EE 505

Lecture 13

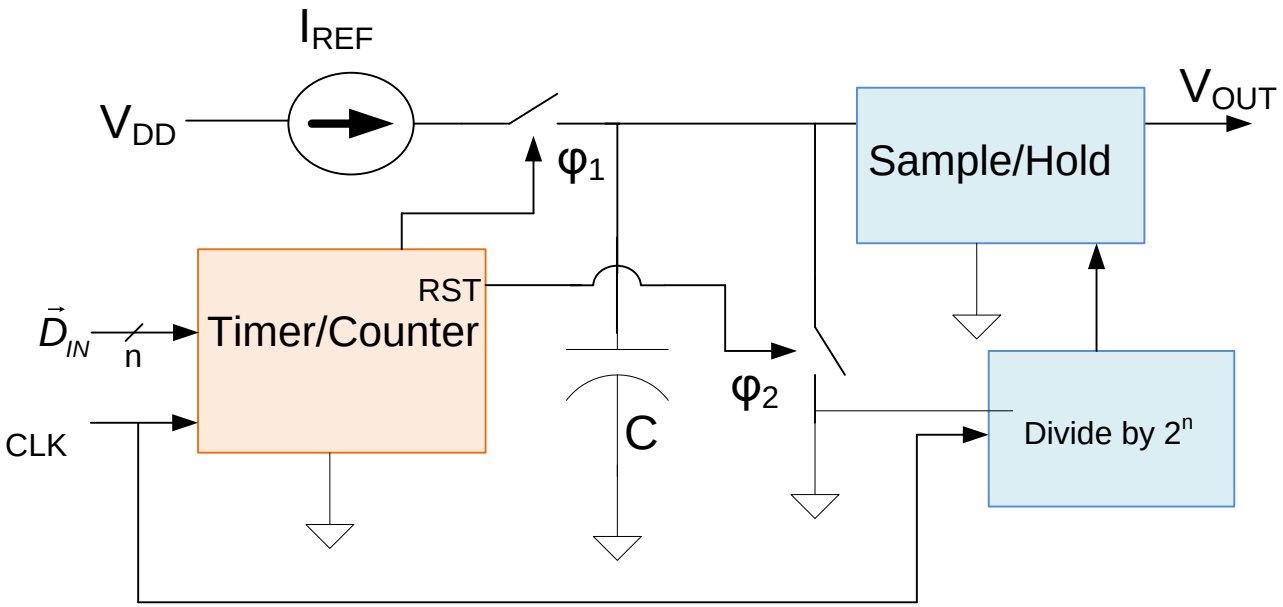
String DACs

Current Steering DACs

Review from Last Lecture

DAC Architectures

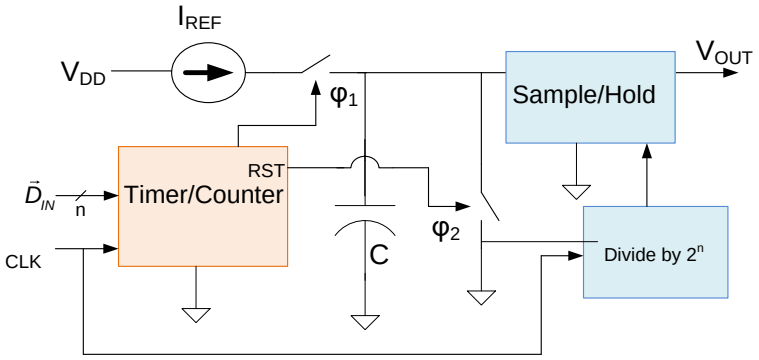
Single Slope



Single-Slope DAC

DAC Architectures

Single Slope



Single-Slope DAC

Can be viewed as a time-domain DAC where resolution headroom is very large

Benefits of Single-Slope ADC?

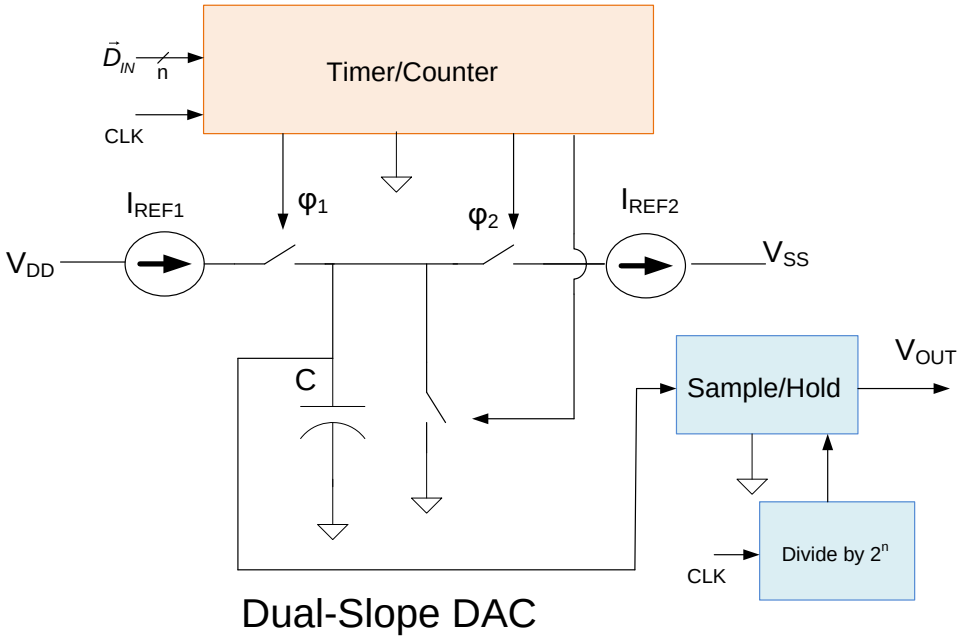
- No matching required
- Very simple structure
- Mostly Digital
- Very low DNL
- Very fine resolution possible
- No previous code dependence
- No binary to thermometer decoder

Limitations of Single-Slope ADC?

- Slow conversion rate
- Large C
- Leakage currents that will be temperature dependent
- Nonlinearity in C?
- Nonlinearity in I_{REF}

DAC Architectures

Dual Slope

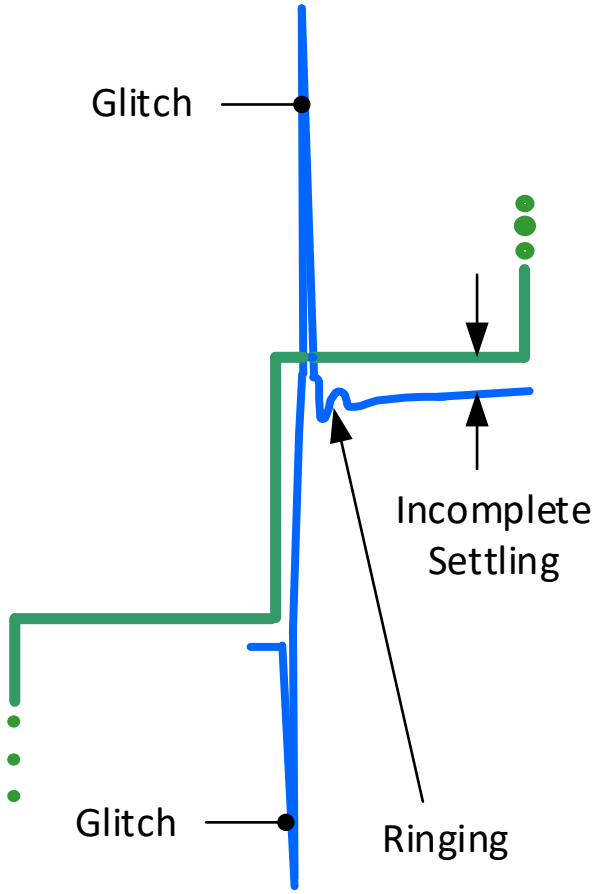


Dual-Slope DAC

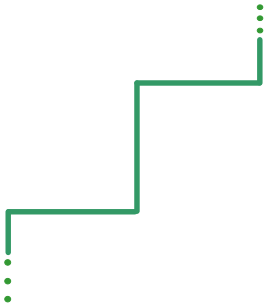
Provides DAC with positive and negative outputs

Term “dual slope” means something different here than what we see in “dual slope” ADCs

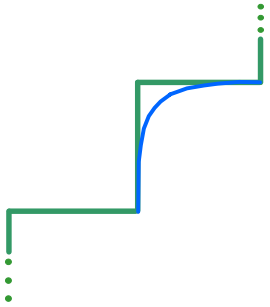
DAC Performance Issues and Concerns



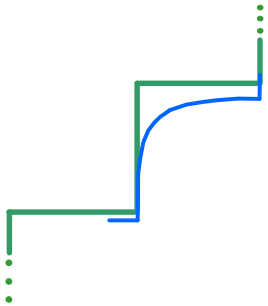
DAC Performance Issues and Concerns



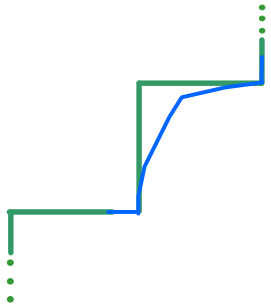
Ideal



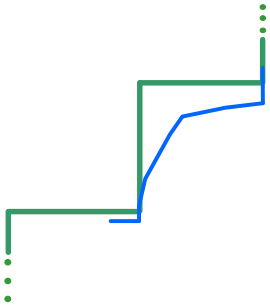
Complete
Linear Settling



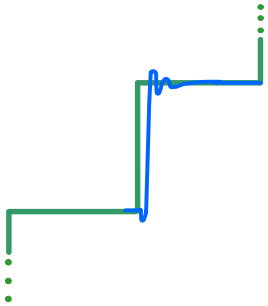
Incomplete
Linear Settling



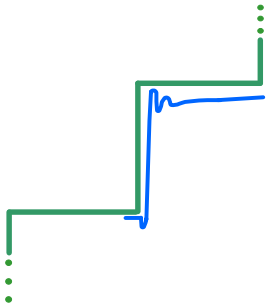
Complete Nonlinear
Settling



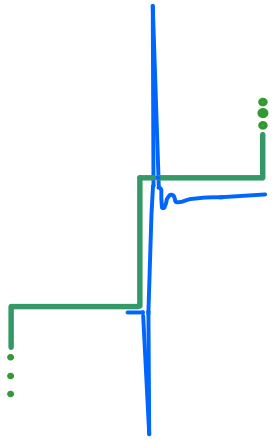
Incomplete
Nonlinear Settling



Complete with glitch

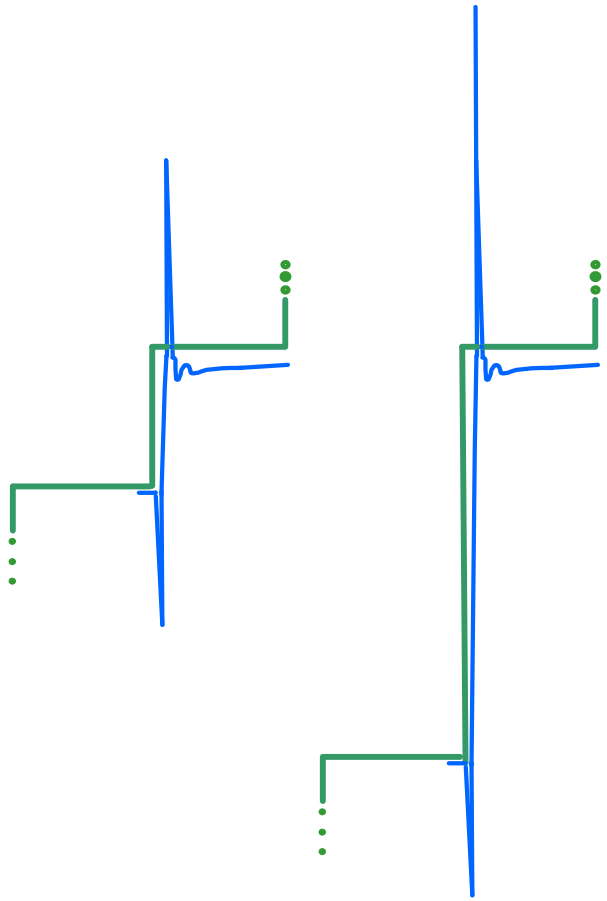


Incomplete with
glitch

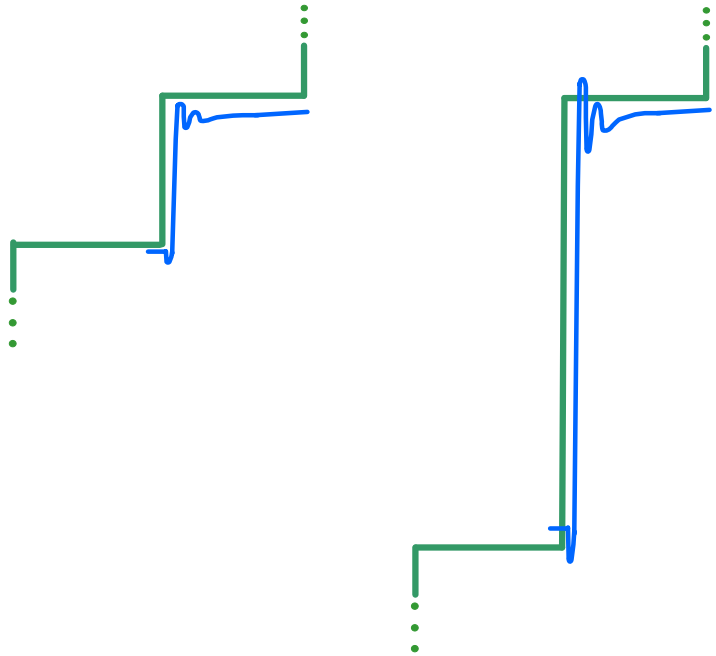


Incomplete with big
glitch

DAC Performance Issues and Concerns



Previous code dependent glitches



Previous code dependent settling

DAC Performance Issues and Concerns

Linear settling of DAC outputs do not affect linearity if all have same settling times (for both sampled outputs and overall transient response

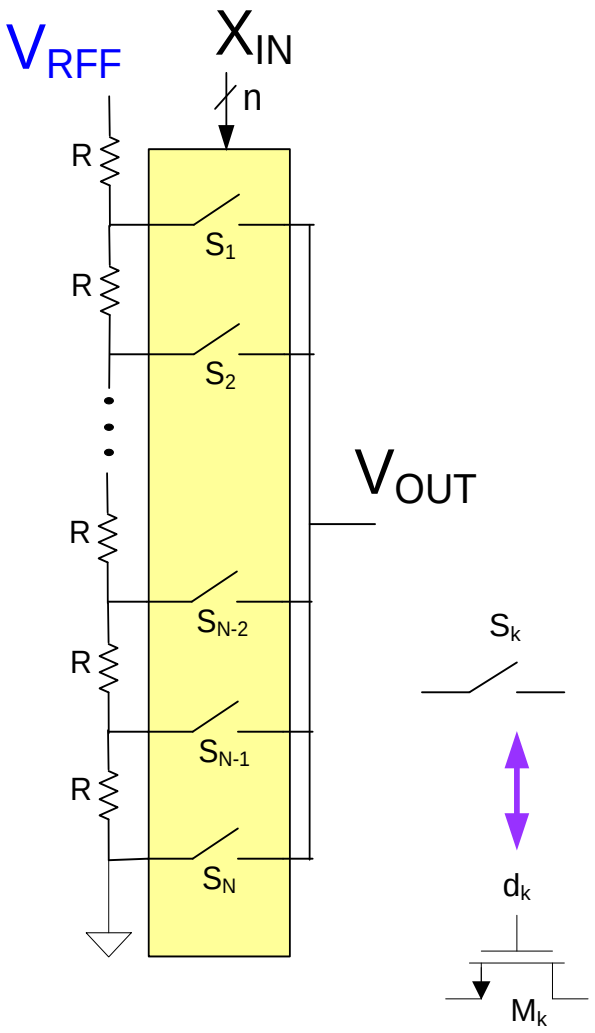
Incomplete settling introduces nonlinearities in transient response and usually in settled response

Previous code dependent outputs or settling almost always introduces nonlinearities

Glitches can be many LSB in magnitude and are often previous-code dependent

Glitches in output at transition points do not introduce nonlinearities in settled outputs but may introduce distortion in continuous-time outputs

R-String DAC



Conceptual

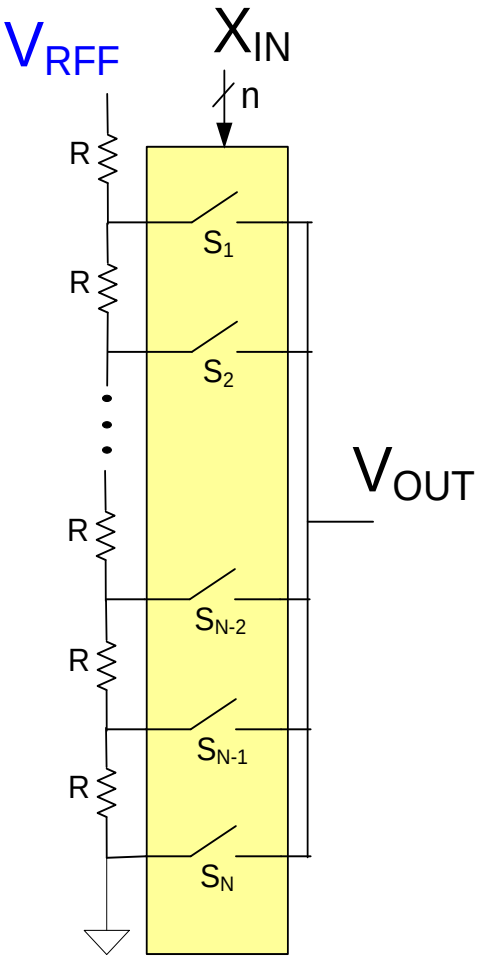
- Simple structure
- Inherently monotone
- Very low DNL
- Potential for being very fast
- Low Power Dissipation
- Widely Used Approach (with appropriate considerations)

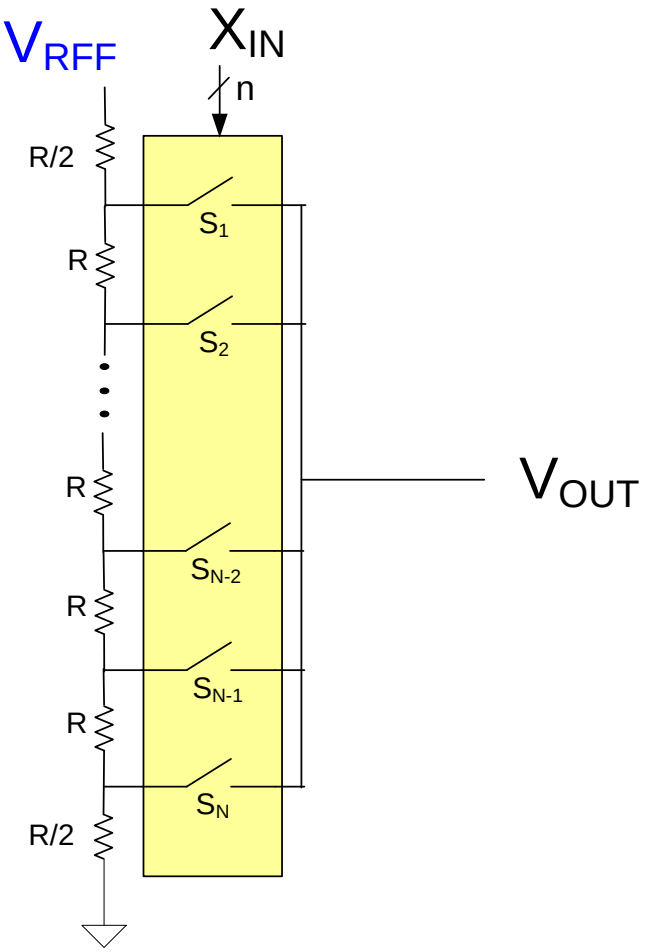
Challenges:

- Managing INL
- Matching (resistors, switches)
- Leakage currents
- Large number of devices for n large (2^n or 2^{n+1} lines)
- Decoder
- Routing thermometer/bubble clocks
- Transients during Boolean transitions
- Glitches
- Switch implementation
- Thevenin impedance facing V_{OUT} highly code dependent

R-String DAC

(minor variant where $V_{OUT}(0,\ldots 0) \neq 0$)



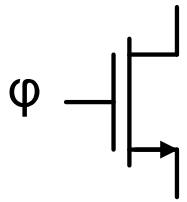


Practical level shift

Switch Implementation

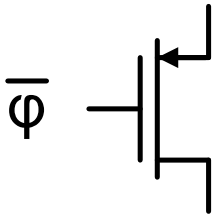
Basic Switch

- Large number required for large resolution
- Simple structure often used



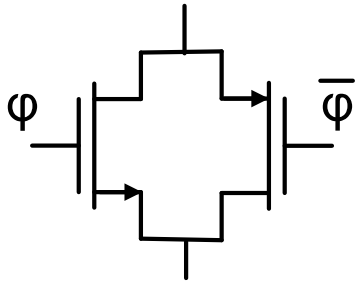
single n-channel device

- Good when switch terminals near gnd
- Will not turn on when terminals near V_{DD}



single p-channel device

- Good when switch terminals near V_{DD}
- Will not turn on when terminals near gnd

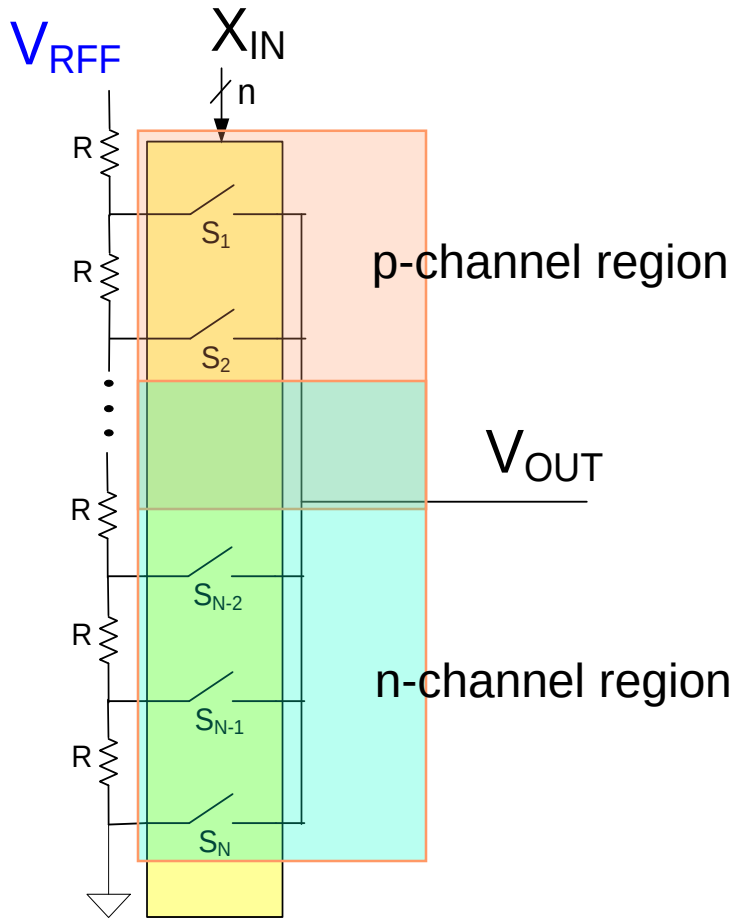


transmission gate switch

- Use devices where cross-over occurs
- Good for both high and low term voltages
- Extra clock signal required
- Try to avoid this complexity

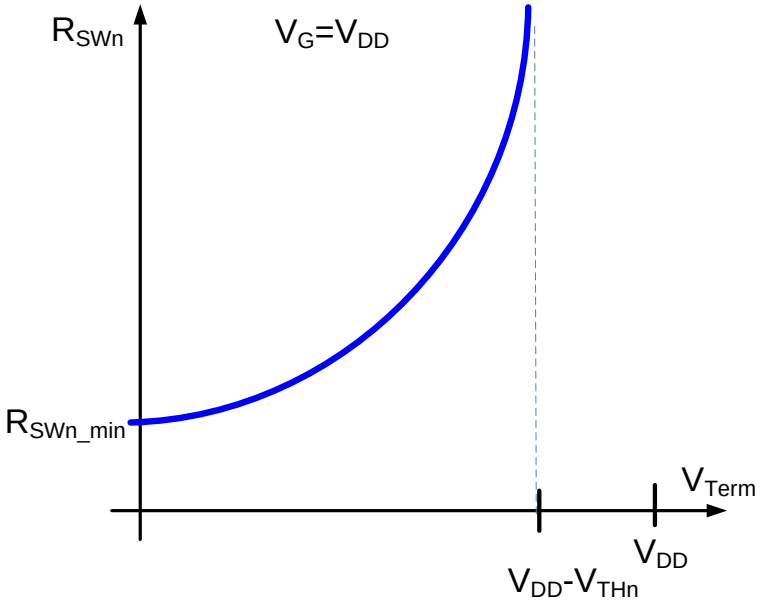
Other switch structures (such as bootstrapped switch) used but not for basic string DACs

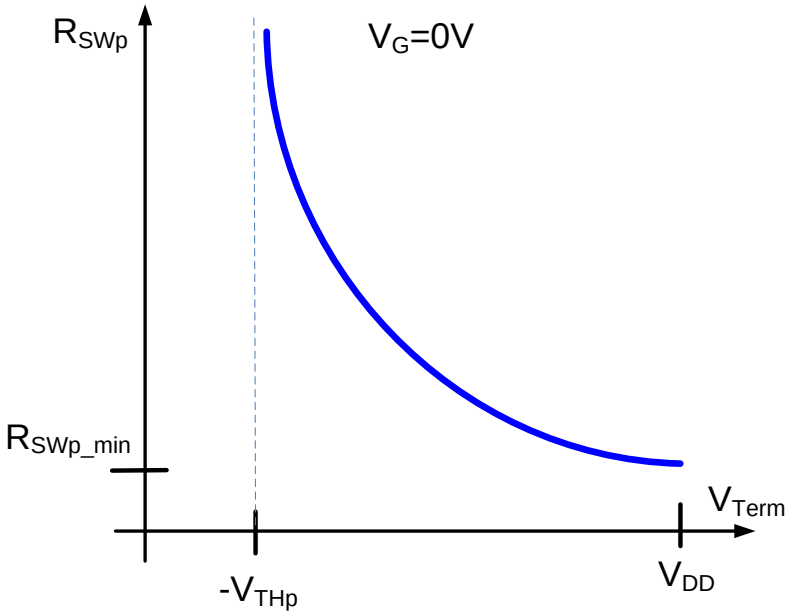
Switch Assignment

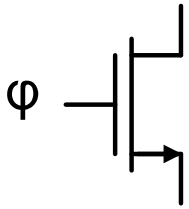


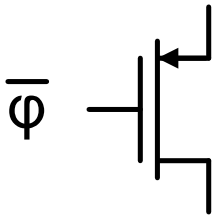
Challenges:

Switch Impedances

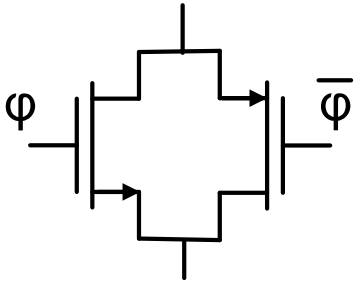


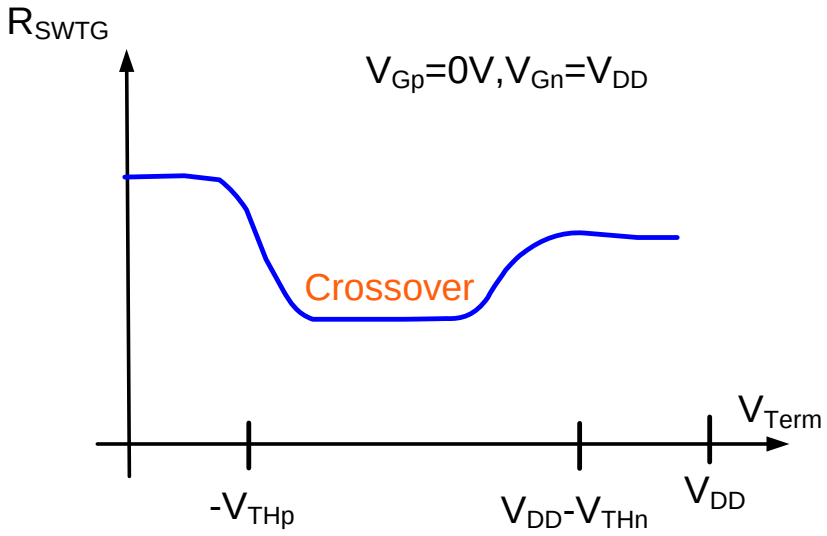


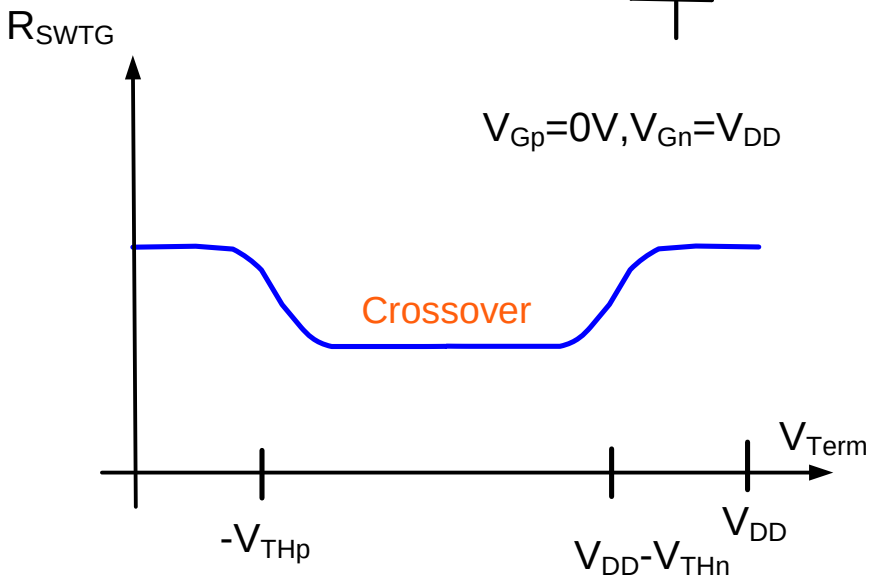


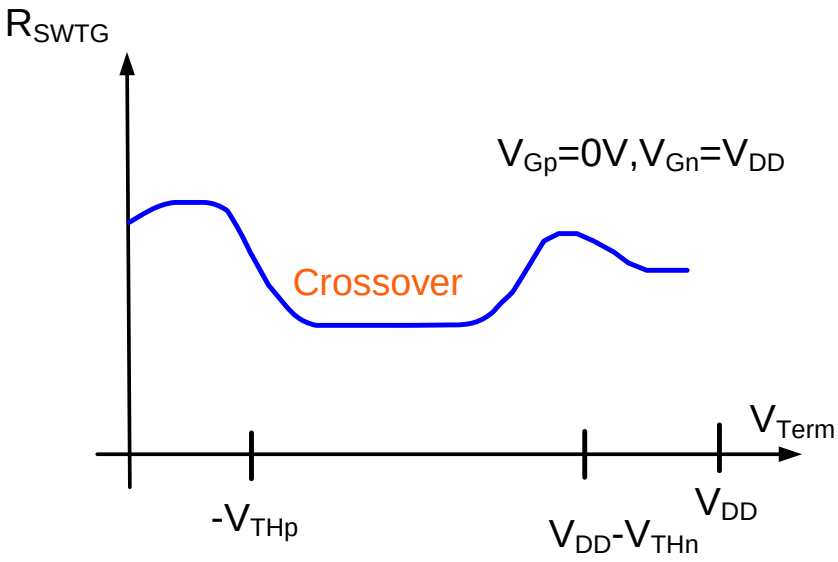


Switch Impedances



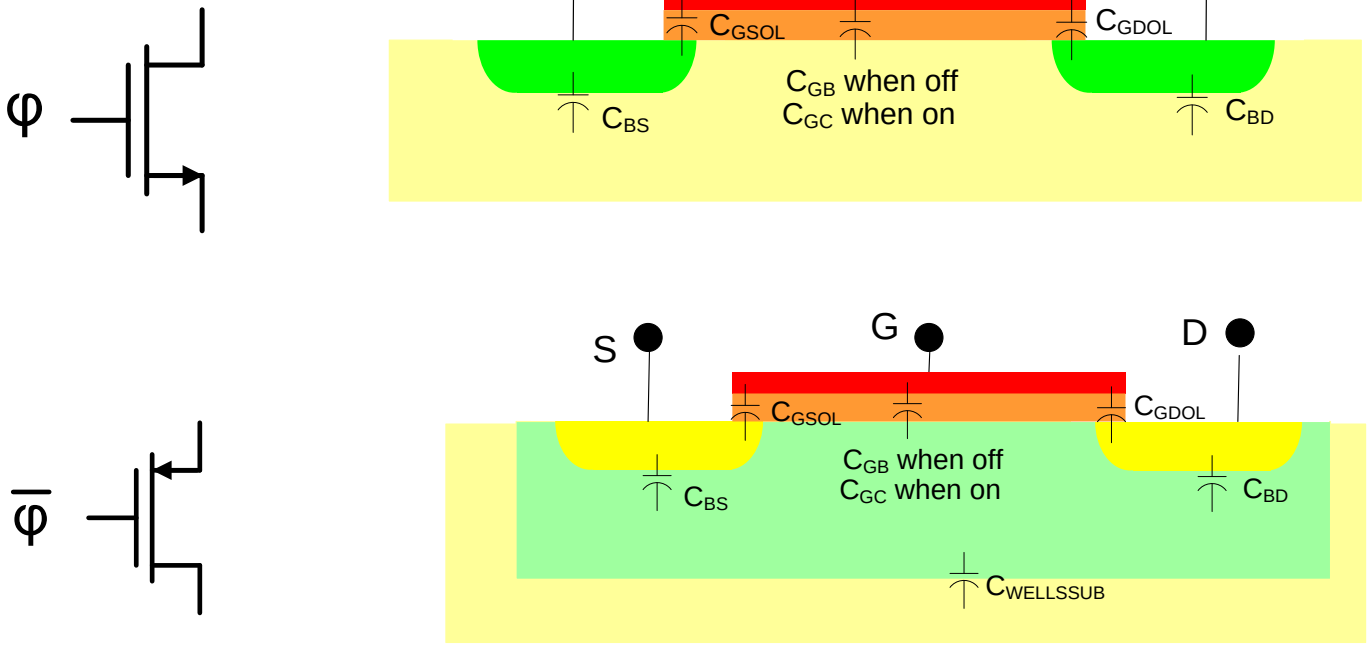






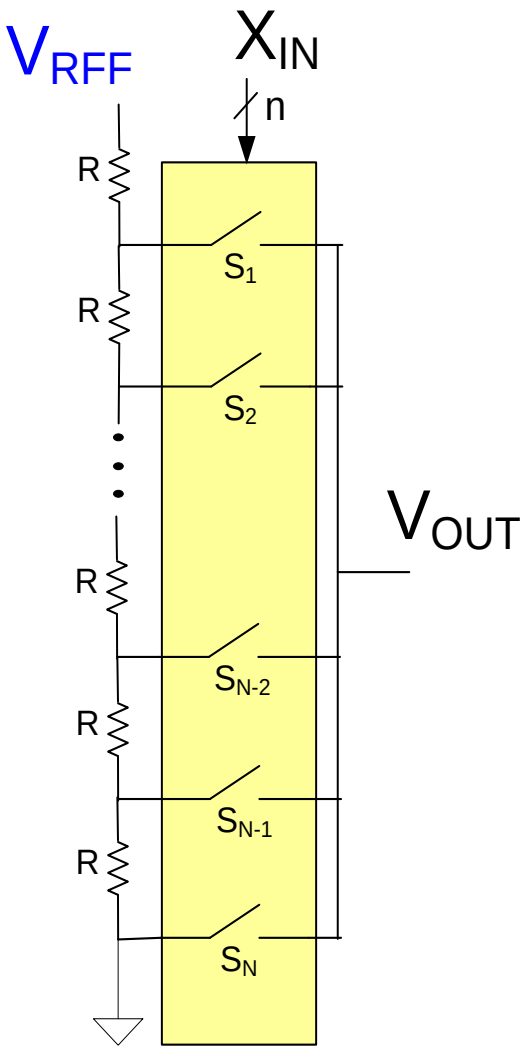
Switch impedance significantly both position and device size dependent

Switch Parasitics



- C_{BD} and C_{BS} can be significant and cause rise-fall times to be position dependent
- C_{GDOL} can cause “kickback” or feed-forward
- C_{GS} can slow turn-on and turn-off time of switch

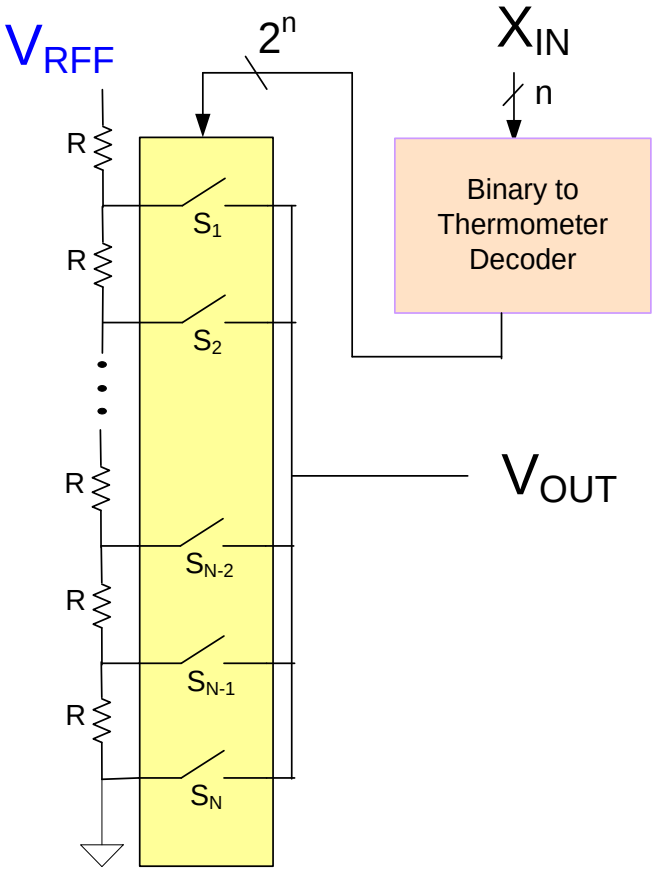
R-String DAC



Additional Challenges:

- Capacitance on V_{OUT} can be large
  - larger for p-channel devices
  - even larger for TG switches
- Switch impedances position dependent
- Kickback from switches to R-string
- Capacitance on each node (though small) of R-string from switch
- Thevenin impedance facing V_{OUT} highly code dependent
- Gradient effects may cause nonlinearities since common-centroid layout may not be practical if n is large

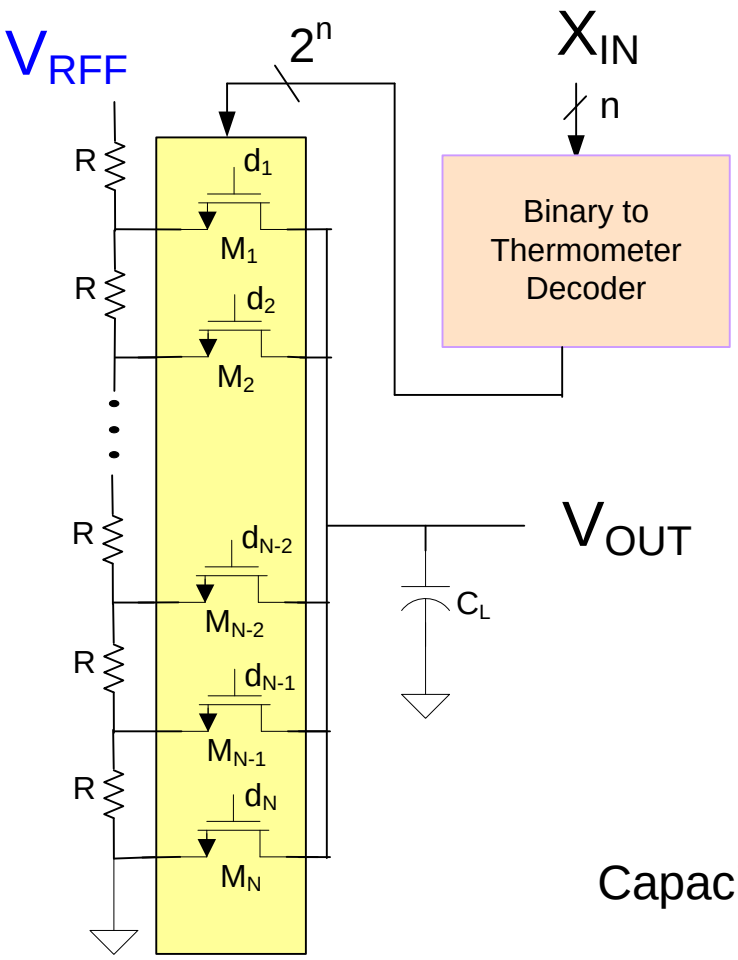
R-String DAC



Additional Challenges

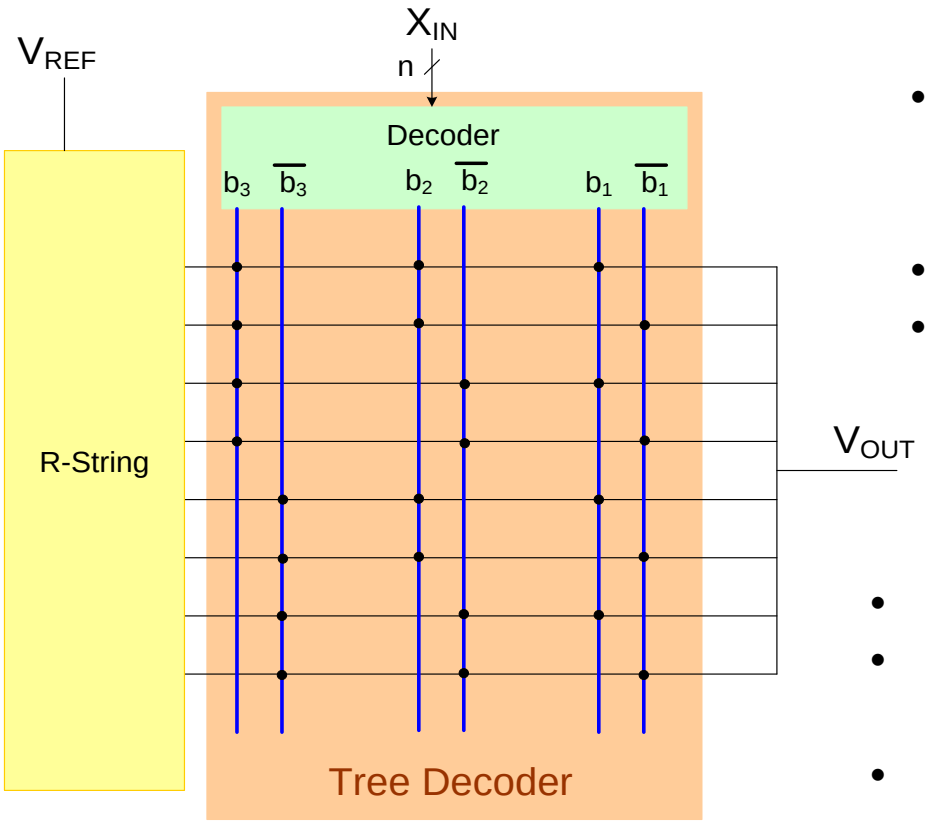
- Delay in Decoder may be significant
- Delay in Decoder may be previous code and current code dependent
- Intermediate undesired Boolean outputs may occur
  - These may cause undesired opening and closing of switches
  - Could momentarily short out taps on R-string
  - Could introduce transients on all nodes of R-string that are code and previous code dependent

R-String DAC



Capacitive loading due to switches

R-String DAC

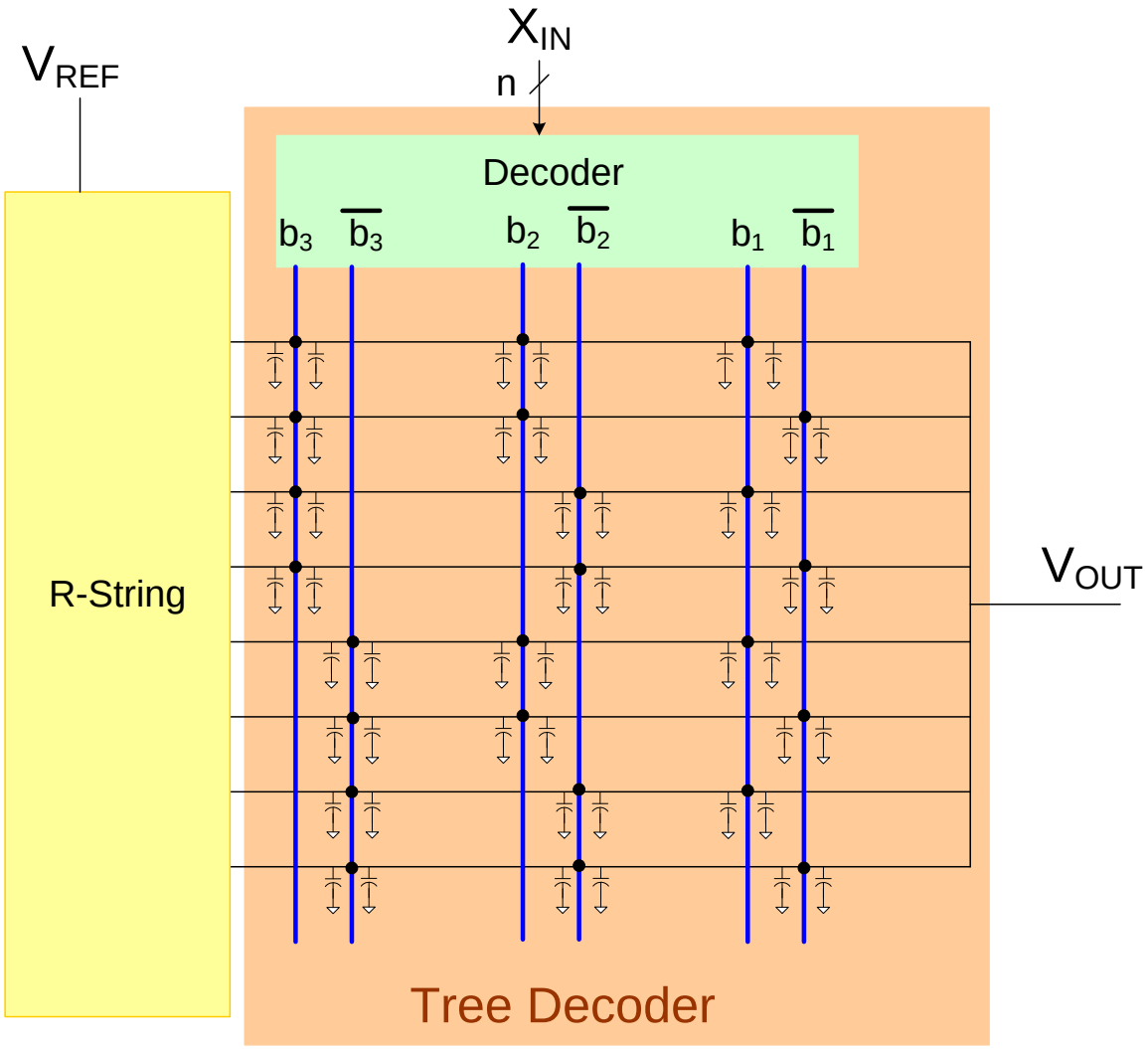


- Uses matrix decoder as analog MUX (don't synthesize decoder)
- Implements binary to decimal conversion with pass transistor analog logic
- Very structured layout
- Interconnection points are switches (combination of n-channel and p-channel)

Challenges

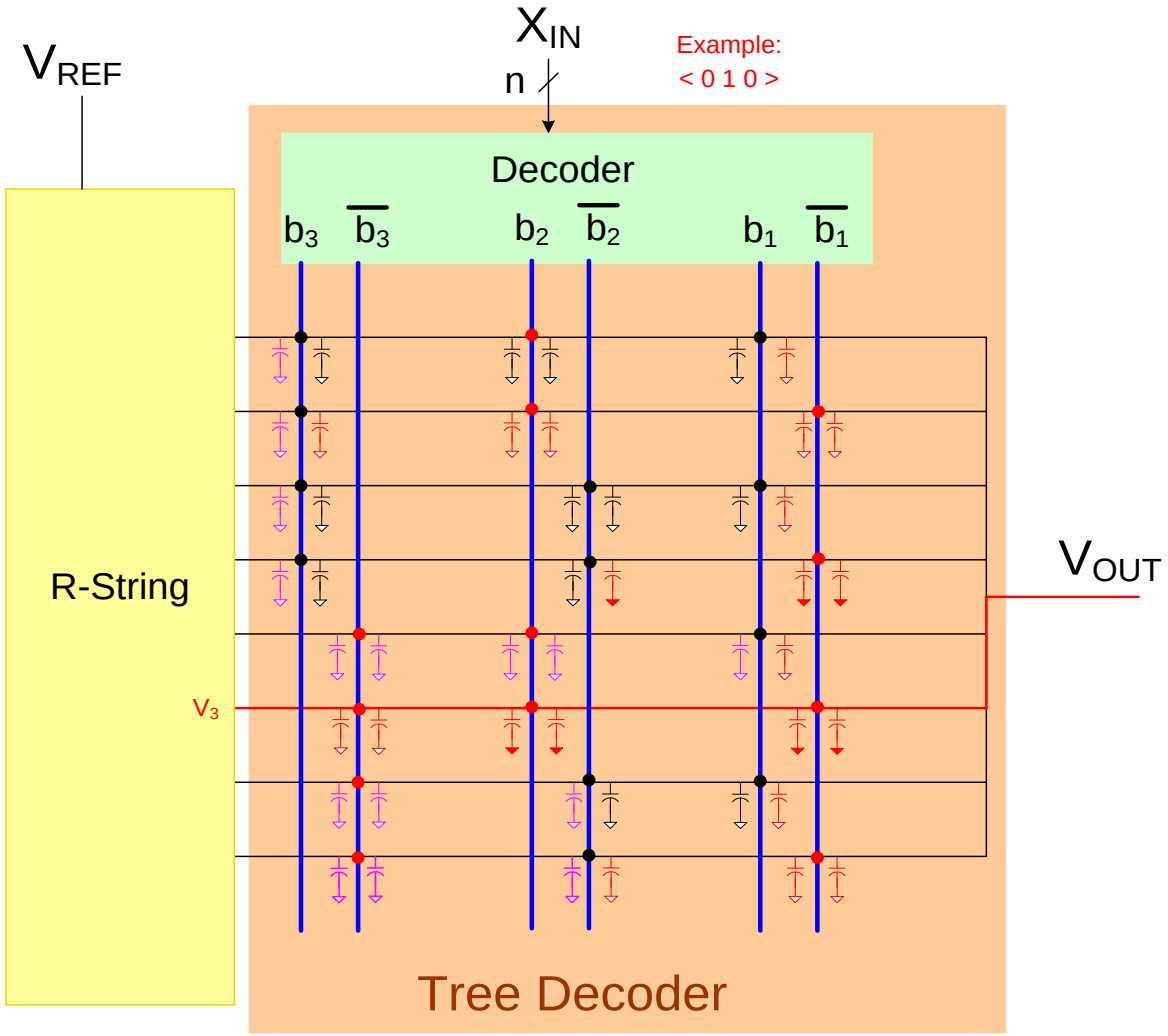
- Still many signals to route
- Large capacitance on V_{OUT} (over 2^{n+1} diff caps)
- Multiple previous code dependencies cause output transition time to be quite unpredictable
- Considerable transients introduced on R-string

R-String DAC



Parasitic Capacitances in Tree Decoder

R-String DAC



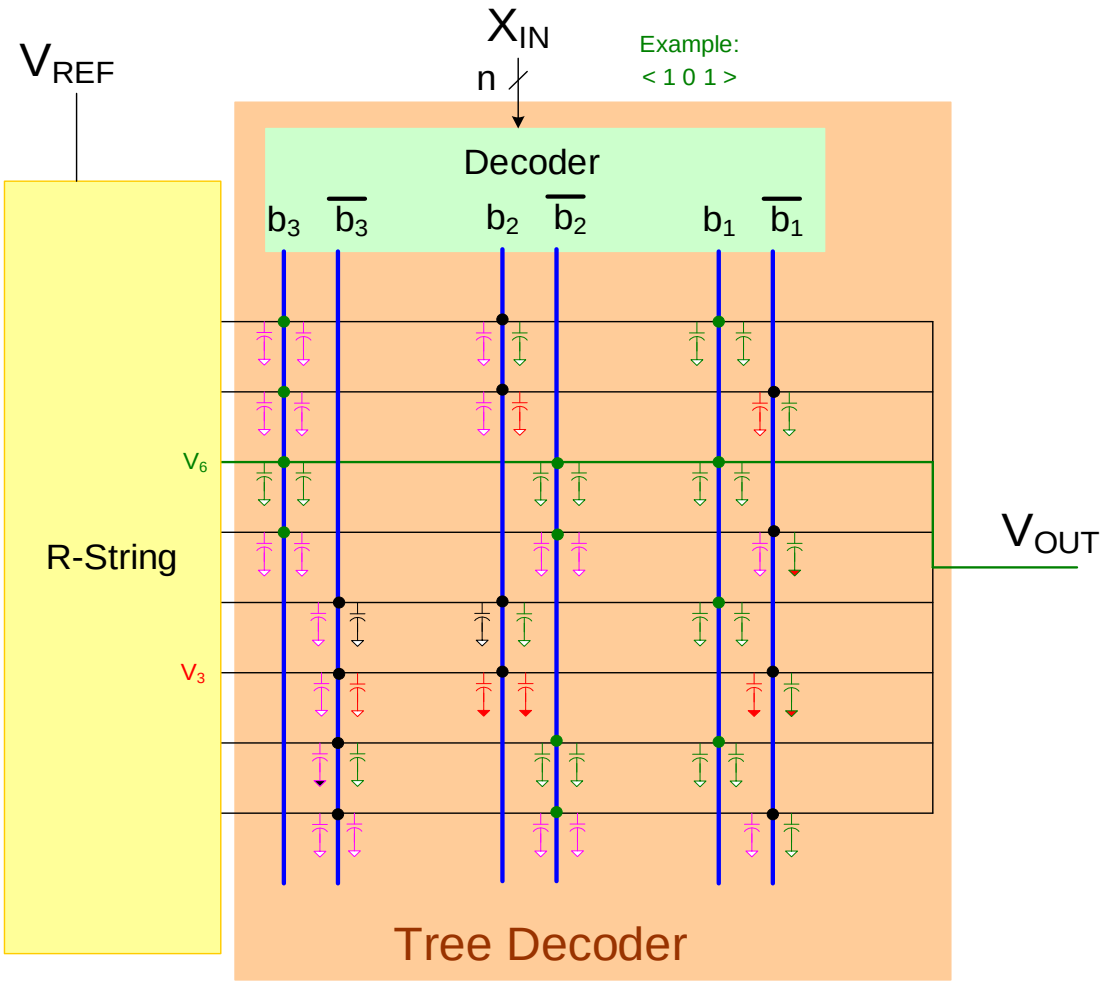
Previous-Code Dependent Settling

Assume all C's (except those on the R-string) initially with 0V

Red denotes V_3, black denotes 0V, Purple some other voltage

R-String DAC

Transition from <010> to <101>



Previous-Code Dependent Settling

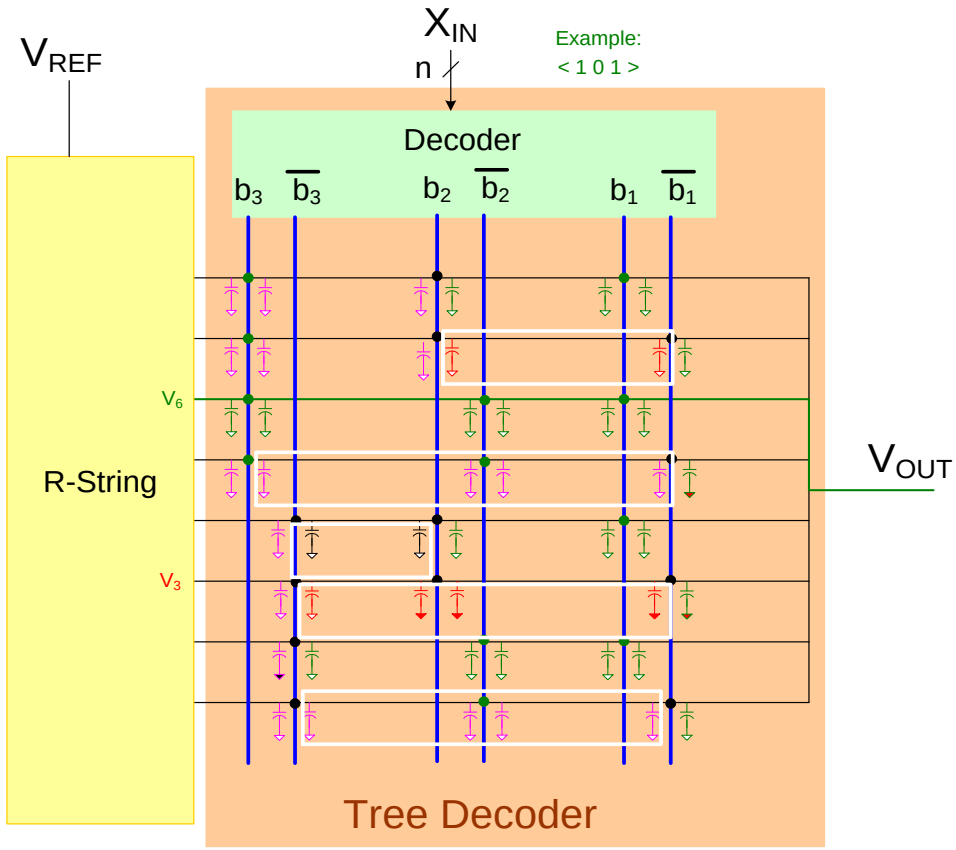
Assume all C's (except those on the R-string) were initially at 0V

Red denotes V_3, green denotes V_6, black denotes 0V, Purple some other voltage

R-String DAC

Transition from <010> to <101>

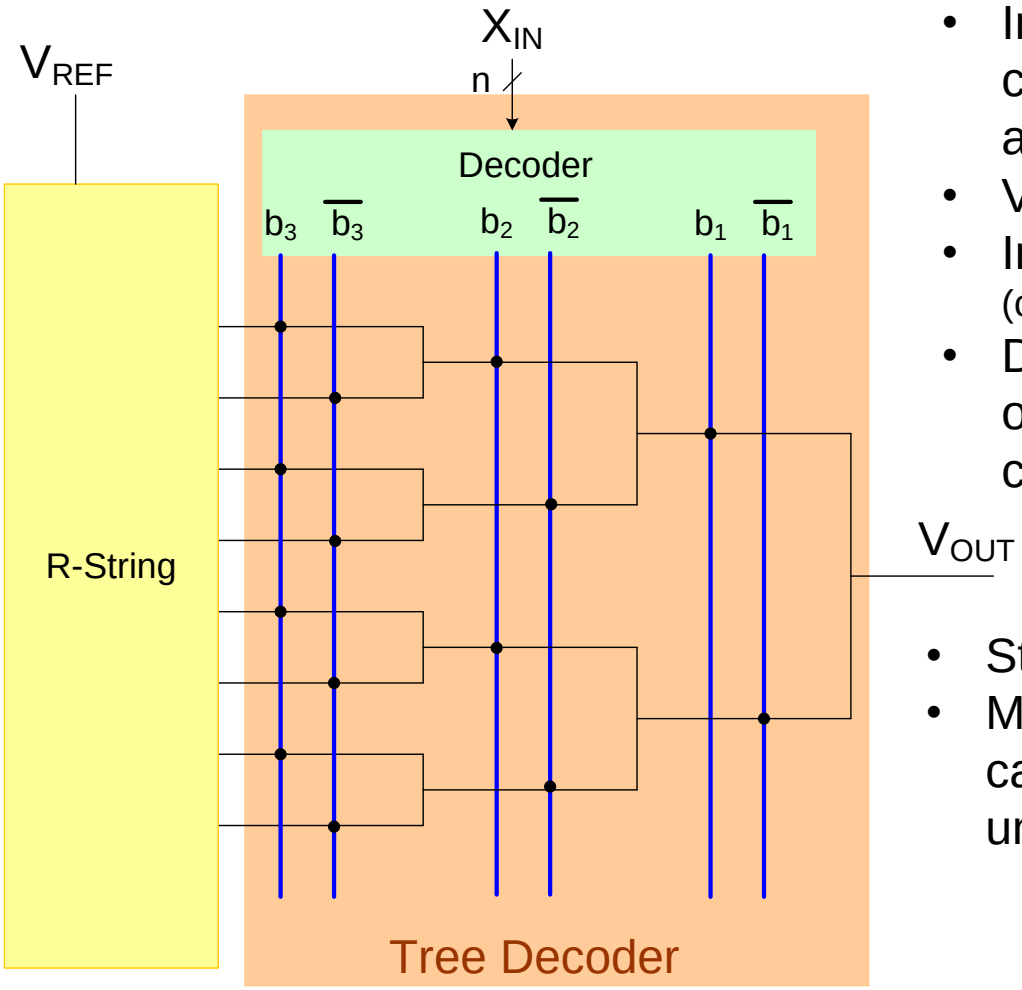
White boxes show capacitors dependen
upon previous code <010>



Previous-Code Dependent Settling

- Assume all C's (except those on the R-string) were initially at 0V
- Red denotes V_3, green denotes V_6, black denotes 0V, Purple some other voltage
- Some capacitors may retain values from a previous input for many clock cycles for some inputs resulting I previous-previous dependence of even longer

R-String DAC

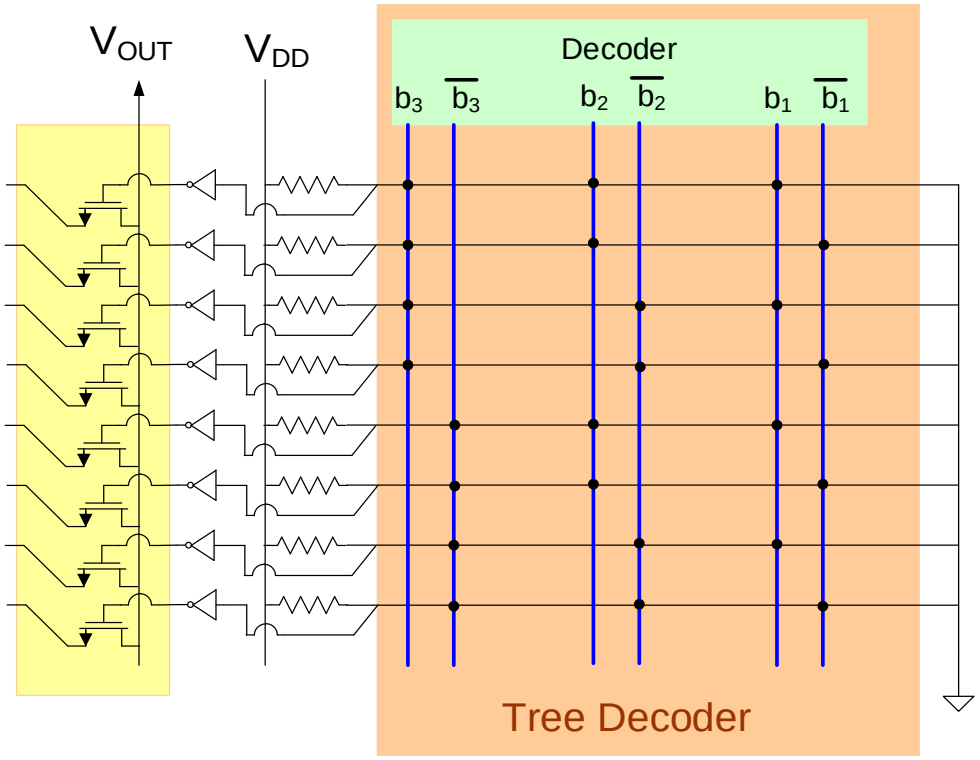


- Uses tree decoder as analog MUX
- Implements binary to decimal conversion with pass transistor analog logic
- Very structured layout
- Interconnection points are switches (combination of n-channel and p-channel)
- Dramatically reduces capacitance on output and switching capacitances

Challenges

- Still many signals to route
- Multiple previous code dependencies cause output transition time to be quite unpredictable

R-String DAC



Tree-Decoder in Digital Domain

Single transistor used at each marked intersection for PTL AND gates

Dramatic reduction in capacitive loading at output

Do the resistors that form part of PTL dissipate any substantial power?

No because only one will be conducting for any DAC output

Will become more complicated if both p-channel and n-channel switches needed

Stay Safe and Stay Healthy !

End of Lecture 13