EE 435

Lecture 34

DAC Design

- String DACs
DAC Architectures

- Type of Classification may not be unique nor mutually exclusive
- Structure is not mutually exclusive
- All approaches listed are used (and probably some others as well)
- Some are much more popular than others
  - Popular Architectures
    - Resistor String (interpolating)
    - Current Source Steering (with segmentation)
- Many new architectures are possible and some may be much better than the best currently available
- All have perfect performance if parasitic and matching performance are ignored!
- Major challenge is in determining appropriate architecture and managing the parasitics
Identifying Problems/Challenges and Clever/Viable Solutions

- Many problems occur repeatedly so should recognize what they are
- Identify clever solutions to basic problems – they often are useful in many applications
- Don’t make the same mistake twice!

The problem:

The perceived solution:

The practical or clever solution:

The List Keeper!
R-String DAC

Basic R-String DAC
R-String DAC

R-String DAC with MOS switches

Possible Limitations:

- Switch impedance is not 0
- Switch may not even turn on at all if $V_{\text{REF}}$ is large
- Switch impedance is input-code dependent
- Time constants are input-code dependent
- Transition times are previous-code dependent
- $C_L$ has $2^n$ diffusion capacitances so can get very large
- Mismatch of resistors
  - local random variation
  - gradient effects
- Decoder can get very large for $n$ large
- Routing of the $2n$ switch signals can become very long and consume lots of area

Review from last lecture:
Basic R-String DAC

\[ V_{RFF} \]

\[ 2^n \]

\[ X_{IN} \]

\[ n \]

Binary to Thermometer Decoder

\[ V_{OUT} \]
Parasitic Capacitances in Tree Decoder
R-String DAC

Previous-Code Dependent Settling

Assume all C’s initially with 0V
Red denotes $V_3$, black denotes 0V, Purple some other voltage

Example:
$V_3$
R-String DAC

Transition from \langle 010 \rangle \text{ to } \langle 101 \rangle

Assume all C’s initially with 0V

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

Previous-Code Dependent Settling

Example:

Transition from \langle 010 \rangle \text{ to } \langle 101 \rangle
R-String DAC

Transition from <010> to <101>

White boxes show capacitors dependent upon previous code <010>

Assume all C’s initially with 0V
Red denotes $V_3$, green denotes $V_6$, black denotes 0V, Purple some other voltage
R-String DAC

Tree-Decoder in Digital Domain
R-String DAC

\[ V_{REF} \]

R-String

\[ X_{IN} \]

Tree Decoder

\[ V_{OUT} \]
R-String DAC

\[ X_{IN} \]

\[ n = n_1 : n_2 \]

\[ V_{RFF} \]

\[ V_{OUT} \]
End of Lecture 34