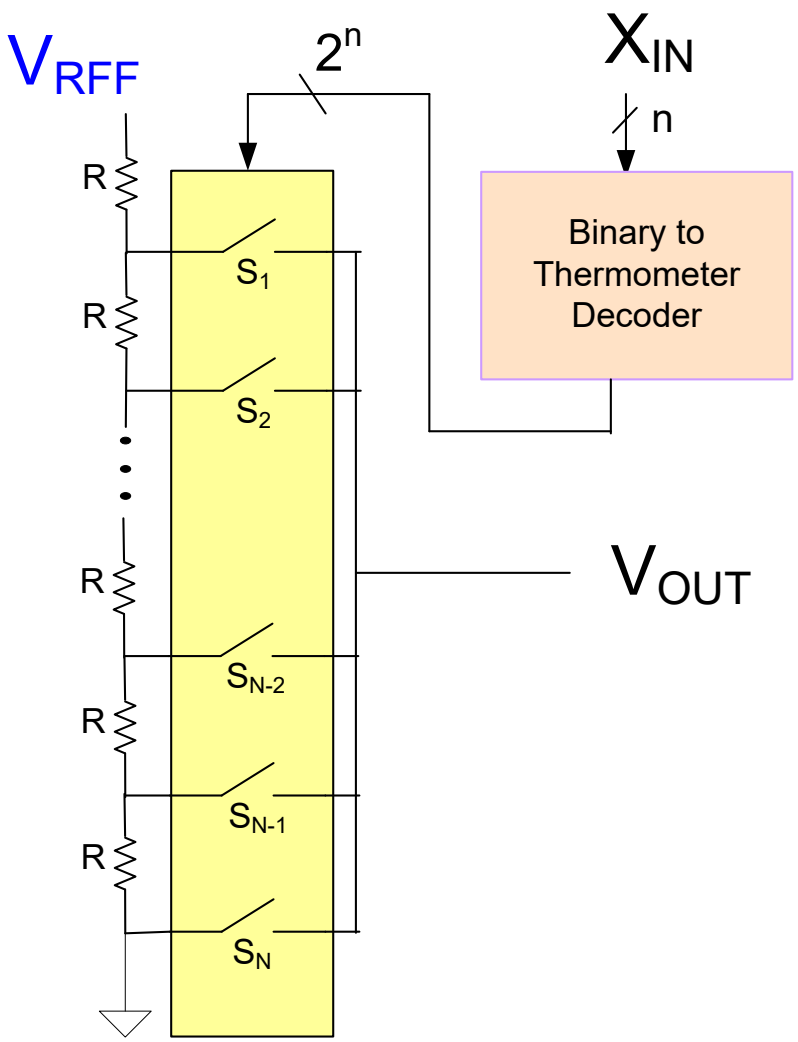
# EE 435

# Lecture 32

- Parasitic Capacitances
- String DACs

Review from Last Lecture

# R-String DAC




**Basic R-String DAC including Logic to Control Switches**

Review from Last Lecture

# Nonideal Effects of Concern

- Matching
- Parasitic Capacitances
  (including Charge injection)
- Loading
- Nonlinearities
- Previous code dependence
- Code-dependent settling
- Interconnect resistors
- Noise
- Slow and plagued by jitter
- Temperature Effects
- Aging
- Package stress

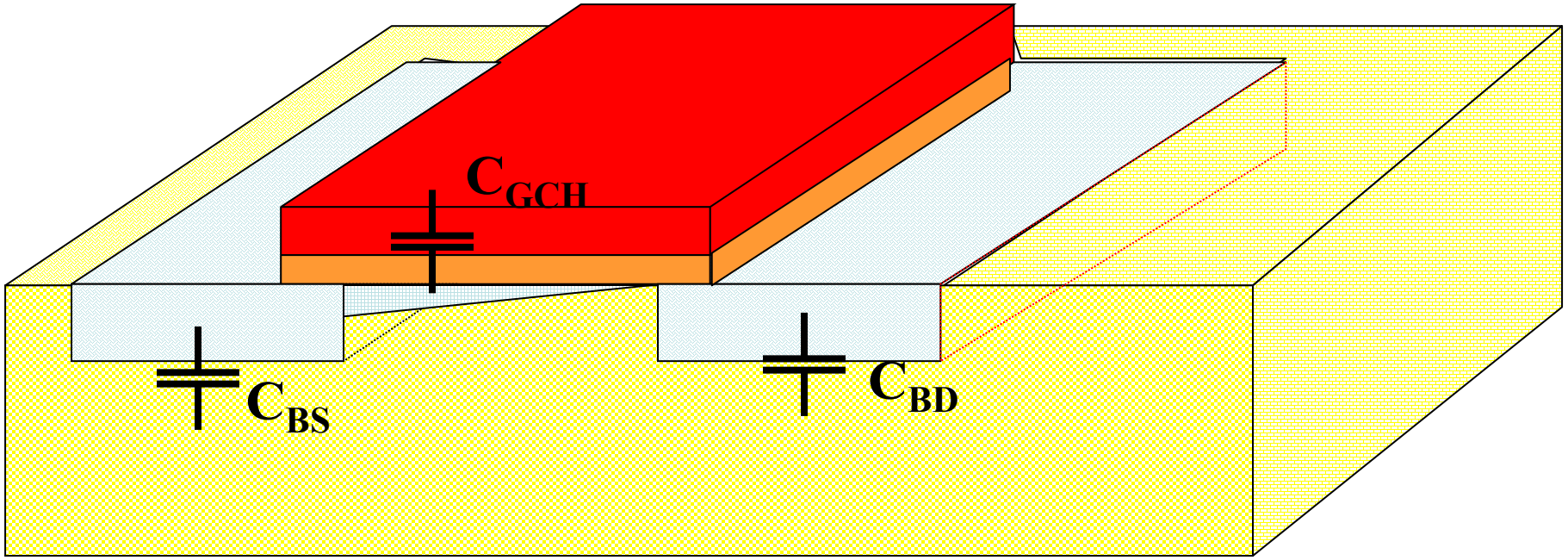
# Data Converter Design Strategies

- There are many different DAC and ADC architectures that have been proposed and that are in widespread use today
- Almost all work perfectly if all components are ideal
- Most data converter design work involves identifying the contributors to nonideal performance and finding work-arounds to these problems
- Some architectures are more difficult to find work-arounds than others
- All contributors to nonidealities that are problematic at a given resolution of speed level must be identified and mitigated
- The effects of not identifying nonidealities generally fall into one of two categories
  - Matching-critical nonidealities (degrade yield)
  - Component nonlinearities (degrade performance even if desired matching is present)

Review from Last Lecture

# Parasitic Capacitors in MOSFET

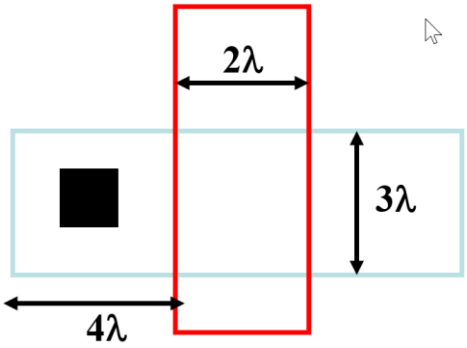
(initially assume saturation and consider two: Gate-channel and diffusion)




- Diffusion capacitances nonlinear (dependent upon voltage)
- Many more actually present
- Operating region may affect parasitics

Review from Last Lecture

# Size of Capacitances




**Gate-Channel Capacitance = $6\lambda^2$ x 2.47fF/$\mu^2$ = 1.33fF**

**Source Diffusion-Substrate Capacitance =**
**$12\lambda^2$ x .424fF/$\mu^2$ + 14$\lambda$ x .315fF/$\mu$ =**
**.46fF + 1.32fF =1.78fF**

Note Sidewall Capacitance larger than Bottom Capacitance

Are these negligible?

# Reminder !!

## Identifying Problems/Challenges and Clever/Viable Solutions

- Many problems occur repeatedly so should recognize when they occur
- 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 !**

**The List Keeper !**

# Types of Capacitors

1. Fixed Capacitors
   a. Fixed Geometry
   b. Junction

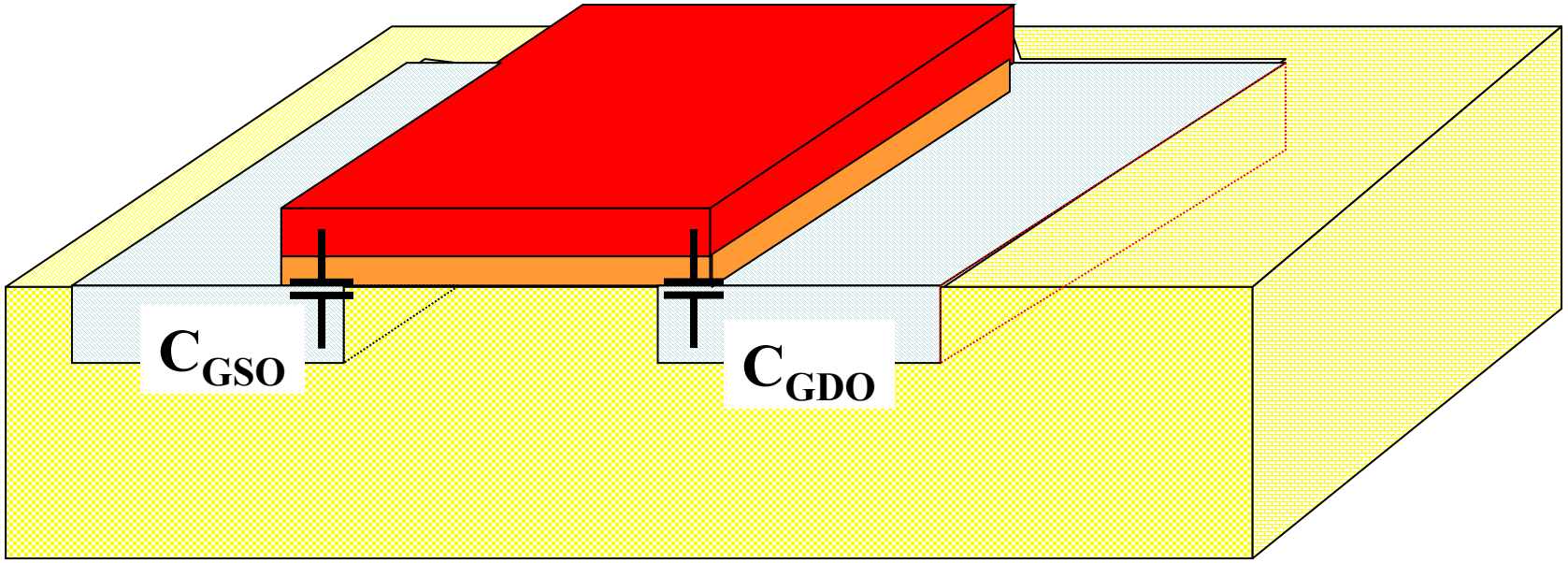
2. Operating Region Dependent
   a. Fixed Geometry
   b. Junction

# Parasitic Capacitors in MOSFET

## Fixed Capacitors

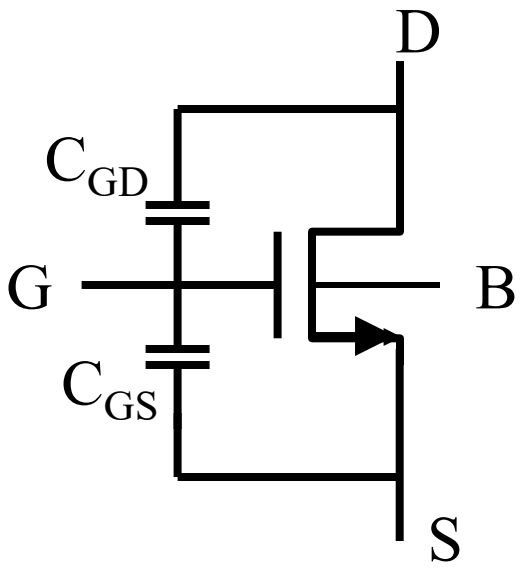
# Parasitic Capacitors in MOSFET

## Fixed Capacitors




Overlap Capacitors: $C_{GDO}$, $C_{GSO}$

# Fixed Parasitic Capacitance Summary

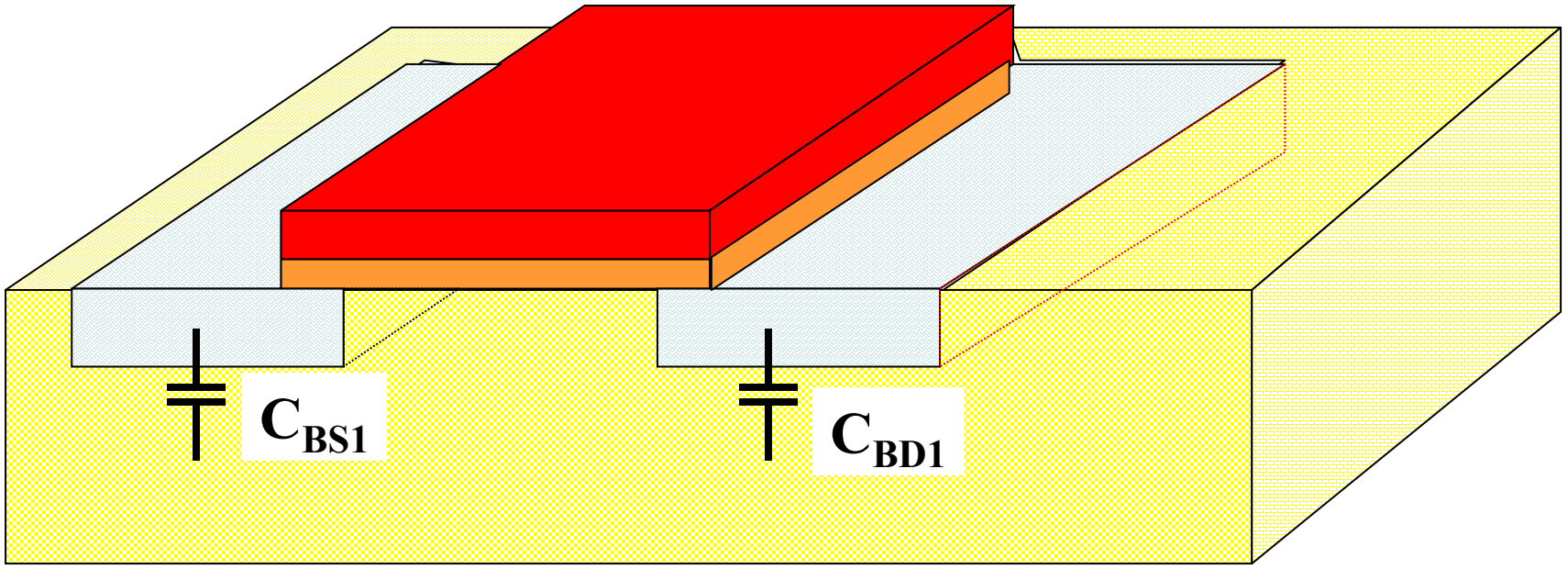



| | Cutoff | Ohmic | Saturation |
|---|---|---|---|
| $C_{GS}$ | $CoxWL_D$ | $CoxWL_D$ | $CoxWL_D$ |
| $C_{GD}$ | $CoxWL_D$ | $CoxWL_D$ | $CoxWL_D$ |

$L_D$ is a model parameter

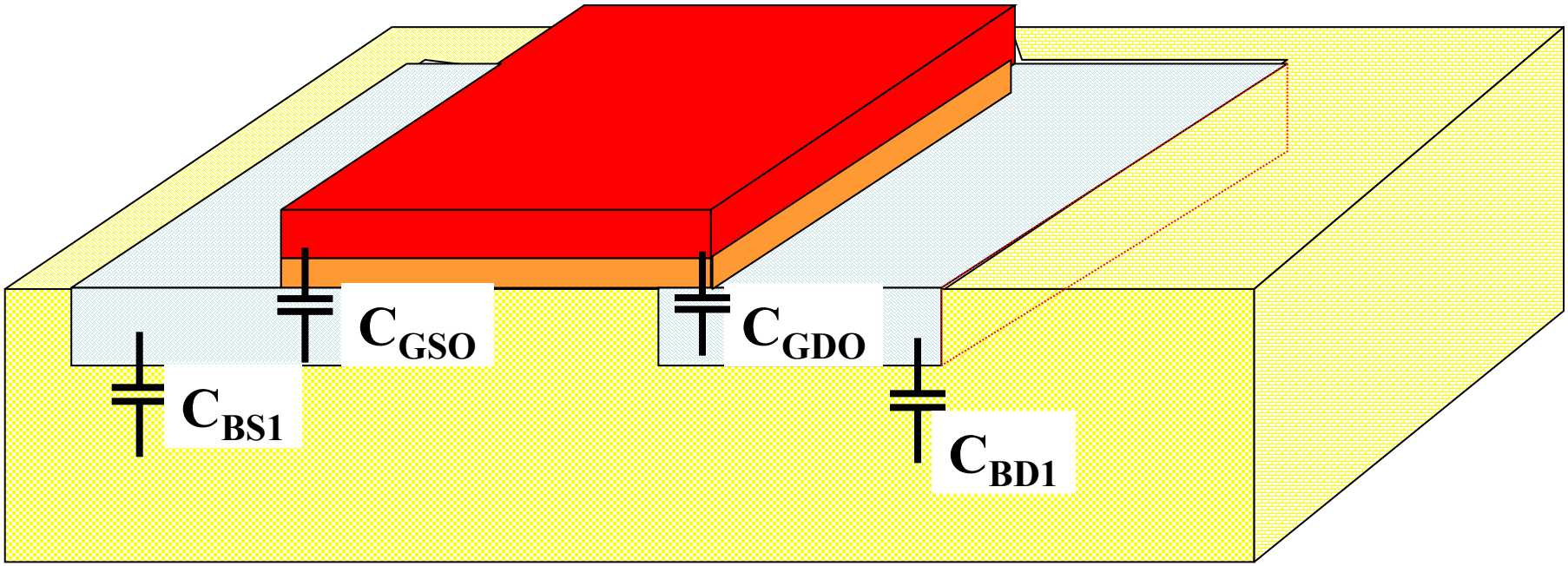
# Parasitic Capacitors in MOSFET

## Fixed Capacitors




Junction Capacitors: $C_{BS1}$, $C_{BD1}$

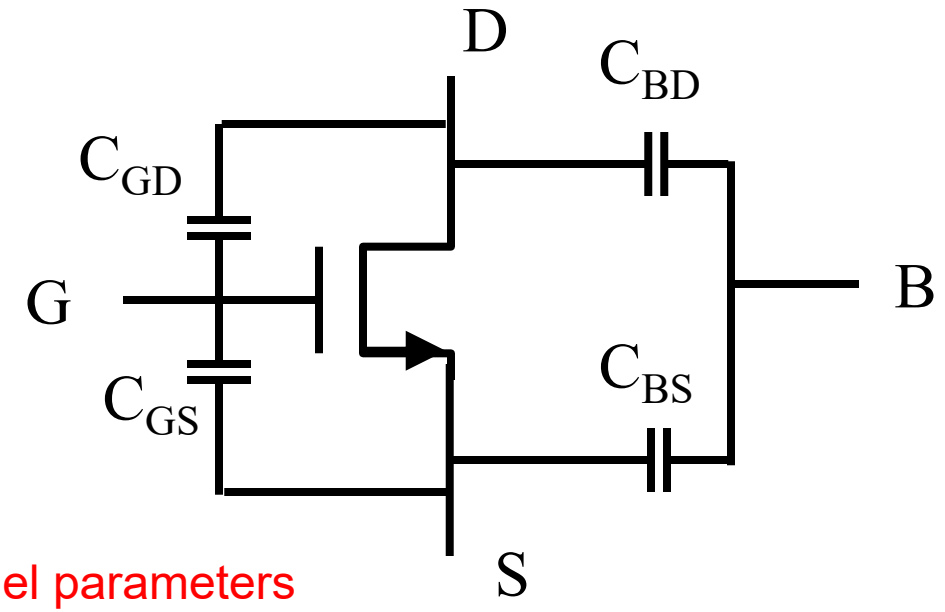
# Parasitic Capacitors in MOSFET
## Fixed Capacitors




Overlap Capacitors: $C_{GDO}$, $C_{GSO}$

Junction Capacitors: $C_{BS1}$, $C_{BD1}$

# Fixed Parasitic Capacitance Summary




$C_{BOT}$ and $C_{SW}$ are model parameters

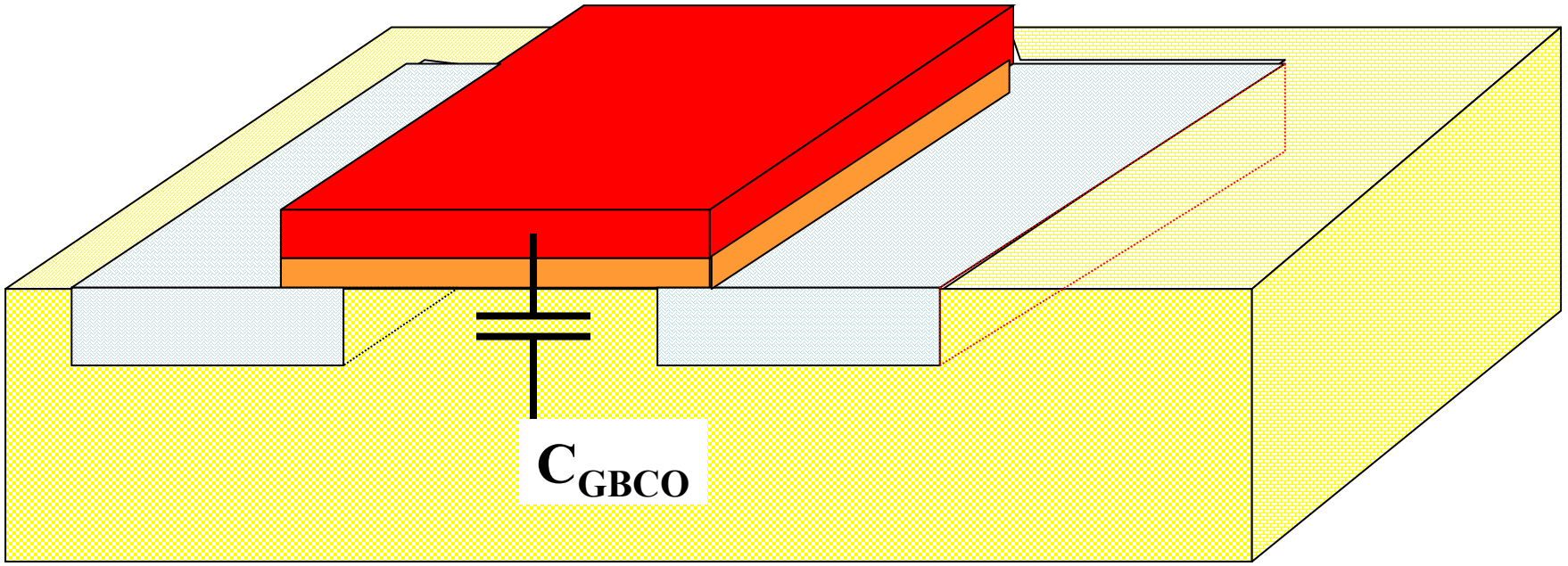
| | Cutoff | Ohmic | Saturation |
|---|---|---|---|
| **$C_{GS}$** | $CoxWL_D$ | $CoxWL_D$ | $CoxWL_D$ |
| **$C_{GD}$** | $CoxWL_D$ | $CoxWL_D$ | $CoxWL_D$ |
| **$C_{BG}$** | | | |
| **$C_{BS}$** | $C_{BS1} = C_{BOT}A_S+C_{SW}P_S$ | $C_{BS1} = C_{BOT}A_S+C_{SW}P_S$ | $C_{BS1} = C_{BOT}A_S+C_{SW}P_S$ |
| **$C_{BD}$** | $C_{BD1} = C_{BOT}A_D+C_{SW}P_D$ | $C_{BD1} = C_{BOT}A_D+C_{SW}P_D$ | $C_{BD1} = C_{BOT}A_D+C_{SW}P_D$ |

# Parasitic Capacitors in MOSFET

## Operation Region Dependent

# Parasitic Capacitors in MOSFET

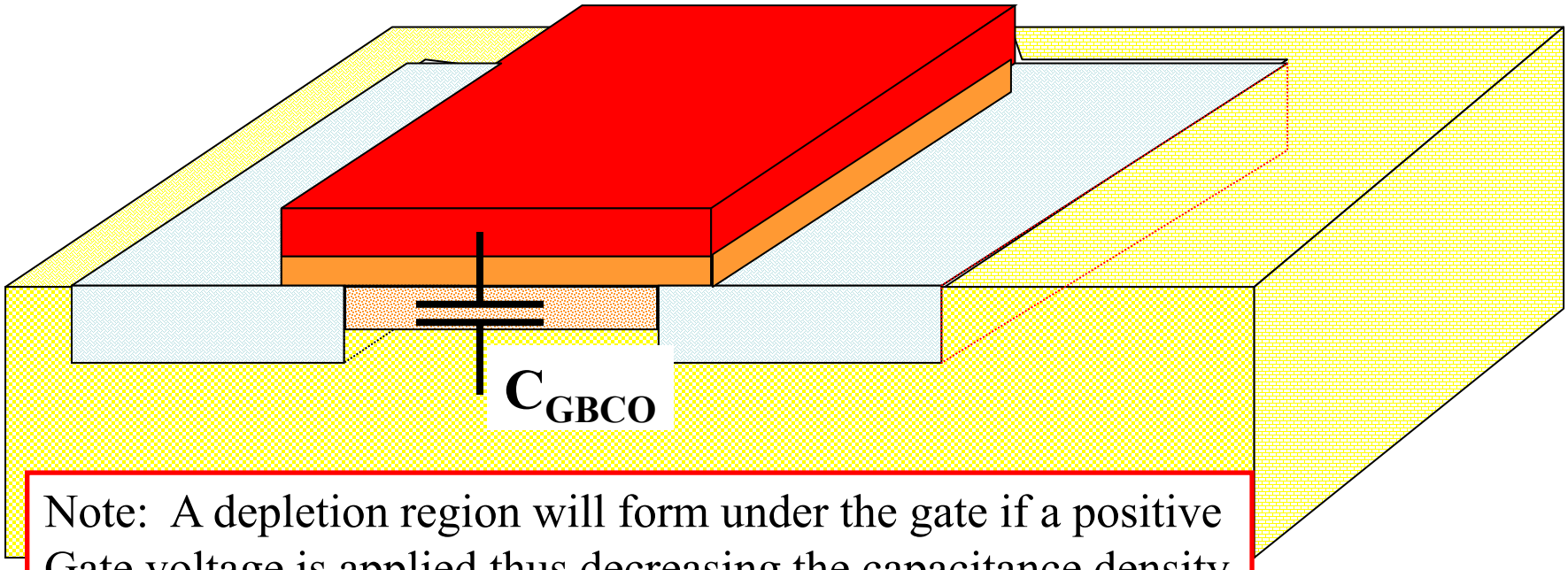
## Operation Region Dependent -- Cutoff




**Cutoff Capacitor: $C_{GBCO}$**

# Parasitic Capacitors in MOSFET

## Operation Region Dependent -- Cutoff

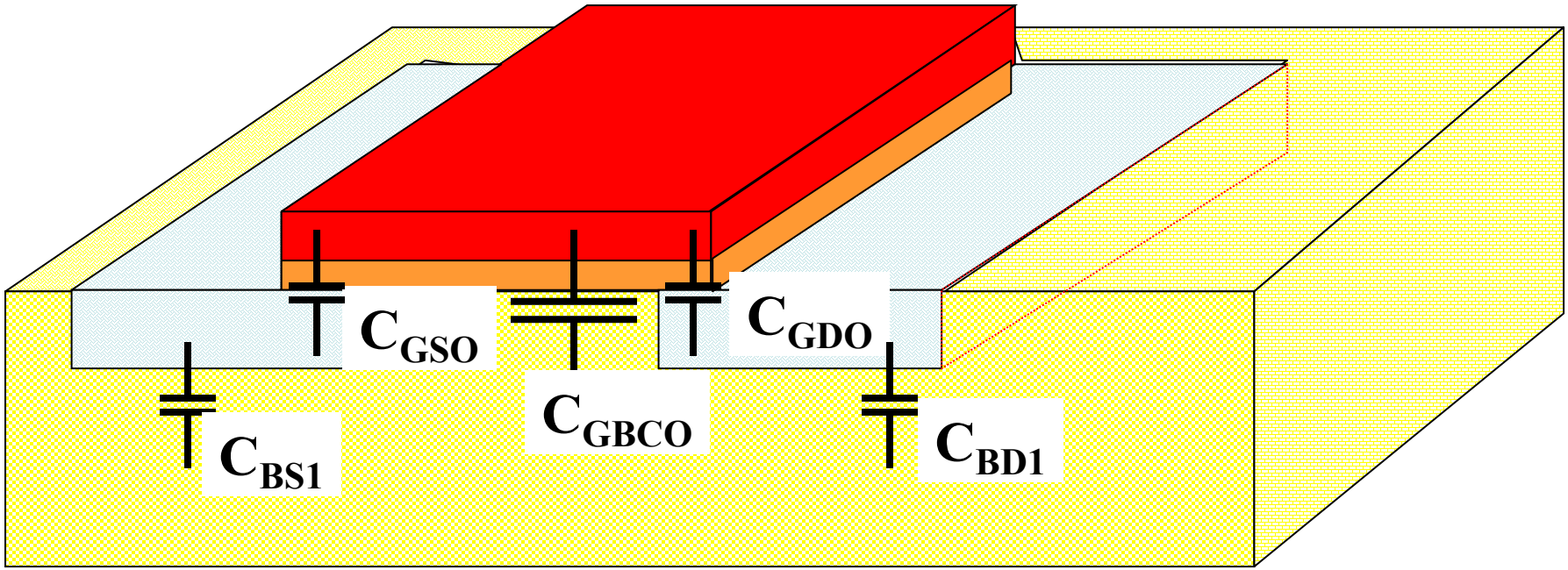



Note: A depletion region will form under the gate if a positive Gate voltage is applied thus decreasing the capacitance density

**Cutoff Capacitor: $C_{GBCO}$**

# Parasitic Capacitors in MOSFET

## Operation Region Dependent and Fixed -- Cutoff

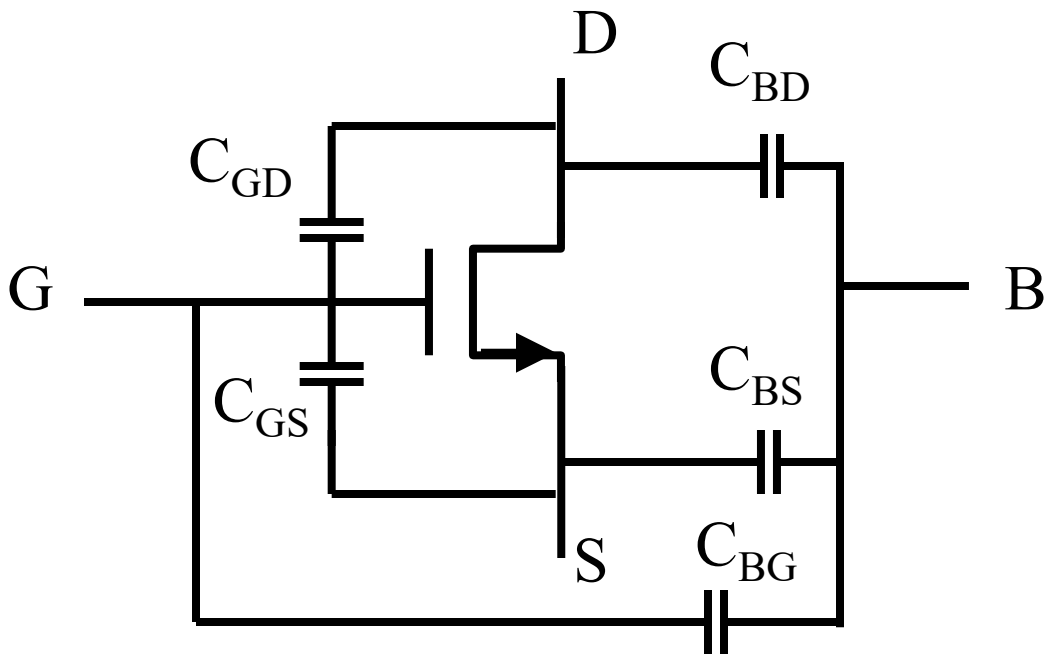



Overlap Capacitors: $C_{GDO}$, $C_{GSO}$

Junction Capacitors: $C_{BS1}$, $C_{BD1}$

**Cutoff Capacitor: $C_{GBCO}$**

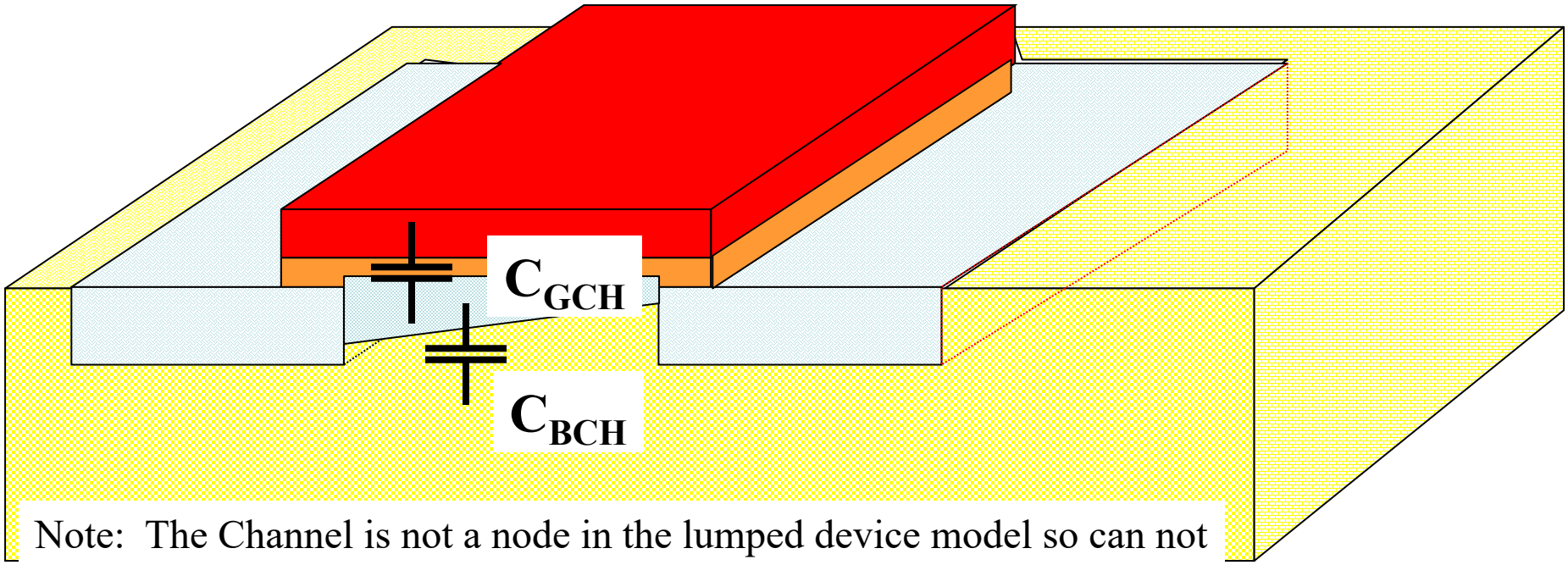
# Parasitic Capacitance Summary




| | Cutoff | Ohmic | Saturation |
|---|---|---|---|
| $C_{GS}$ | $CoxWL_D$ | | |
| $C_{GD}$ | $CoxWL_D$ | | |
| $C_{BG}$ | CoxWL (or less) | | |
| $C_{BS}$ | $C_{BOT}A_S+C_{SW}P_S$ | | |
| $C_{BD}$ | $C_{BOT}A_D+C_{SW}P_D$ | | |

# Parasitic Capacitors in MOSFET

## Operation Region Dependent -- Ohmic



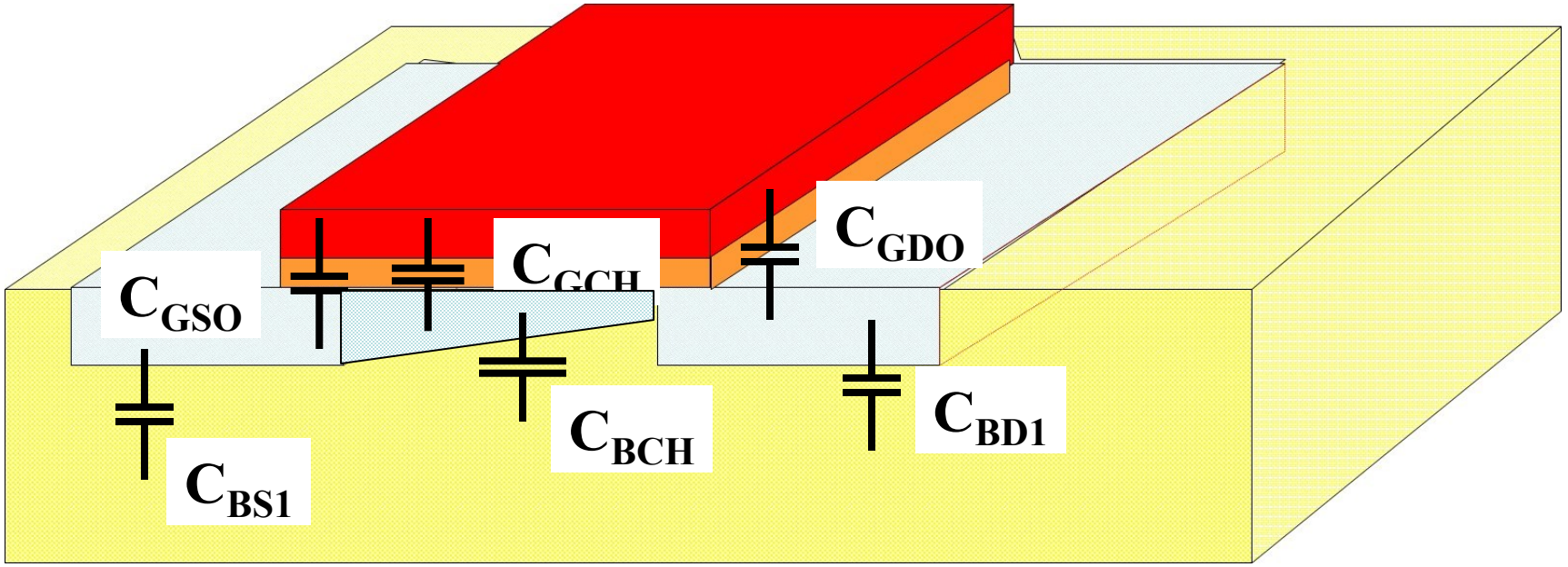

Note: The Channel is not a node in the lumped device model so can not directly include this distributed capacitance in existing models

Note: The distributed channel capacitance is usually lumped and split evenly between the source and drain nodes

**Ohmic Capacitor: $C_{GCH}$, $C_{BCH}$**

# Parasitic Capacitors in MOSFET

## Operation Region Dependent and Fixed -- Ohmic

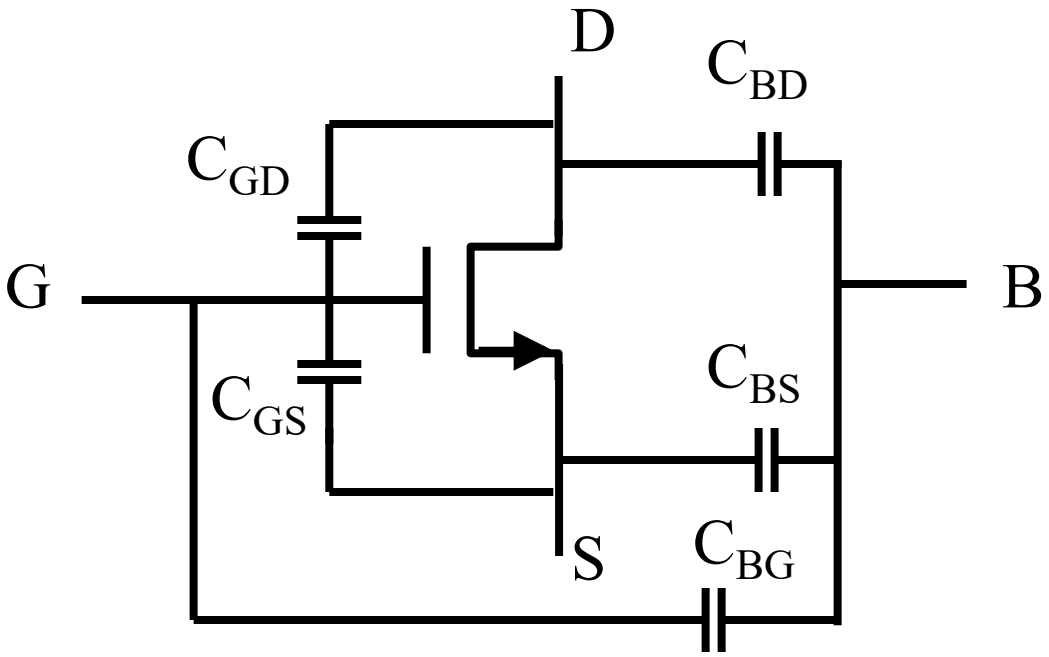



Overlap Capacitors: $C_{GDO}$, $C_{GSO}$

Junction Capacitors: $C_{BS1}$, $C_{BD1}$

**Ohmic Capacitor: $C_{GCH}$ , $C_{BCH}$**

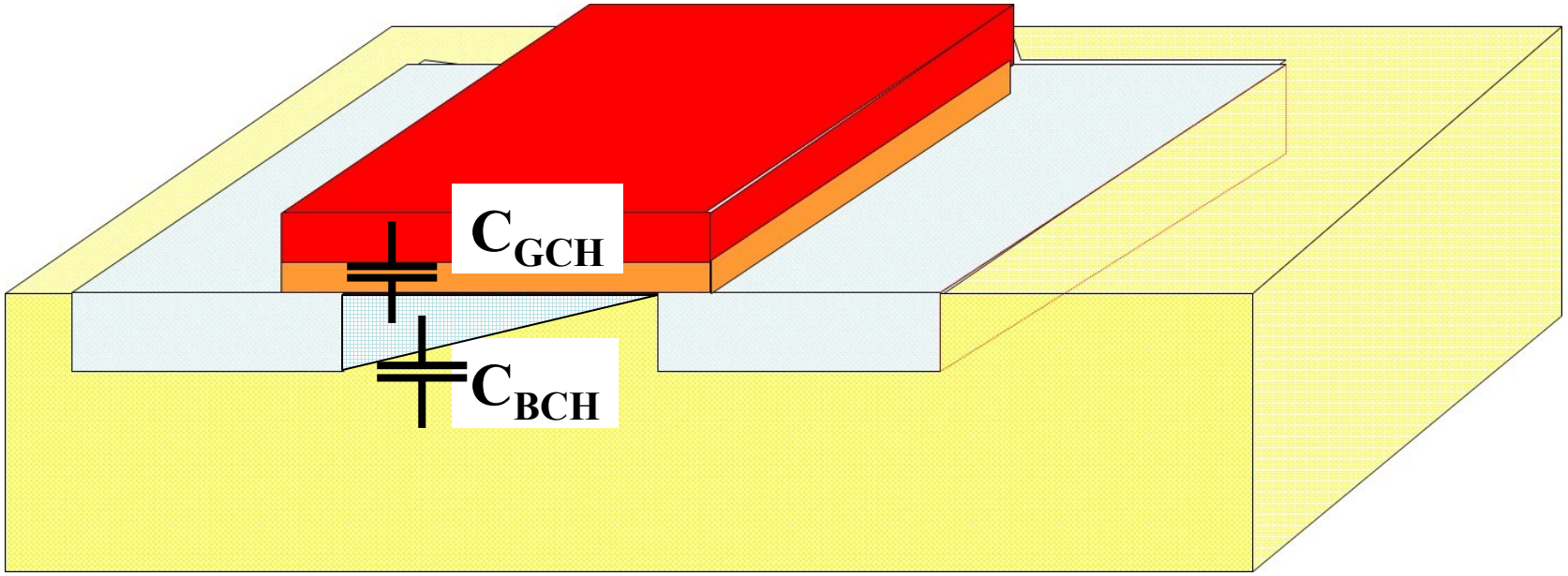
# Parasitic Capacitance Summary




| | Cutoff | Ohmic | Saturation |
|---|---|---|---|
| $C_{GS}$ | $CoxWL_D$ | $CoxWL_D$ | |
| $C_{GD}$ | $CoxWL_D$ | $CoxWL_D$ | |
| $C_{BG}$ | CoxWL (or less) | | |
| $C_{BS}$ | $C_{BOT}A_S+C_{SW}P_S$ | $C_{BS1} = C_{BOT}A_S+C_{SW}P_S$ | |
| $C_{BD}$ | $C_{BOT}A_D+C_{SW}P_D$ | $C_{BD1} = C_{BOT}A_D+C_{SW}P_D$ | |

# Parasitic Capacitors in MOSFET

## Operation Region Dependent -- Saturation

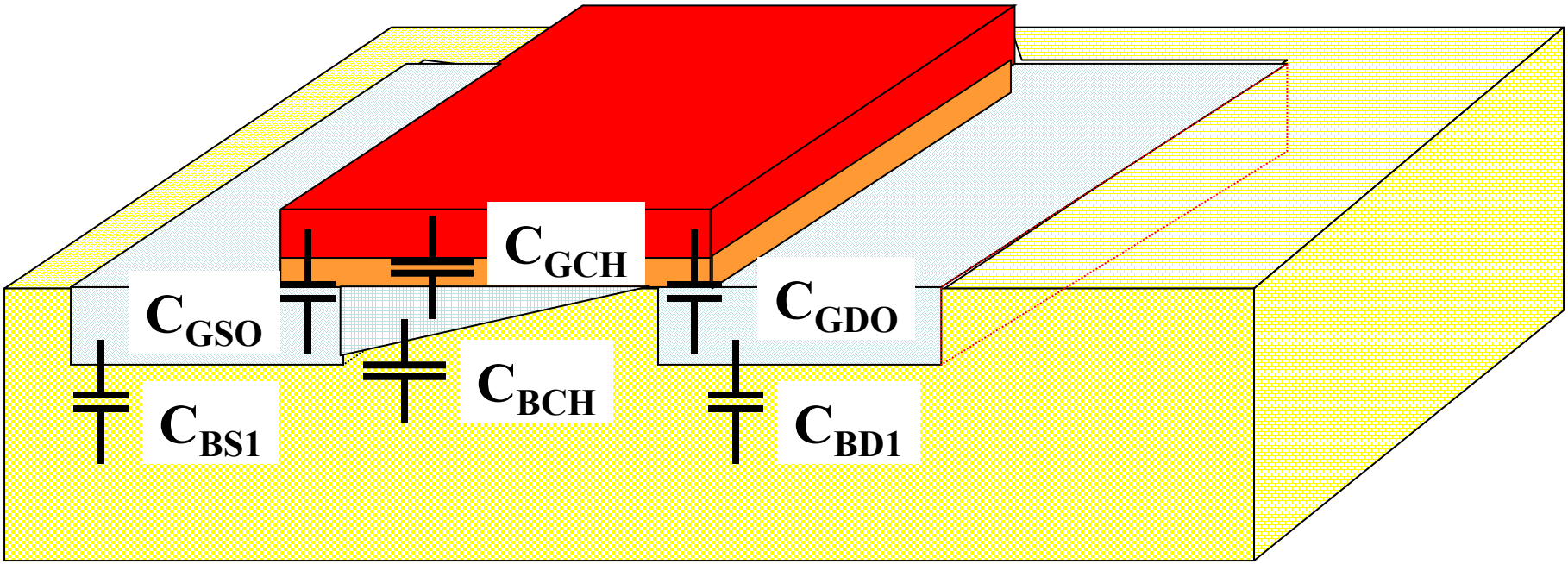



Note: Since the channel is an extension of the source when in saturation, the distributed capacitors to the channel are generally lumped to the source node

**Saturation Capacitors: $C_{GCH}$, $C_{BCH}$**

# Parasitic Capacitors in MOSFET

## Operation Region Dependent and Fixed --Saturation

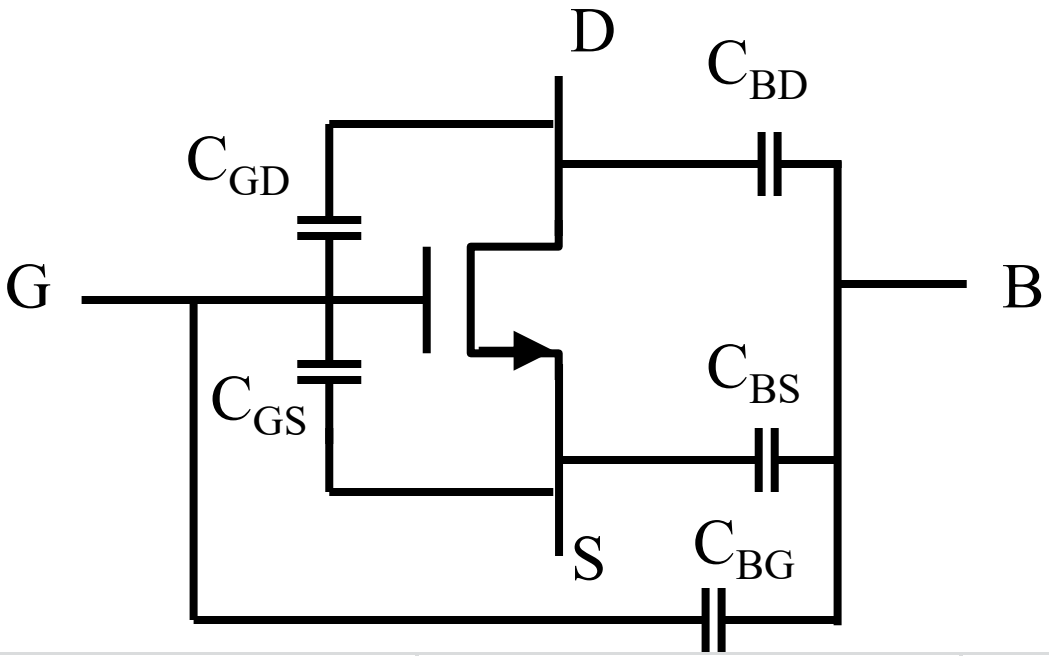



Overlap Capacitors: $C_{GDO}$, $C_{GSO}$

Junction Capacitors: $C_{BS1}$, $C_{BD1}$

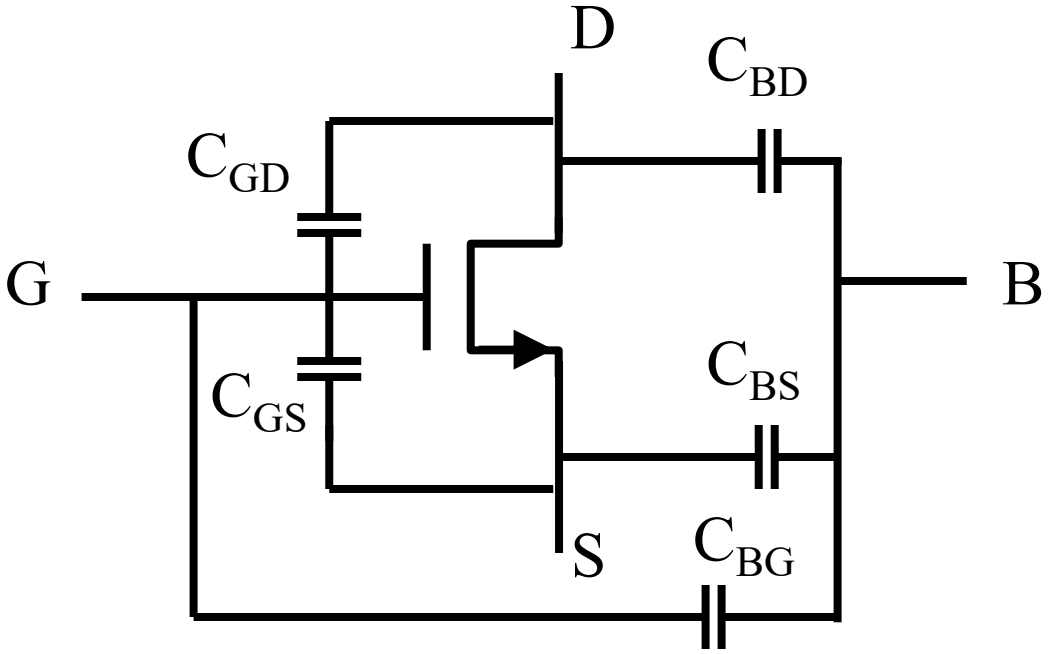
**Saturation Capacitors: $C_{GCH}$ , $C_{BCH}$**

# Parasitic Capacitance Summary




| | Cutoff | Ohmic | Saturation |
|---|---|---|---|
| $C_{GS}$ | $CoxWL_D$ | $CoxWL_D + 0.5C_{OX}WL$ | $CoxWL_D+(2/3)C_{OX}WL$ |
| $C_{GD}$ | $CoxWL_D$ | $CoxWL_D + 0.5C_{OX}WL$ | $CoxWL_D$ |
| $C_{BG}$ | CoxWL (or less) | 0 | 0 |
| $C_{BS}$ | $C_{BOT}A_S+C_{SW}P_S$ | $C_{BOT}A_S+C_{SW}P_S+0.5WLC_{BOTCH}$ | $C_{BOT}A_S+C_{SW}P_S +(2/3)WLC_{BOTCH}$ |
| $C_{BD}$ | $C_{BOT}A_D+C_{SW}P_D$ | $C_{BOT}A_D+C_{SW}P_D+0.5WLC_{BOTCH}$ | $C_{BOT}A_D+C_{SW}P_D$ |

# Parasitic Capacitance Summary

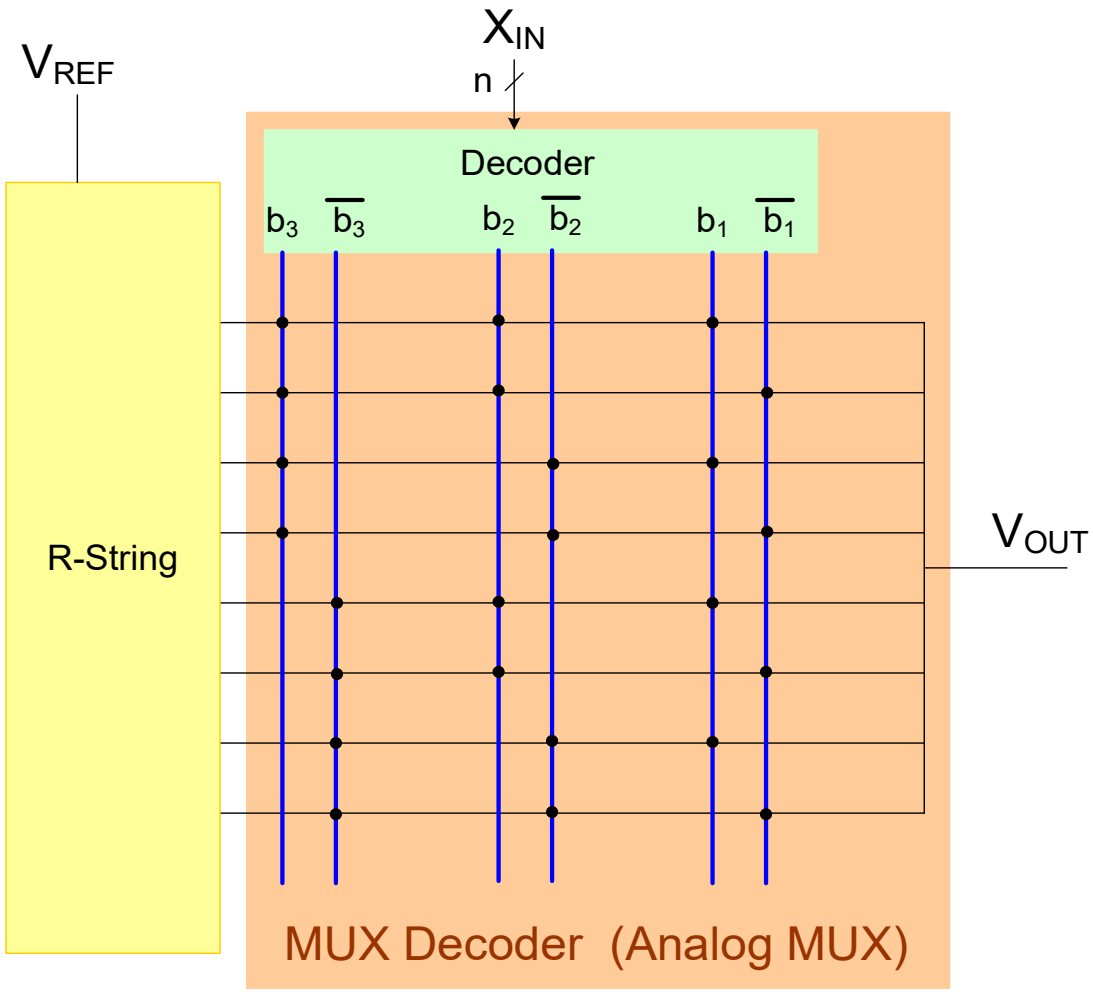



| | Cutoff | Ohmic | Saturation |
|---|---|---|---|
| **$C_{GS}$** | $CoxWL_D$ | $CoxWL_D + 0.5C_{OX}WL$ | $CoxWL_D+(2/3)C_{OX}WL$ |
| **$C_{GD}$** | $CoxWL_D$ | $CoxWL_D + 0.5C_{OX}WL$ | $CoxWL_D$ |
| **$C_{BG}$** | CoxWL (or less) | 0 | 0 |
| **$C_{BS}$** | $C_{BOT}A_S+C_{SW}P_S$ | $C_{BOT}A_S+C_{SW}P_S+0.5WLC_{BOTCH}$ | $C_{BOT}A_S+C_{SW}P_S +(2/3)WLC_{BOTCH}$ |
| **$C_{BD}$** | $C_{BOT}A_D+C_{SW}P_D$ | $C_{BOT}A_D+C_{SW}P_D+0.5WLC_{BOTCH}$ | $C_{BOT}A_D+C_{SW}P_D$ |

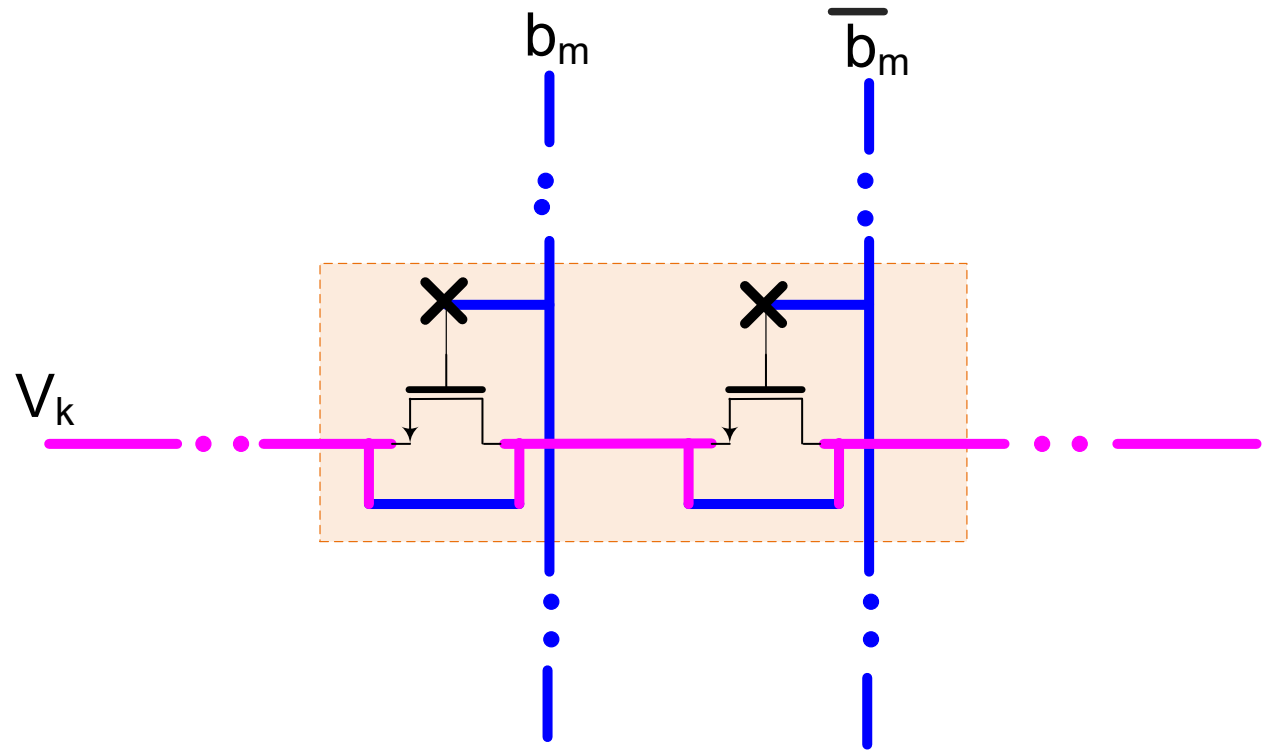
# R-String DAC

Tree-Decoder Layout/Architecture

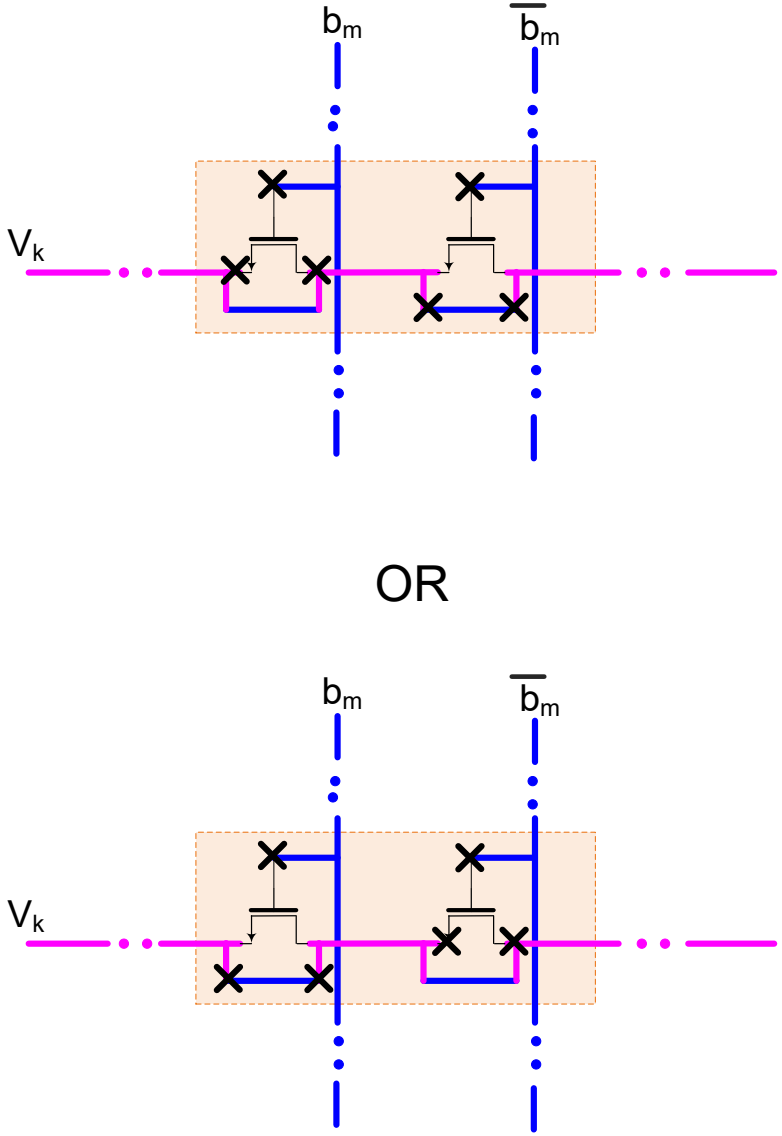
Each intersection is a reserved site for a switch



# Uncontacted Row-Column Structure

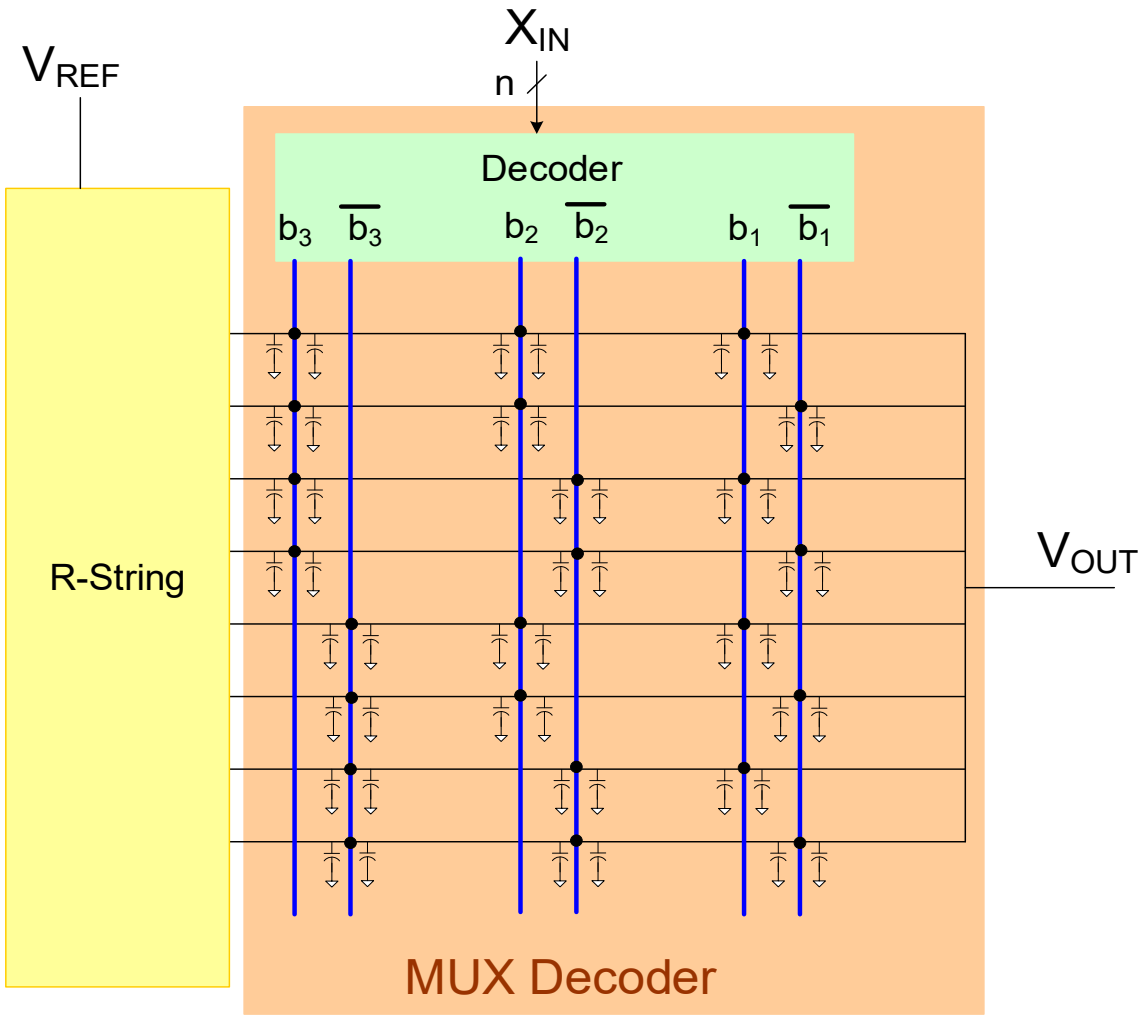


# Row-Column Structure with Contacts Added




Programmed entirely with the contact mask

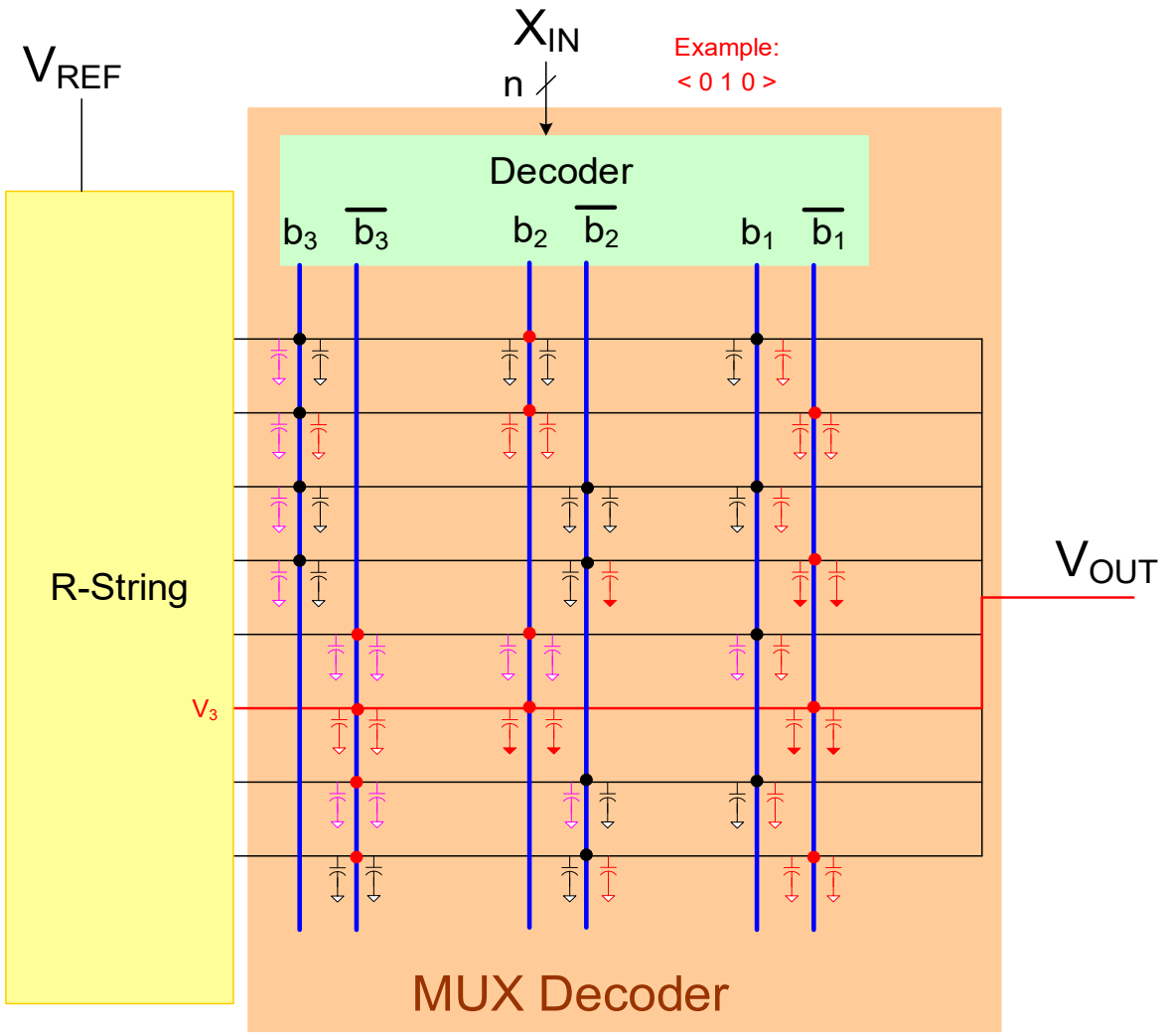
# R-String DAC




## Parasitic Capacitances in MUX Decoder

(for convenience have not shown non-contacted transistor effects)

# R-String DAC



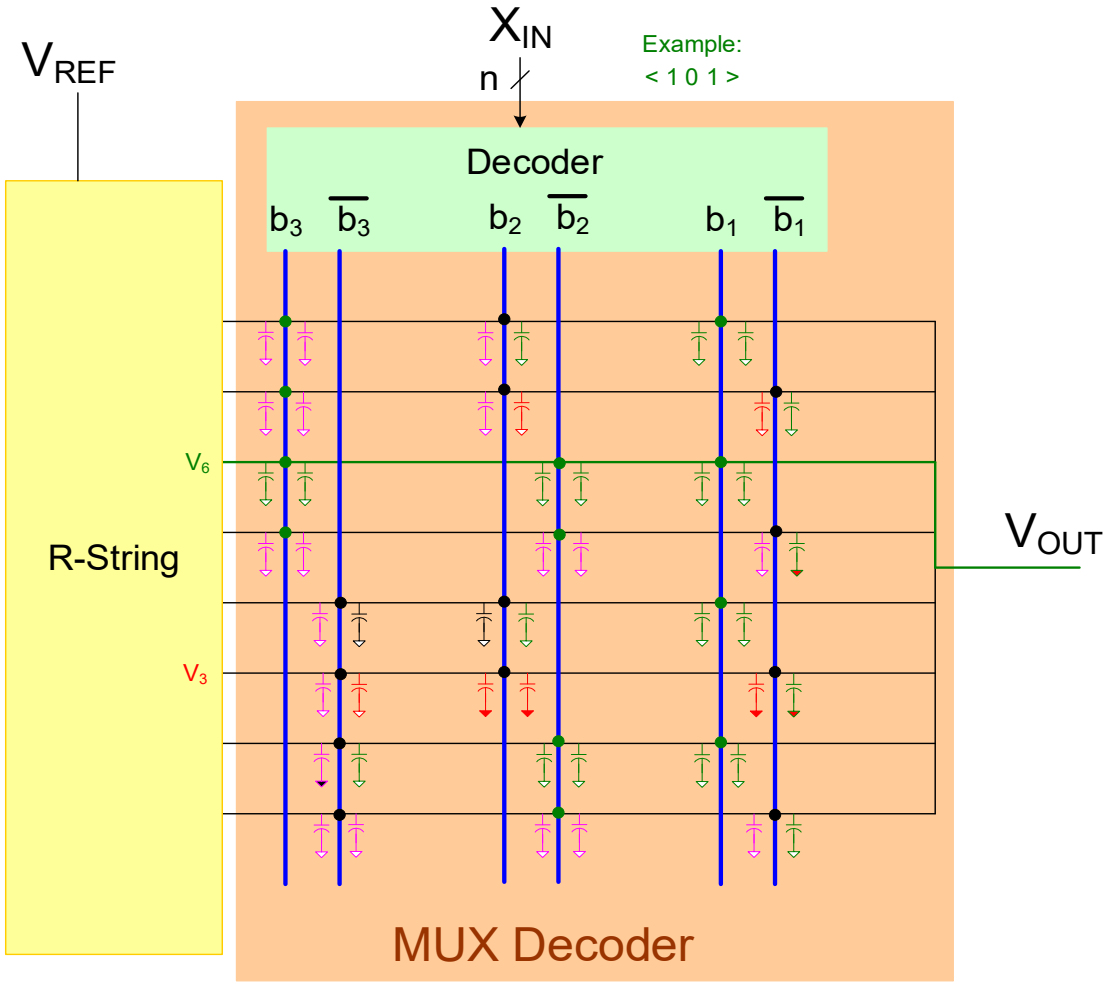

## Previous-Code Dependent Settling

Assume all C's initially with 0V

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

# R-String DAC

Transition from <010> to <101>




## Previous-Code Dependent Settling

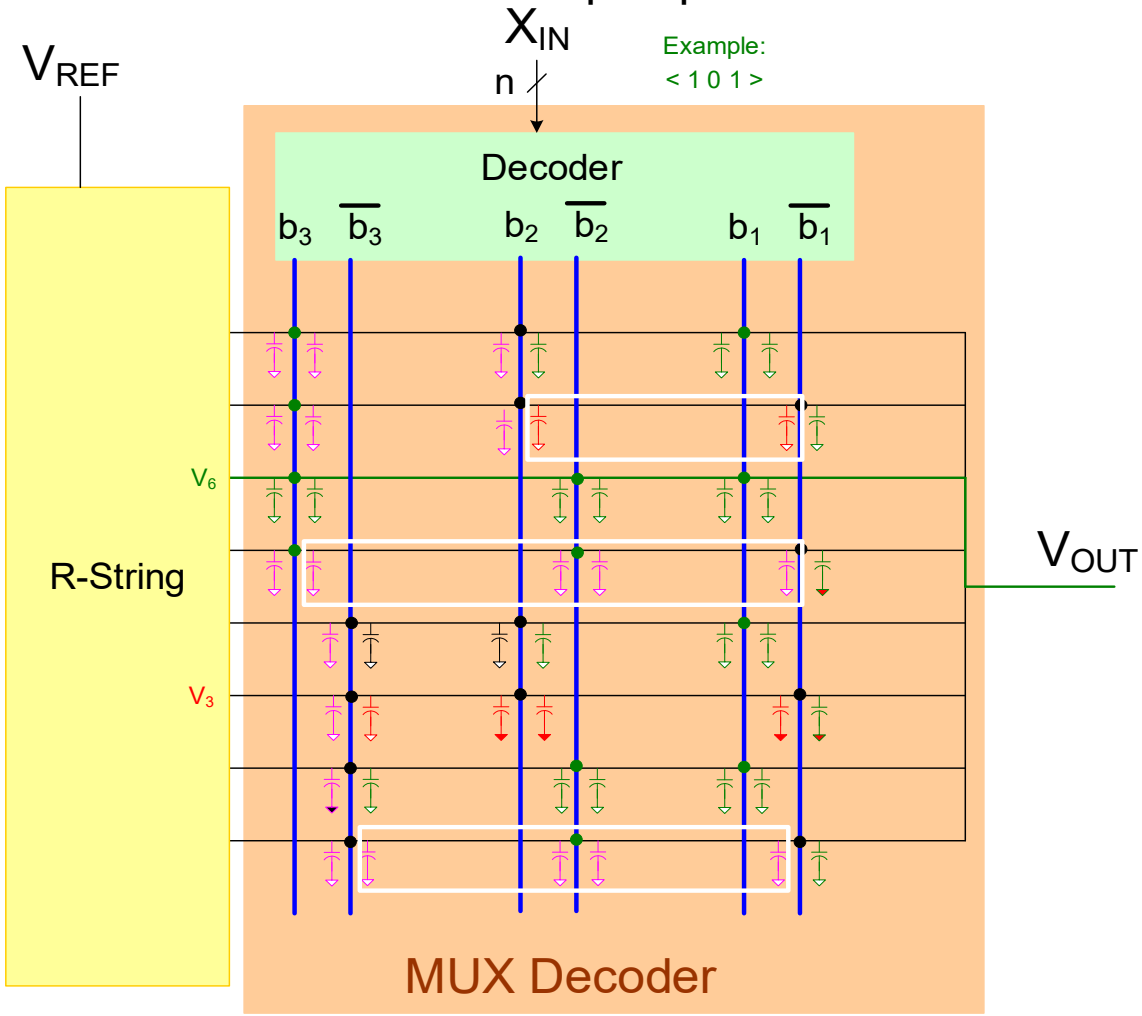
Assume all C's initially with 0V

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

# R-String DAC

Transition from <010> to <101>

White boxes show capacitors dependent upon previous code <010>

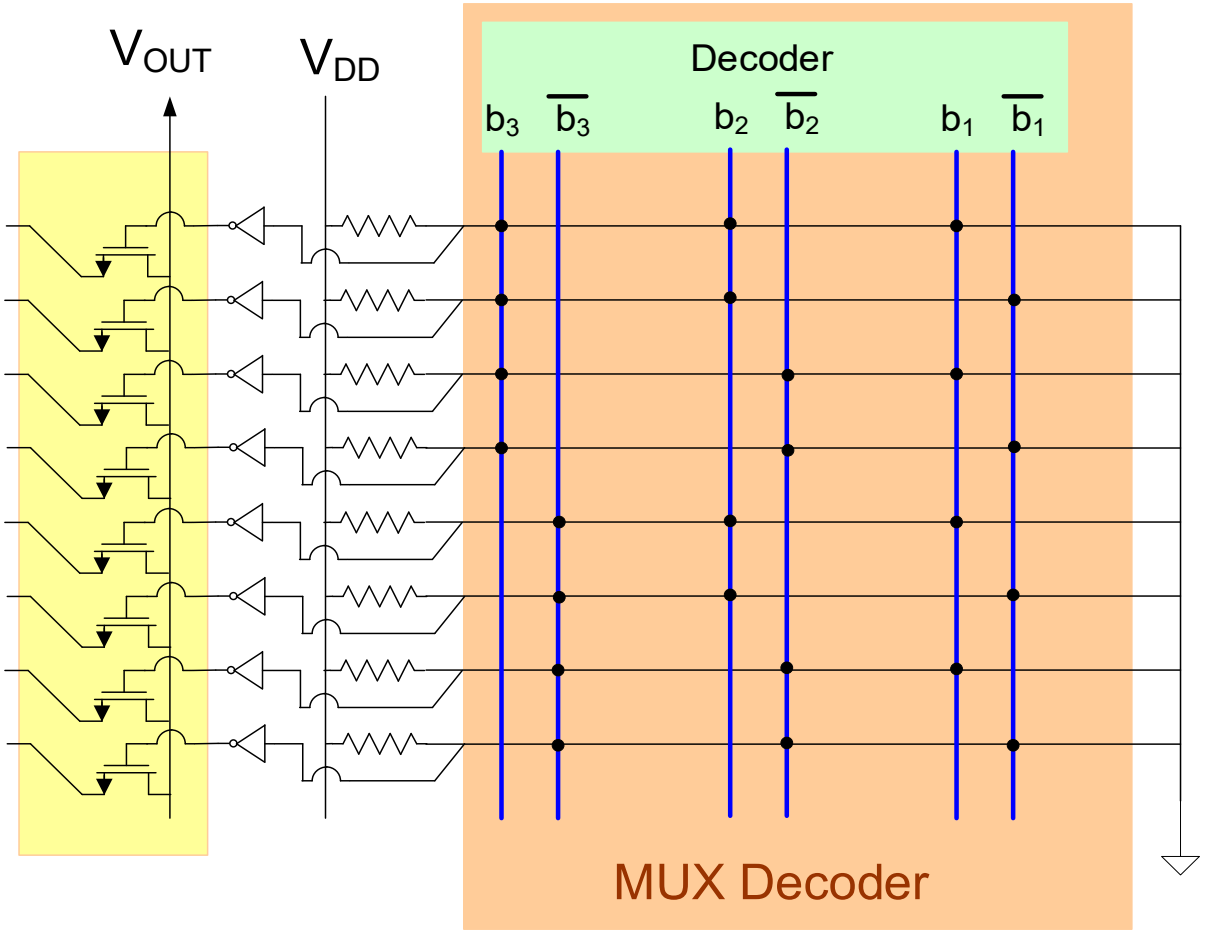



## Previous-Code Dependent Settling

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



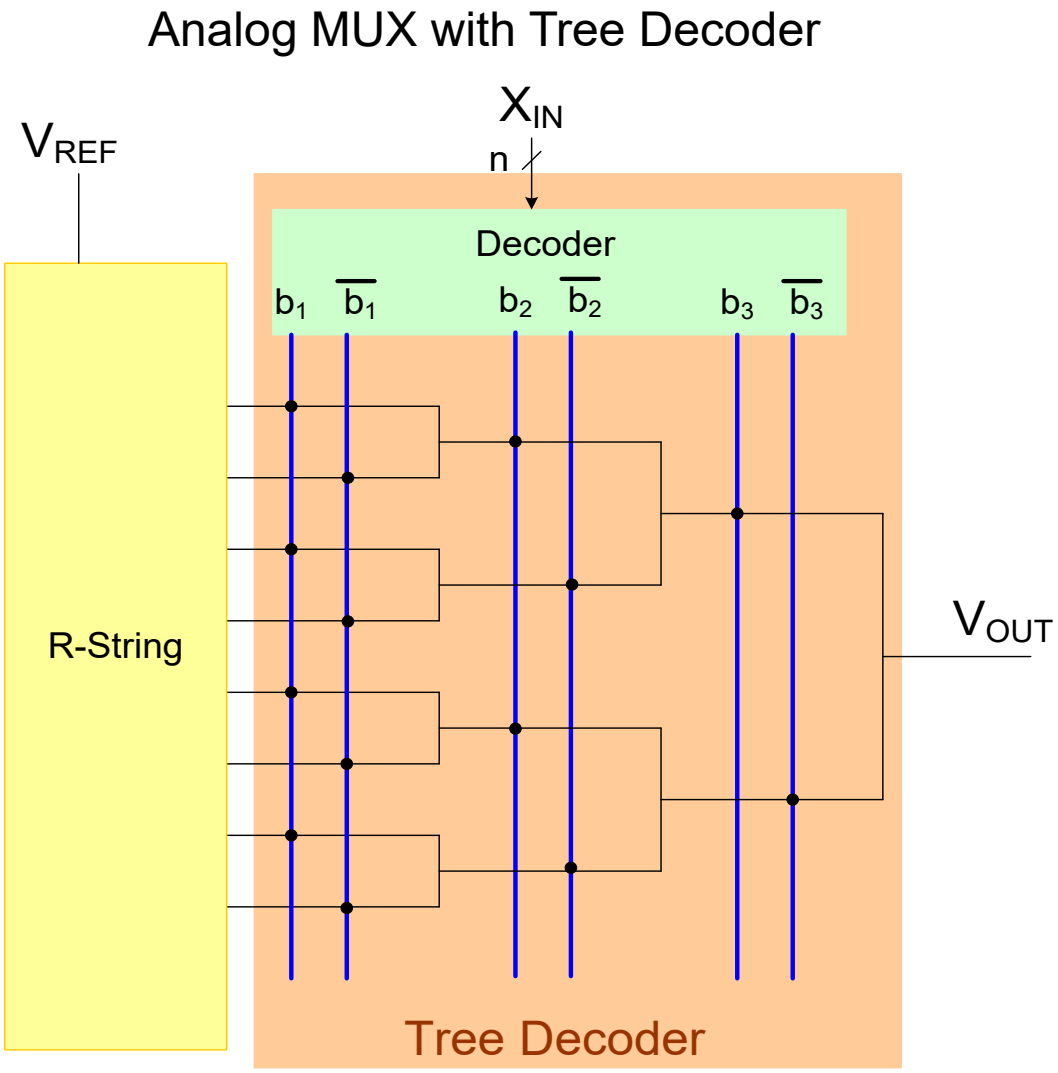

## Tree-Decoder in Digital Domain

Single transistor used at each marked intersection to form PTL -AND gates

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

No because only one will be conducting for any DAC output

# R-String DAC


Analog MUX with Tree Decoder
$X_{IN}$
n
$V_{REF}$
Decoder
$b_1$ $\overline{b_1}$ $b_2$ $\overline{b_2}$ $b_3$ $\overline{b_3}$
R-String
$V_{OUT}$
Tree Decoder

TEXAS INSTRUMENTS

DAC8560

SLAS464C – DECEMBER 2006 – REVISED JANUARY 2018

# DAC8560 16-Bit, Ultra-Low Glitch, Voltage Output Digital-to-Analog Converter With 2.5-V, 2-ppm/°C Internal Reference

## 1 Features

- Relative Accuracy: 4 LSB
- Glitch Energy: 0.15 nV-s
- *Micro*Power Operation: 510 μA at 2.7 V
- Internal Reference:
  - 2.5-V Reference Voltage (Enabled by Default)
  - 0.02% Initial Accuracy
  - 2-ppm/°C Temperature Drift (Typical)
  - 5-ppm/°C Temperature Drift (Maximum)
  - 20-mA Sink/Source Capability
- Power-On Reset to Zero
- Power Supply: 2.7 V to 5.5 V
- 16-Bit Monotonic Over Temperature Range
- Settling Time: 10 μs to ±0.003% FSR
- Low-Power Serial Interface With Schmitt-Triggered Inputs
- On-Chip Output Buffer Amplifier With Rail-to-Rail Operation
- Power-Down Capability
- Drop-In Compatible With DAC8531/01 and DAC8550 /51
- Temperature Range: –40°C to +105°C
- Available in a Tiny 8-Pin VSSOP Package

## 3 Description

The DAC8560 is a low-power, voltage output, 16-bit digital-to-analog converter (DAC). The DAC8560 includes a 2.5-V, 2-ppm/°C internal reference (enabled by default), giving a full-scale output voltage range of 0 V to 2.5 V. The internal reference has an initial accuracy of 0.02% and can source up to 20 mA at the $V_{REF}$ pin. The device is monotonic, provides very good linearity, and minimizes undesired code-to-code transient voltages (glitch). The DAC8560 uses a versatile 3-wire serial interface that operates at clock rates up to 30 MHz. It is compatible with standard SPI, QSPI, Microwire, and digital-signal-processor (DSP) interfaces.

The DAC8560 incorporates a power-on-reset (POR) circuit that ensures the DAC output powers up at zero scale and remains there until a valid code is written to the device. The DAC8560 contains a power-down feature, accessed over the serial interface, that reduces the current consumption of the device to 1.2 μA at 5 V.

The low-power consumption, internal reference, and small footprint make this device ideal for portable, battery-operated equipment. The power consumption is 2.6 mW at 5 V, reducing to 6 μW in power-down mode.

The DAC8560 is available in an 8-pin VSSOP package.

### 7.3.1 Digital-to-Analog Converter (DAC)

The DAC8560 architecture consists of a string DAC followed by an output buffer amplifier. Figure 63 shows a block diagram of the DAC architecture.

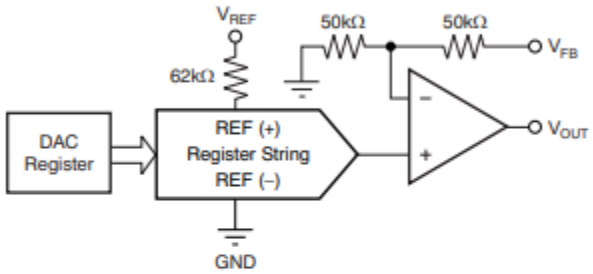



**Figure 63. DAC8560 Architecture**

The input coding to the DAC8560 is straight binary, so the ideal output voltage is given by:

$$V_{OUT} = \frac{D_{IN}}{65536} \times V_{REF} \quad (1)$$

where $D_{IN}$ = decimal equivalent of the binary code that is loaded to the DAC register; it can range from 0 to 65535.

### 7.3.2 Resistor String

The resistor string section is shown in Figure 64. It is simply a string of resistors, each of value R. The code loaded into the DAC register determines at which node on the string the voltage is tapped off to be fed into the output amplifier by closing one of the switches connecting the string to the amplifier. It is monotonic because it is a string of resistors.

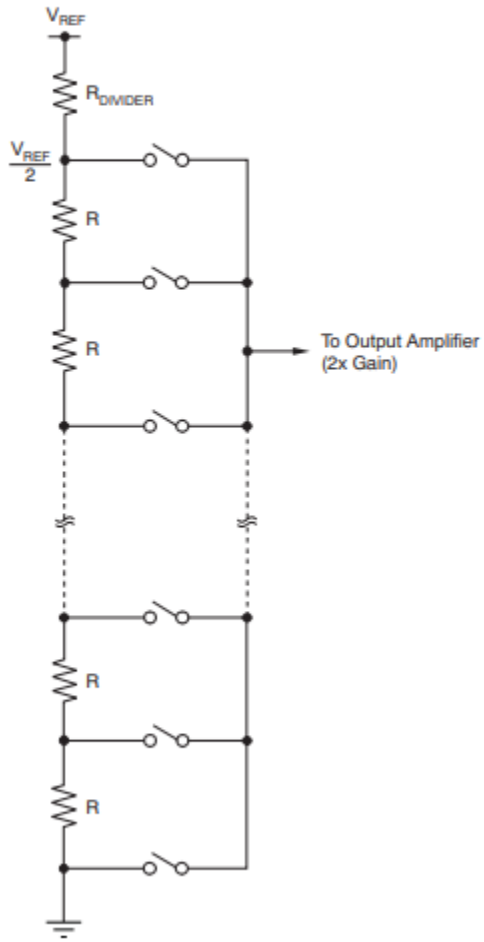



**Figure 64. Resistor String**

## 2 Applications

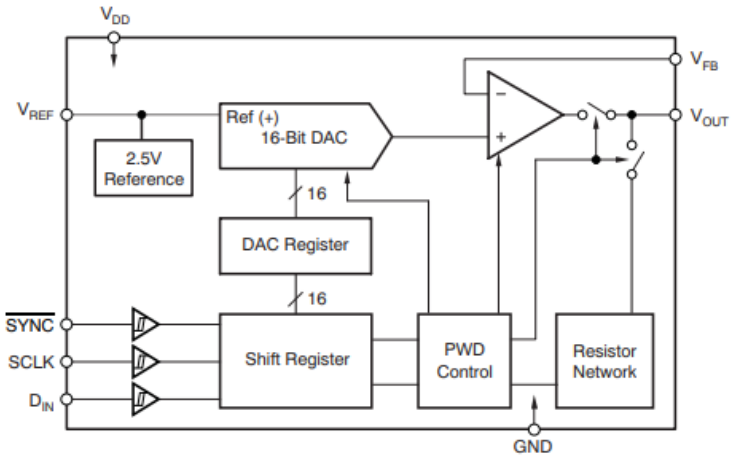
- Process Control
- Data Acquisition Systems
- Closed-Loop Servo-Control
- PC Peripherals
- Portable Instrumentation

**Device Information(1)**

| PART NUMBER | PACKAGE | BODY SIZE (NOM) |
|---|---|---|
| DAC8560 | VSSOP (8) | 3.00 mm × 3.00 mm |

(1) For all available packages, see the orderable addendum at the end of the data sheet.

**Functional Block Diagram**



## 6.5 Electrical Characteristics

$V_{DD}$ = 2.7 V to 5.5 V, –40°C to +105°C range (unless otherwise noted)

| PARAMETER | TEST CONDITIONS | | MIN | TYP | MAX | UNIT |
|---|---|---|---|---|---|---|
| **STATIC PERFORMANCE**(1) | | | | | | |
| Resolution | | | 16 | | | Bits |
| Relative accuracy | Measured by line passing through codes 485 and 64714 | DAC8560A, DAC8560C | | ±4 | ±12 | LSB |
| | | DAC8560B, DAC8560D | | ±4 | ±8 | LSB |
| Differential nonlinearity | 16-bit Monotonic | | | ±0.5 | ±1 | LSB |
| Zero-code error | Measured by line passing through codes 485 and 64714. | | | ±5 | ±12 | mV |
| Full-scale error | | | | ±0.2 | ±0.5 | % of FSR |
| Gain error | | | | ±0.05 | ±0.2 | % of FSR |
| Zero-code error drift | | | | ±4 | | µV/°C |
| Gain temperature coefficient | $V_{DD}$ = 5 V | | | ±1 | | ppm of FSR/°C |
| | $V_{DD}$ = 2.7 V | | | ±3 | | |
| PSRR | Power supply rejection ratio | Output unloaded | | 1 | | mV/V |
| **OUTPUT CHARACTERISTICS**(2) | | | | | | |
| Output voltage range | | | 0 | | $V_{REF}$ | V |
| Output voltage settling time | To ±0.003% FSR, 0200h to FD00h, $R_L$ = 2 kΩ, 0 pF < $C_L$ < 200 pF | | | 8 | 10 | µs |
| | $R_L$ = 2 kΩ, $C_L$ = 500 pF | | | 12 | | |
| Slew rate | | | | 1.8 | | V/µs |
| Capacitive load stability | $R_L$ = ∞ | | | 470 | | pF |
| | $R_L$ = 2 kΩ | | | 1000 | | |
| Code change glitch impulse | 1 LSB change around major carry | | | 0.15 | | nV-s |
| Digital feedthrough | SCLK toggling, $\overline{SYNC}$ high | | | 0.15 | | nV-s |
| DC output impedance | At mid-code input | | | 1 | | Ω |
| Short-circuit current | $V_{DD}$ = 5 V | | | 50 | | mA |
| | $V_{DD}$ = 3 V | | | 20 | | |
| Power-up time | Coming out of power-down mode $V_{DD}$ = 5 V | | | 2.5 | | µs |
| | Coming out of power-down mode $V_{DD}$ = 3 V | | | 5 | | |
| **AC PERFORMANCE**(2) | | | | | | |
| SNR | $T_A$ = 25°C, BW = 20 kHz, $V_{DD}$ = 5 V, $f_{OUT}$ = 1 kHz, 1st 19 harmonics removed for SNR calculation | | | 88 | | dB |
| THD | | | | –77 | | dB |
| SFDR | | | | 79 | | dB |
| SINAD | | | | 77 | | dB |
| DAC output noise density | $T_A$ = 25°C, at mid-code input, $f_{OUT}$ = 1 kHz | | | 170 | | nV/√Hz |
| DAC output noise | $T_A$ = 25°C, at mid-code input, 0.1 Hz to 10 Hz | | | 50 | | $\mu V_{PP}$ |

(1) Linearity calculated using a reduced code range of 485 to 64714; output unloaded.
(2) Ensured by design and characterization, not production tested.

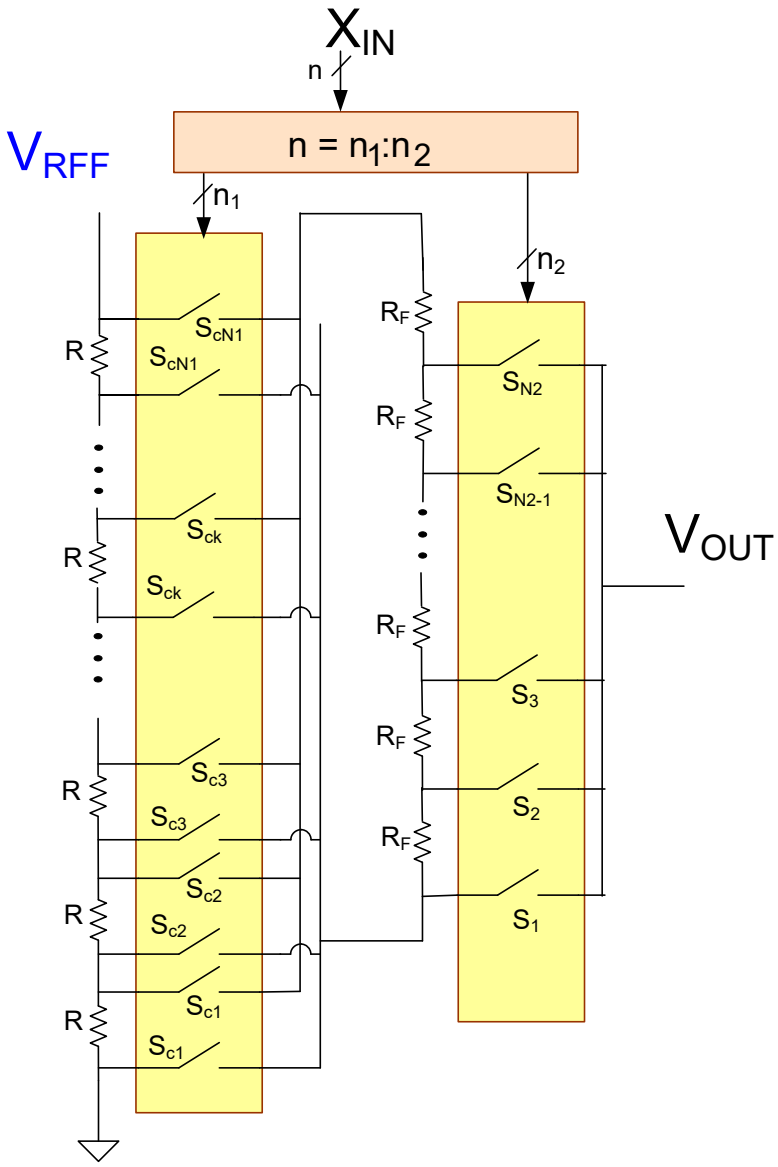
# The DAC 8560

What is the INL performance of this DAC?

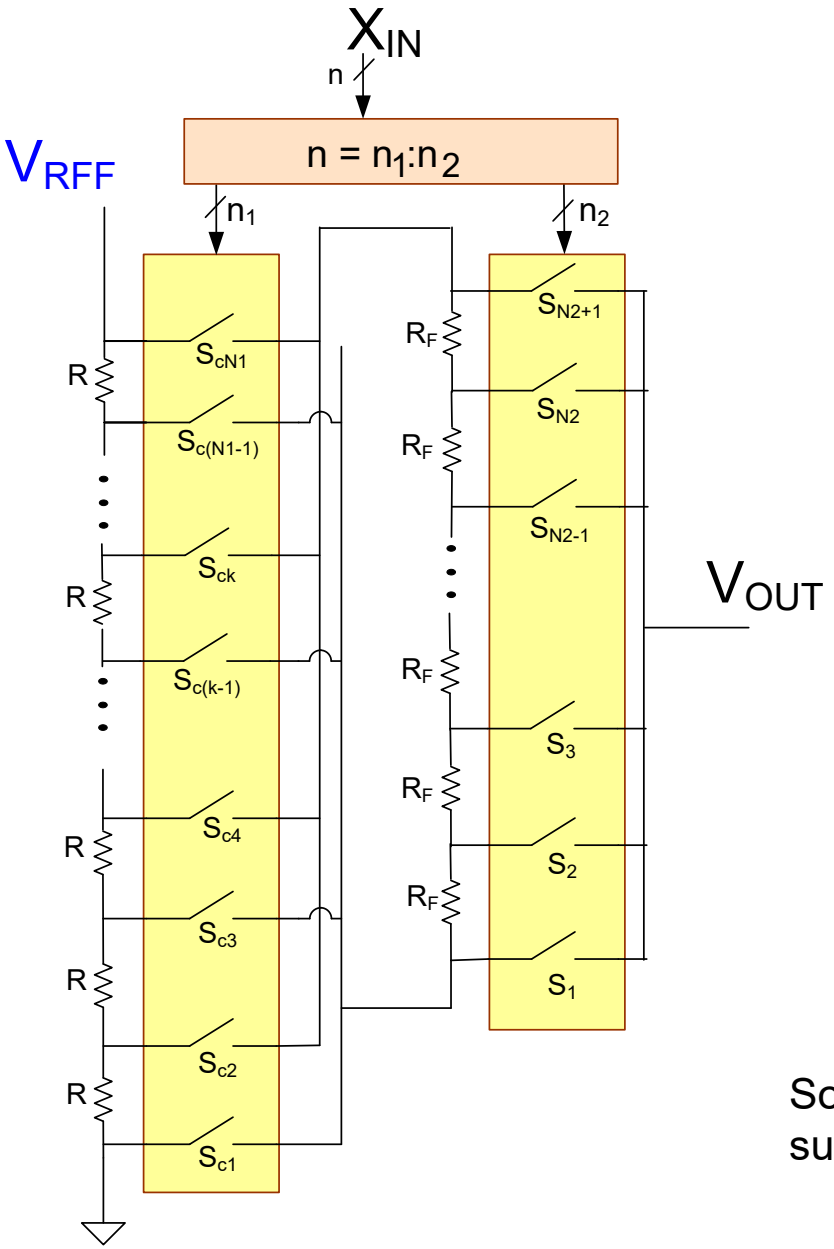
ENOB?

What is the spectral performance?

# R-String DAC

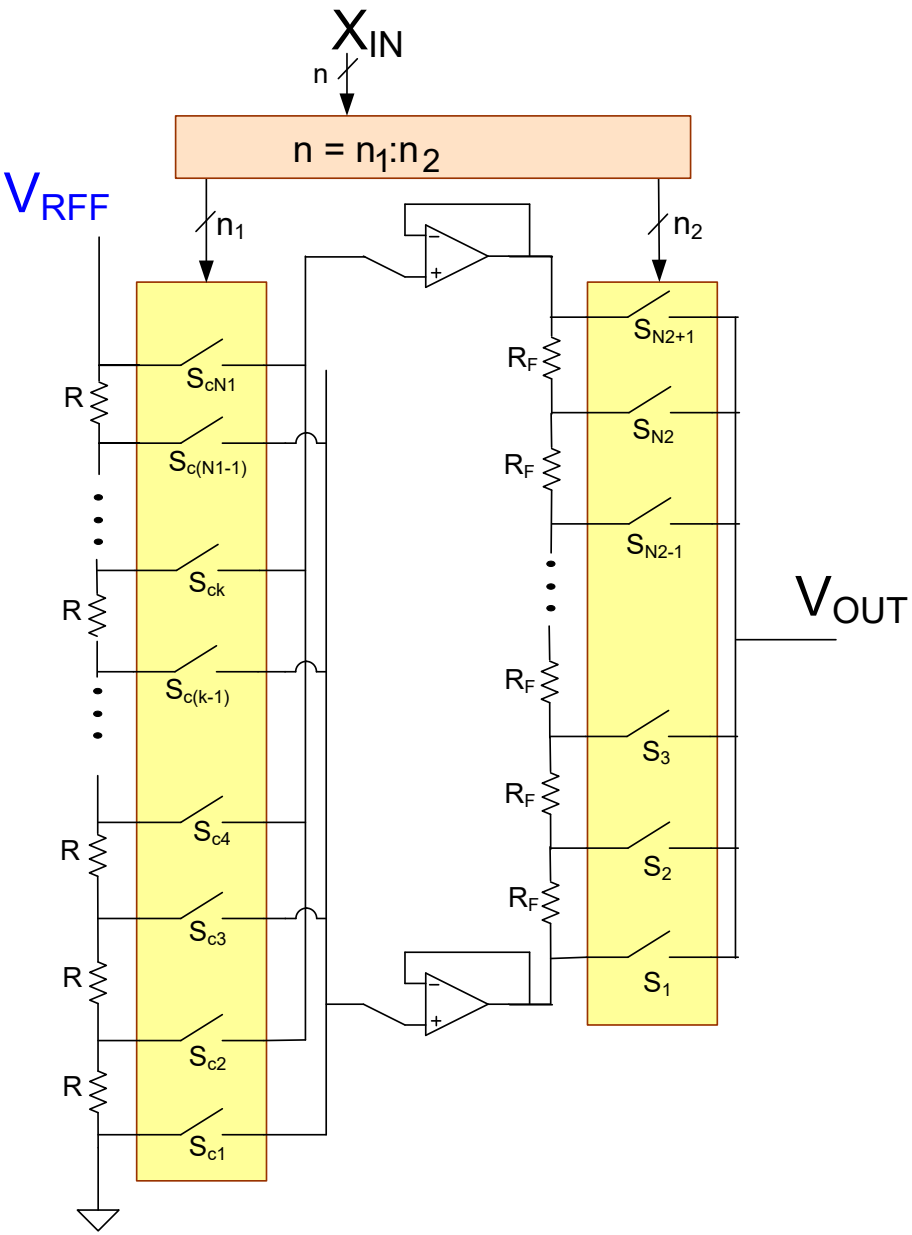

X_IN
n
n = n1:n2
V_RFF
n1
n2
R
R
R
R
R
S_cN1
S_cN1
S_ck
S_ck
S_c3
S_c3
S_c2
S_c2
S_c1
S_c1
R_F
R_F
R_F
R_F
R_F
S_N2
S_N2-1
S_3
S_2
S_1
V_OUT

# R-String DAC

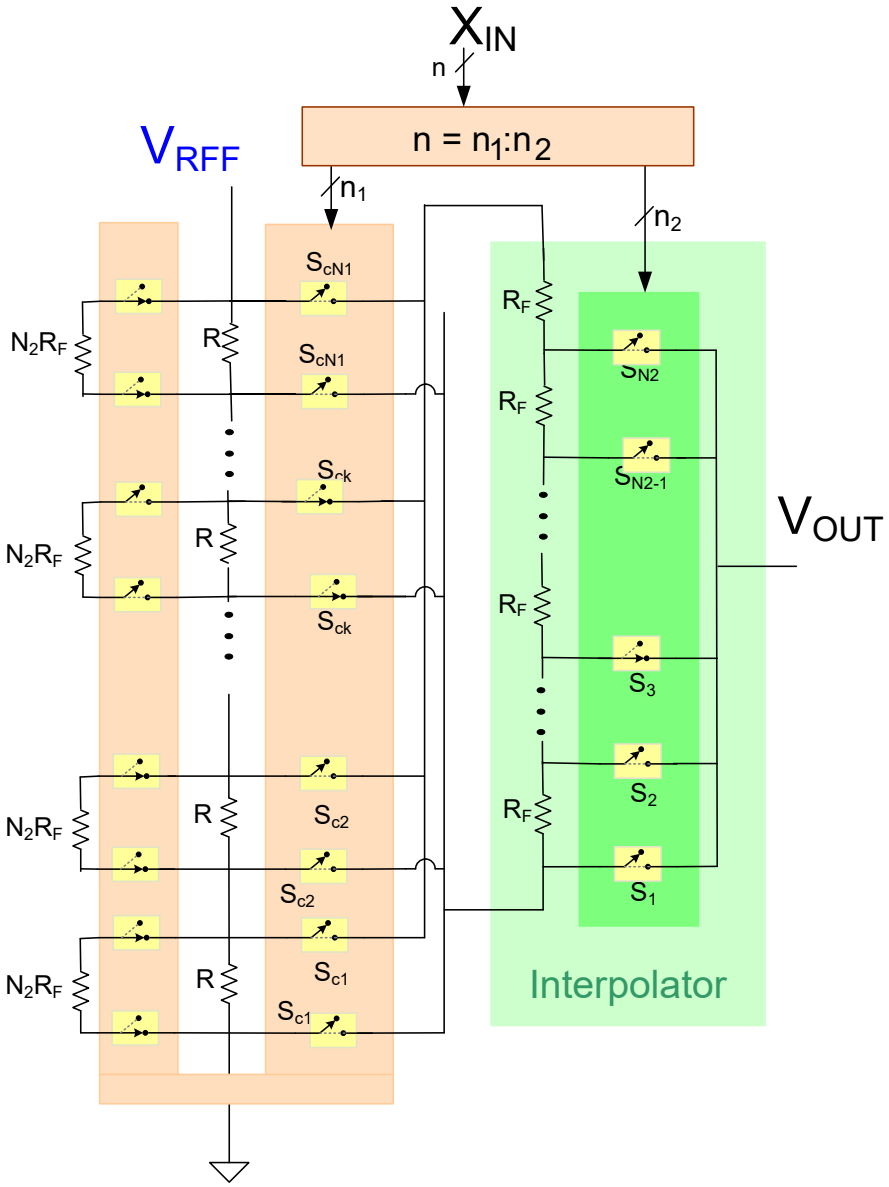



Sometimes termed sub-divider, sub-range or dual-string DAC

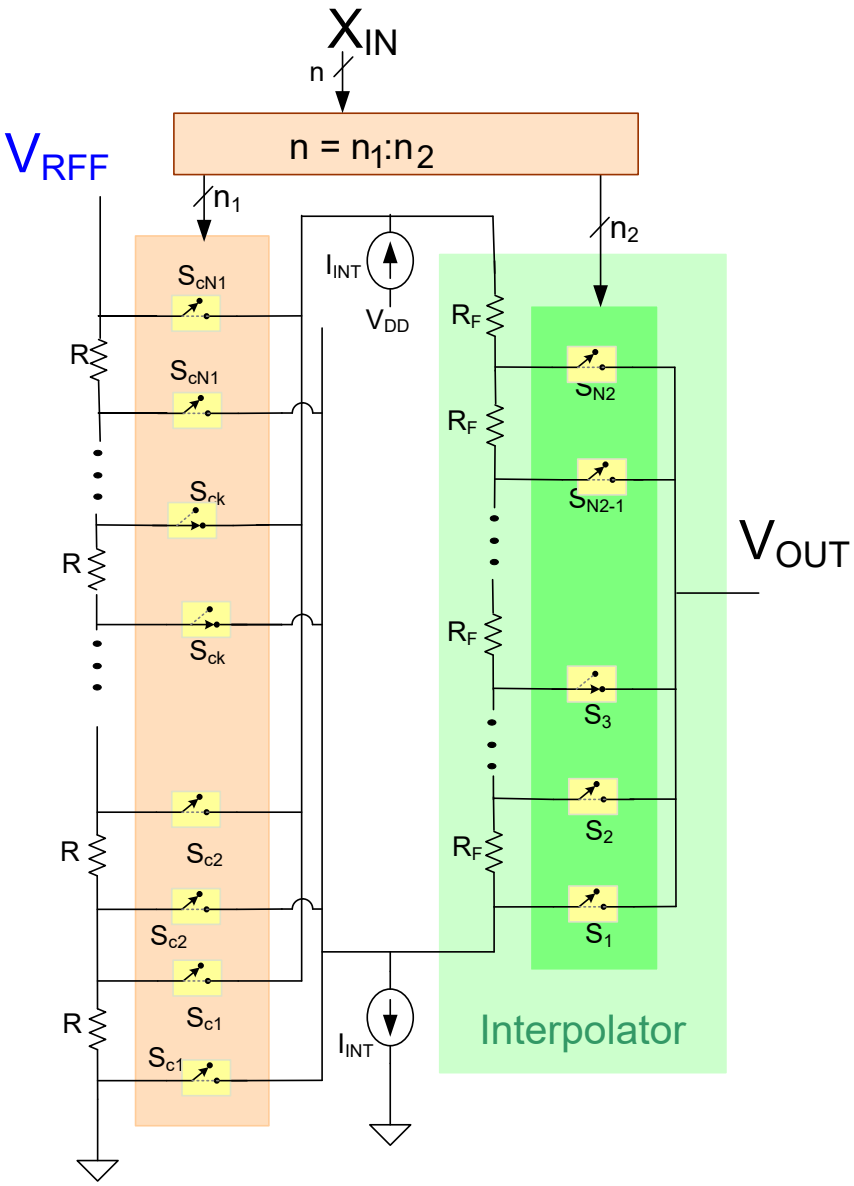
# R-String DAC


X_IN
n
n = n1:n2
V_RFF
n1
n2
S_N2+1
S_cN1
R_F
R
S_N2
S_c(N1-1)
S_N2-1
S_ck
V_OUT
S_c(k-1)
S_3
S_c4
S_2
S_c3
S_1
S_c2
S_c1

# R-String DAC


X_IN
n
n = n1:n2
V_RFF
n1
n2
S_cN1
S_cN1
N2R_F
R
S_ck
S_ck
N2R_F
R
N2R_F
R
S_c2
S_c2
N2R_F
R
S_c1
S_c1
R_F
R_F
R_F
R_F
S_N2
S_N2-1
S_3
S_2
S_1
V_OUT
Interpolator

# R-String DAC


X_IN
n
n = n1:n2
V_RFF
n1
n2
I_INT
V_DD
S_cN1
S_cN1
R
S_ck
R
S_ck
R
S_c2
S_c2
R
S_c1
S_c1
R_F
R_F
R_F
R_F
S_N2
S_N2-1
S_3
S_2
S_1
V_OUT
I_INT
Interpolator

Stay Safe and Stay Healthy !

End of Lecture 32