EE 505

Lecture 16

String DACs
Example

\[ g(x) = x^2 \]

What is the variance of \( x^2 \)?

Note: The statistical functions of interest are highly nonlinear functions of many random variables.

Note: In general, obtaining statistics of a multivariate random variable that is nonlinear is very difficult.

"Linearization and decorrelation" was used to address this problem.

Stopped after we obtained the variance.
\[ g = g_N + g_R \quad \text{where} \quad g_R \ll g_N \]

\[ g = x^2 \]

\[ x = (x_N + x_R) \]

\[ g = (x_N + x_R)^2 = x_N^2 + 2x_Nx_R + x_R^2 \]

\[ \gamma^2 = x_N^2 \left( 1 + 2 \frac{x_R}{x_N} \right) \]

\[ \sigma_g = 4x_N^2 \sigma_{x_R} \]
If $g \sim N(0, \sigma)$

$$P \left\{ -g_1 \leq g \leq g_1 \right\} = \int_{-g_1}^{g_1} f(g) \, dg$$

Let $h = g / \sigma$  \hspace{1cm} $h \sim N(0, 1)$

$$P \left\{ -g_1 \leq g \leq g_1 \right\} = P \left\{ -\frac{g_1}{\sigma} \leq h \leq \frac{g_1}{\sigma} \right\}$$

$$P \left\{ -h_1 \leq h \leq h_1 \right\} = \int_{-h_1}^{h_1} f_{N_1}(h) \, dh$$

$$= \int_{-h_1}^{h_1} f_{N_1}(h) \, dh - \int_{-\infty}^{-h_1} f_{N_1}(h) \, dh$$

$$= \Phi(h_1) - \left( 1 - \Phi(h_1) \right)$$
\[ P\{-h_1 \leq h \leq h_1\} = 2F(h_1) - 1 \]

Case 1. If \(-g_1 + g_1\) are the limits for acceptance of a parameter, then \(Y = 2F(h_1) - 1\) where \(h_1 = \frac{g_1}{\delta}\)

Case 2. If \(Y\) is known and want to determine \(-g_1 + g_1\),

Solve \(Y = 2F(h_1) - 1\) for \(h_1\)

Solve \(h_1 = \frac{g_1}{\delta}\) for \(g_1\)
If a product must meet several performance requirements and if the mechanisms that cause yield loss are uncorrelated between those mechanisms, then

\[ Y = \prod_{i=1}^{k} Y_i \]

where \( Y_i \) is the yield for performance specification \( i \).
\[ I_1 = (D \times V_{REF})(K) \]

\[ V_0 = -I_1R \]

\[ \therefore V_0 = -R(KD) \times V_{REF} \]

\[ = D(F - RK) \times V_{REF} \]
\[ I_2 = \frac{V_0 D_1 K}{D_2} \]
\[ V_i = -I_2 R \]
\[ V_i = -V_0 D_1 K R \]

\[ V_0 = +V_i \left( \frac{1}{D_2} \right) \left( \frac{-1}{KR} \right) \]

Could call this \( \frac{V_0}{D_1} \) DAC

\[
\begin{align*}
I_1 &= K_1 V_i D_1 \\
I_2 &= V_0 D_2 K_2
\end{align*}
\]

\[ V_0 = \frac{D_1}{D_2} \left[ \frac{-K_1 V_i}{K_2} \right] \]
R-String DAC

\[ V_{\text{REF}} \]

\[ R_n \]

\[ V_{n-1} \]

\[ I_{\text{REF}} \]

\[ R_1 \]

\[ V_1 \]

\[ V_0 \]
$2^n : 1$ analog decoder

Logic required for a $2^n : 1$ Mux can get very involved.
Tree Decoder.

\[ V_{n-1} \]

\[ V_0 \]

\[ \begin{align*} b_{n-1} & \quad \overline{b}_{n-1} \\ \overline{b}_2 & \quad \overline{b}_2 \end{align*} \]

\[ b_1 \quad \overline{b}_1 \quad b_0 \quad \overline{b}_0 \]

\[ V_0 \]
$2^n$ switches

$2^{n+1}$ switch sites

+ Very simple decoding structure
+ Very structured layout
+ Requires $N \cdot 2^n$ switches
+ $n$ levels of switches
+ $2^n$ diffusions on each output node
- Response time is previous code dependent
Review

$V_{REF}$

$R_5$

$R_4$

$R_3$

$R_2$

$R_1$

$U_0$

$D_i n$
- Code-dependent Delay

- Previous Code Dependent Delay

"Gotcha" list
Alternate Tree Decoder

2^n \left( \sum_{i=0}^{n-1} \frac{1}{2^i} \right) \text{ switches}

= 2^n \left( 2 - \frac{1}{2^{n-1}} \right) = 2^{n+1} - 2 = 2 \left( 2^n \right)

\rightarrow \text{ decoder simply } n \text{ inverters}

\rightarrow \text{ modest decrease in number of switches but not necessarily in total switch area}

\rightarrow \text{ total capacitive switch loading reduced by factor of } n/2
Tree Decoder

\[ b_n \bar{b}_{n-1} b_{n-1} \bar{b}_{n-2} b_{n-2} \bar{b}_{n-3} b_{n-3} \bar{b}_{n-4} b_{n-4} \bar{b}_{n-5} b_{n-5} \bar{b}_{n-6} b_{n-6} \bar{b}_{n-7} b_{n-7} \bar{b}_{n-8} b_{n-8} \bar{b}_{n-9} b_{n-9} \bar{b}_{n-10} \]

- \( n \cdot 2^n \) switches
- Decoders simply \( n \) inverters and may not be needed if differential logic is used.
- Number of switches not too bad since \( n \) is not large for R-string DACs.
- Total capacitive loading on R-string is large but distributed.
In switches are ideal

1) Assume at code $<0000 \ldots>$

$T = 0$

2) Assume at code $<00 \ldots 01>$

$R_{eq} = R \parallel (2^{n-1})R = R$

$\therefore T = R_{eq}C_L = RC_L$

3) Assume at code $<100 \ldots 0>$

$R_{eq} = \frac{2^n}{2} R \parallel \frac{2^n}{2} R = 2^{n-2} R$
If $n = 10$, $R_{eq} = 256 R$

$k = R_{eq} C_L = 2^{n-2} R C_L$

If $n = 10$, $k = 256 R C_L$

Dramatically larger!

This will cause significant spectral degradation.
If previous code was 000
next code is 110
If previous code was 101
next code is 110
- reduction in number of analog switches
- major routing requirements
- large C-loading on output node
- code-dependent time constants
- code-dependent switch impedance
  - transmission gate switches (but too much cap. on this structure)
  - size switches to compensate for change in impedance
  - use p-mos rather than n-mos for upper switches