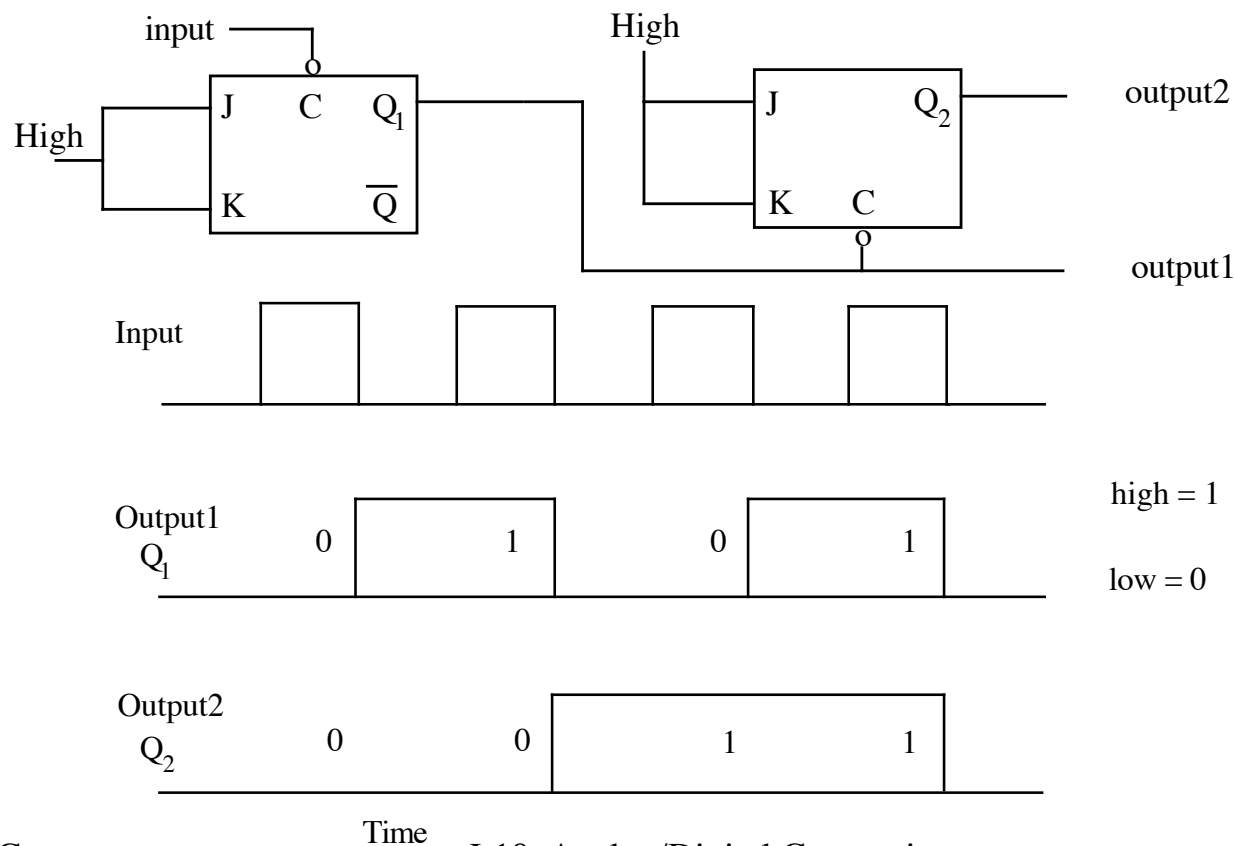
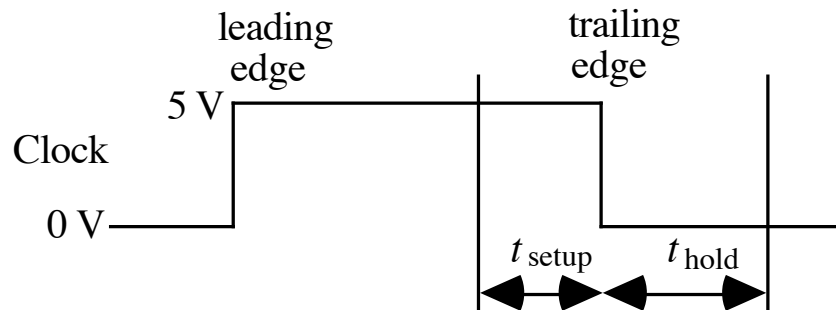


Lecture 10: Analog/Digital Conversion

- Example: Counter made from two JK flip-flops.
 - ◆ This circuit counts from 0 1 2 3 0 1 2 3 0...
 - ◆ Q_1 is the lowest order bit, Q_2 is the higher order bit.
 - ◆ The output is a binary number = Q_2Q_1 .
 - ◆ The o's on the clock means that the output transition occurs on the trailing edge of the clock pulse.
 - ◆ Output of circuit is most conveniently displayed using a timing diagram:

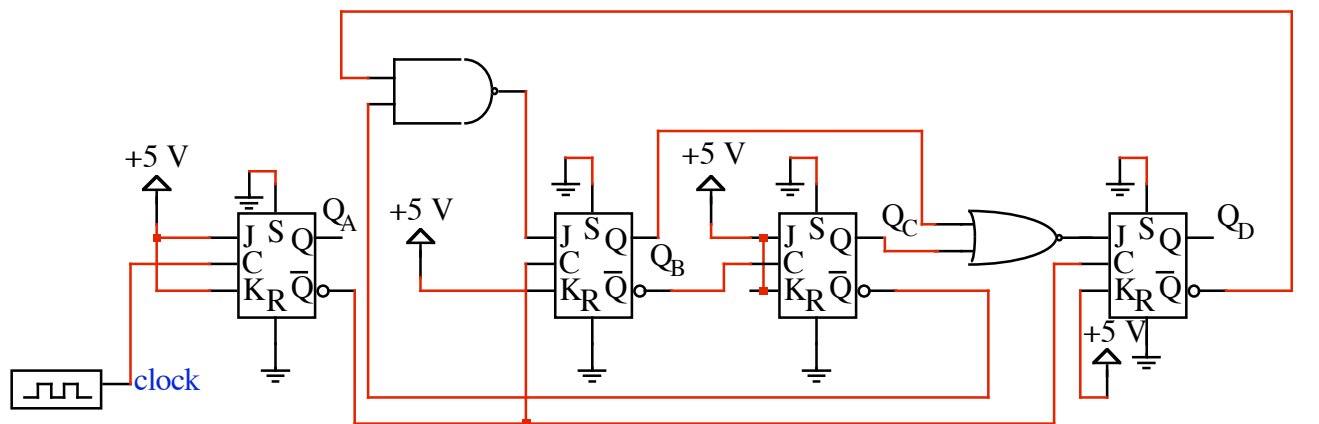


- A few words about clocking the flip-flops and timing of inputs.
 - ◆ Setup time:
For each type of flip-flop there is a minimum specified time relative to the clock pulse during which time the input(s) to the FF must be stable (i.e. not change logic levels).
 - ◆ Hold time:
For each type of flip-flop there is a minimum specified time after Q changes state that the input(s) to the FF must be stable (i.e. not change logic levels).
 - ◆ Example: 74LS112 JK flip-flop (the one we use in lab)
 - This FF changes state (Q) relative to the trailing edge of the clock.
 - The setup time is 20 nsec (2×10^{-8} sec) while the hold time is ≈ 0 nsec.
 - The maximum clock speed of this FF is 30 MHz.



the data on J and K must be stable for at least $t_{\text{setup}} + t_{\text{hold}}$.

- Sometimes circuits with flip-flops are classified according to how the clock is distributed to the FF's.
 - ◆ There are two clocking schemes:
 - Synchronous: All FF's are clocked at the same time.
 - The easiest way to do this is to use one clock and distribute it to all the FF's.
 - Asynchronous: FF's are clocked at different times, usually by different clocks.
 - The previous circuit with two flip-flops was an example of this type of circuit.
 - The first FF was clocked by a “clock”.
 - The second FF was clocked by the output (Q) of the first FF.

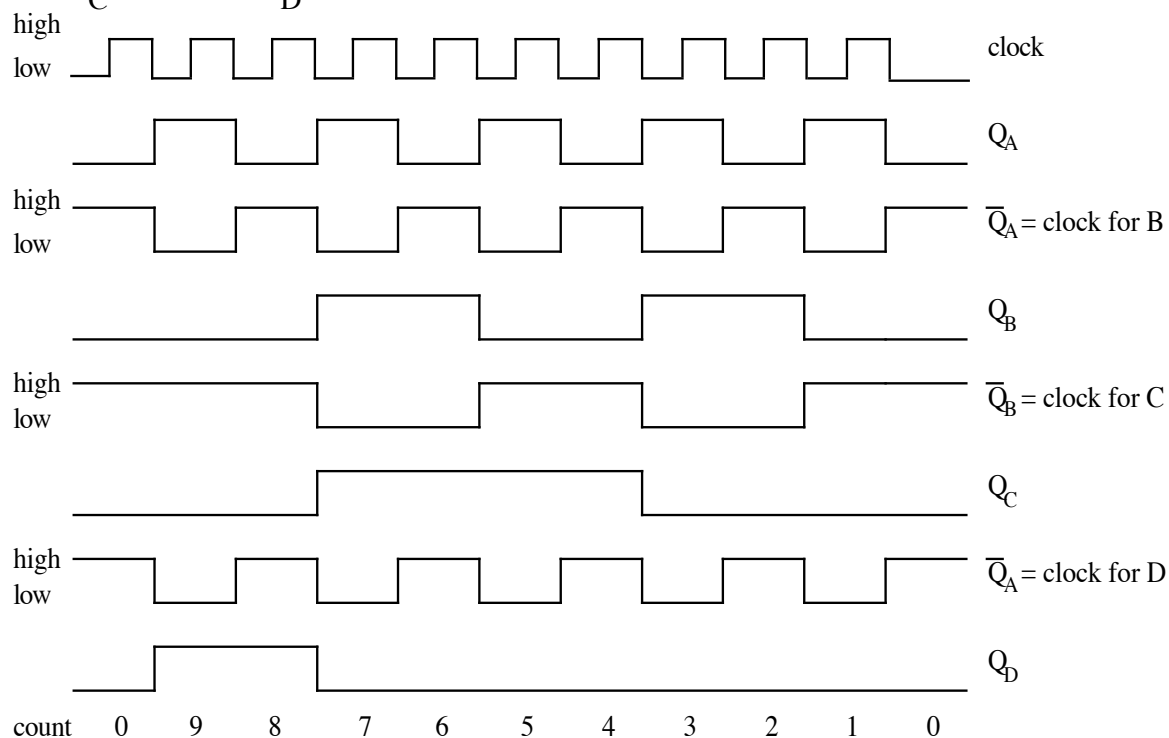


$$J_A = 1 \quad J_B = \overline{Q_C} \overline{Q_D} = Q_C + Q_D \quad J_C = 1 \quad J_D = \overline{Q_B} + Q_C = \overline{Q_B} \overline{Q_C}$$

$$K_A = 1 \quad K_B = 1 \quad K_C = 1 \quad K_D = 1$$

Divide by 10 ripple down counter
(counts from 9 to zero)

Asynchronous counter



Digital to Analog Conversion (DAC):

- There are two simple circuits commonly used to convert a digital signal to an analog voltage.

- Weighted Resistor Ladder:

- We assume that the input voltages (V_1 , V_2 , V_3 , and V_4) are logic levels.
- Let us use TTL levels and assume a high = 3 V and a low = 0 V.
- The output voltage is given by:

$$V_{\text{out}} = \frac{\frac{R_b}{R_a} + 1}{\frac{1}{R_0} + \frac{1}{R_1} + \frac{1}{R_2} + \frac{1}{R_3} + \frac{1}{R_4}} \left[\frac{V_1}{R_1} + \frac{V_2}{R_2} + \frac{V_3}{R_3} + \frac{V_4}{R_4} \right]$$

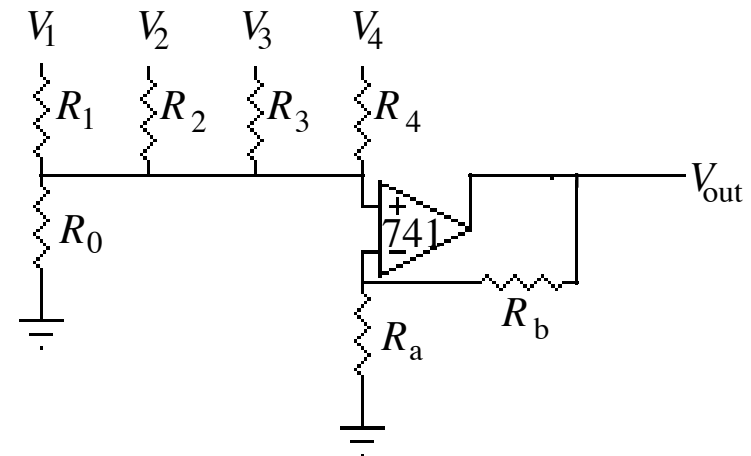
- If we choose the resistors as follows:

$$R_1 = R_a = 1 \text{ k}\Omega, R_2 = 2 \text{ k}\Omega, R_3 = 4 \text{ k}\Omega, \\ R_4 = R_0 = 8 \text{ k}\Omega \text{ and } R_b = 15 \text{ k}\Omega,$$

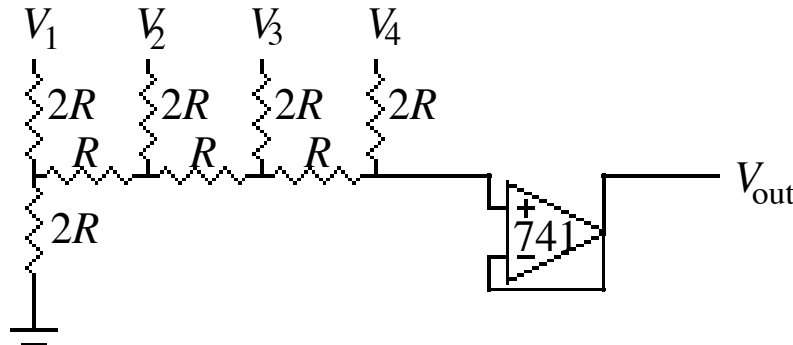
- we get the following simple relationship for V_{out} :

$$V_{\text{out}} = 8V_1 + 4V_2 + 2V_3 + V_4$$

- If V_{in} represents a binary number (e.g. 1001 = $V_1V_2V_3V_4$ with V_1 being the highest order bit)
 - the output voltage varies from 0 to 45 Volts (remember $V_1..V_4$ are all either 0 or 3 V)
 - example: the digital input 1001 has an analog output of 27 V = $(3 \times 8 + 3)$ V.
- Unfortunately there are several bad points with this conversion scheme:
 - the output can be a large voltage (e.g. 45 V)
 - circuit needs 5 high precision resistors (expensive)
 - the current (and therefore power) in the resistors varies by a factor of 15



- Binary Ladder Network (R-2R Network):



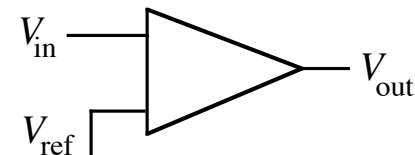
This circuit fixes up many of the problems in previous circuit

- The output voltage for this circuit is:

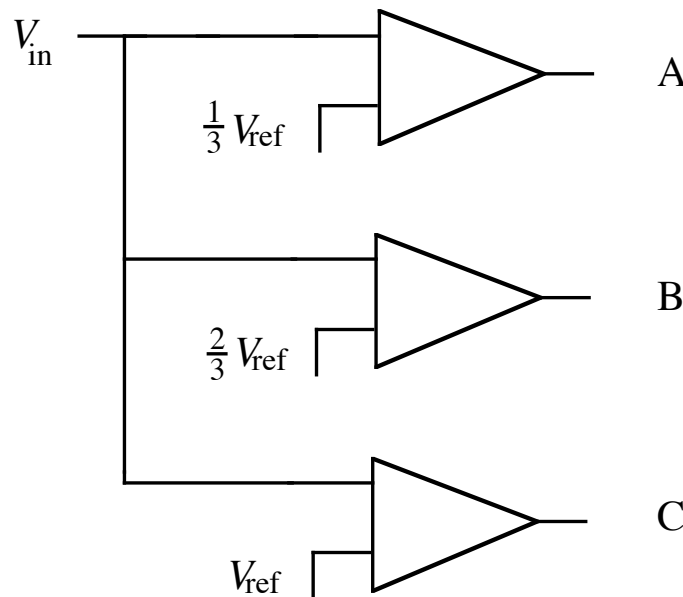
$$V_{out} = V_1/2 + V_2/4 + V_3/8 + V_4/16$$
- This circuit needs only 2 precision resistors compared with the 5 in the previous design.
 - Power dissipated in the resistors varies by a factor of 2, compared with 15 in the previous design.
- There are still some bad points:
 - need precision components
 - the output voltage will usually be a fraction of the input (low noise immunity)
 - Example: if $V_{in} = 1001$, then $V_{out} = 3/2 + 0/4 + 0/8 + 3/16 = 27/16$ V for TTL logic levels.

Analog to Digital Conversion (ADC)

- Parallel A to D conversion (“Flash Encoder” or Flash ADC)
 - very fast and very simple method
 - use comparators for the conversion
 - Example: one bit ADC using one comparator
 - $V_{out} = 1$ (high) if $V_{in} > V_{ref}$
 - $V_{out} = 0$ (low) if $V_{in} < V_{ref}$



- ◆ How many comparators do we need for a given accuracy?
 - Suppose we want to convert an analog number into a 2 bit digital number.
 - For 2 bits there are 4 possible outcomes (00, 01, 10, 11), it takes 3 comparators.
- ◆ Example: 2 bit parallel converter



Truth Table					
V_{in}	A	B	C	Output	
$< \frac{1}{3} V_R$	0	0	0	0	
$\frac{1}{3} V_R < V_{in} < \frac{2}{3} V_R$	1	0	0	1	
$\frac{2}{3} V_R < V_{in} < V_R$	1	1	0	2	
$> V_R$	1	1	1	3	

- Logic gates are needed to implement the truth table:

$$\text{LSB} = A\bar{B}\bar{C} + ABC = A(\bar{B}\bar{C} + BC)$$

$$\text{MSB} = A\bar{B}\bar{C} + ABC = AB(C + \bar{C}) = AB$$

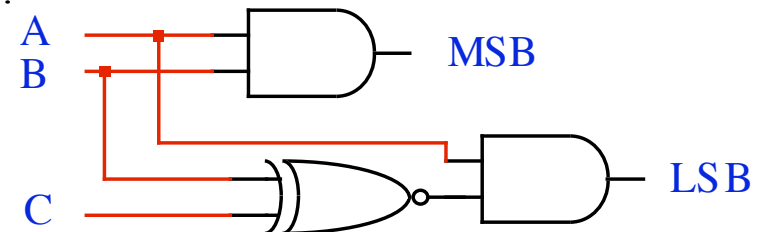
- ◆ There are two problems with this scheme:

- Need lots of comparators: $2^n - 1$ for n bit accuracy.

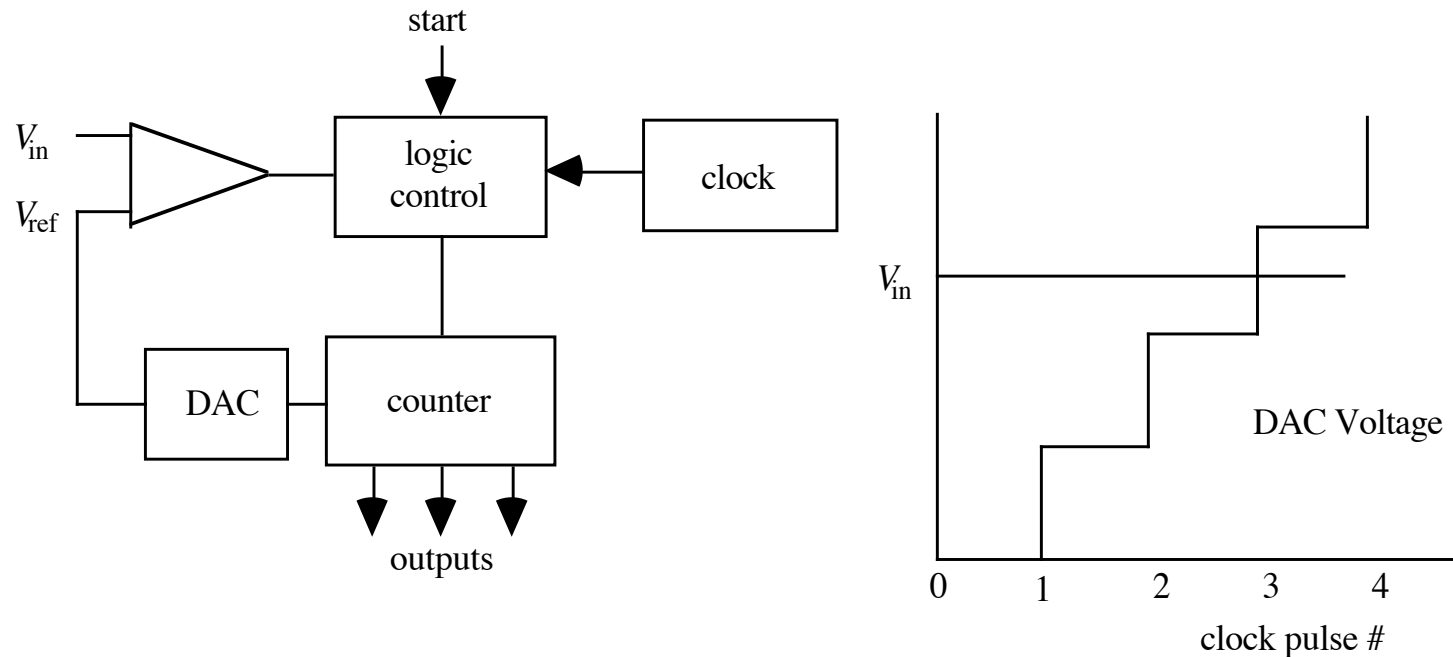
- If we want 1 part in 1000 accuracy (0.1%), it takes 10 bits

$$2^{10} - 1 = 1023 \text{ comparators!}$$

- Number of logic gates necessary to code the output is large and the logic gets complicated.



- Counter ADC (staircase method):
 - ◆ Good news: only uses one comparator
 - ◆ Bad news: much more complicated than parallel method



- ◆ When $V_{DAC} > V_{in}$
 - ☞ the logic circuit stops the clock
 - ☞ the counter outputs a binary number which is just the number of clock pulses
- ◆ The DAC could be:
 - an integrator
 - a resistor ladder
 - a voltage reference

- ◆ Problems with this system:
 - control logic is complicated (use microprocessors + gates +...)
 - time to digitize depends on V_{in} .
 - Example: suppose clock runs at 5 MHz, and you want 10 bit accuracy.
 - 10 bits = 1024 clock pulses.
 - ☞ Can only digitize at 5 MHz/1024 ~ 5 kHz, *which is fairly slow!*

- Successive Approximation ADC:

- ◆ Control logic is very complicated, but easy to program, like a binary search.
- ◆ Conversion time is fast and almost independent of V_{in} .
- ◆ Example: 5 bit ADC:

- Steps for converting V_{in} into a binary number:

- turn on MSB in DAC and compare with V_{in} .
 - output = 0 if $V_{DAC} > V_{in}$ (steps 1, 2, 5)
 - output = 1 if $V_{DAC} < V_{in}$ (steps 2, 3)
- turn on next highest bit, output = 1 if $V_{DAC} > V_{in}$
- repeat until least significant bit is checked
- ☞ n comparisons for n bit accuracy
- staircase approach requires $2^n - 1$ comparisons
- parallel ADC requires 1 step, independent of accuracy
- Time for a 10 bit conversion for the three methods:

parallel : success approx. : staircase 1:10:1023

