

# CMOS

CIRCUIT DESIGN,  
LAYOUT, AND  
SIMULATION

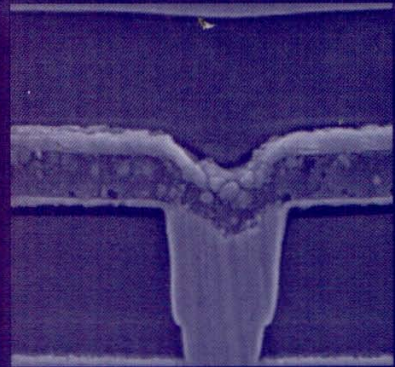
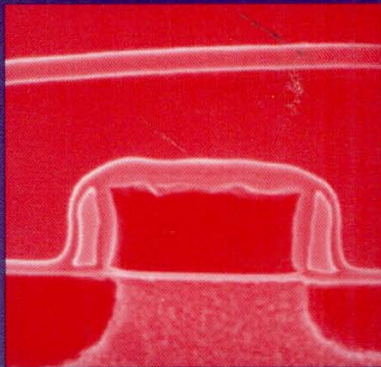
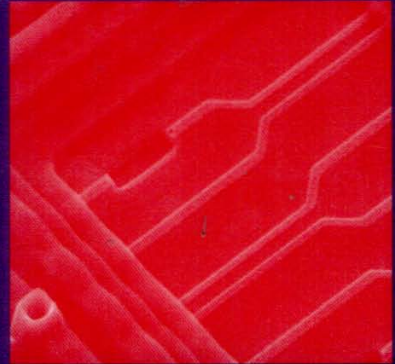
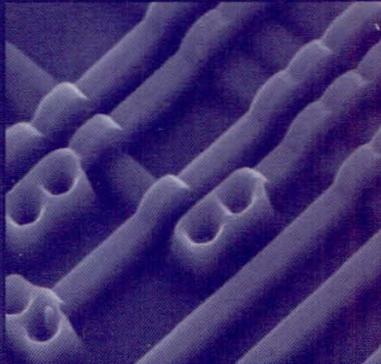
Baker



TK  
7871.99  
.M44B335  
1998  
C.1

# CMOS

CIRCUIT DESIGN, LAYOUT, AND SIMULATION



R. Jacob Baker ■ Harry W. Li ■ David E. Boyce

IEEE Press Series on Microelectronic Systems  
Stuart K. Tewksbury, *Series Editor*

The LASI Layout Software, as well as other  
layout software, can be downloaded at  
[http://www.mrc.uidaho.edu/vlsi/cad\\_free.htm](http://www.mrc.uidaho.edu/vlsi/cad_free.htm)

## Process Information for Hand-Calculations

| Process                                   | CN20 NMOS                       | CN20 PMOS                       | CMOS14 NMOS                              | CMOS14 PMOS                               |
|-------------------------------------------|---------------------------------|---------------------------------|------------------------------------------|-------------------------------------------|
| Information                               | App. A                          | App. A                          | App. C                                   | App. C                                    |
| $V_{DD}$ ( $V_{SS} = 0$ )                 | 5 V                             | 5 V                             | 3.3 V                                    | 3.3 V                                     |
| minimum $L_{drawn}$                       | 2.0 $\mu\text{m}$               | 2.0 $\mu\text{m}$               | 0.6 $\mu\text{m}$                        | 0.6 $\mu\text{m}$                         |
| $DL = L_{drawn} - L_{eff}$                | 0.6 $\mu\text{m}$               | 0.8 $\mu\text{m}$               | 0.06 $\mu\text{m}$                       | 0.09 $\mu\text{m}$                        |
| minimum $W_{drawn}$                       | 3.0 $\mu\text{m}$               | 3.0 $\mu\text{m}$               | 0.9 $\mu\text{m}$                        | 0.9 $\mu\text{m}$                         |
| $DW = W_{drawn} - W_{eff}$                | 0.14 $\mu\text{m}$              | 0.16 $\mu\text{m}$              | 0.35 $\mu\text{m}$                       | 0.35 $\mu\text{m}$                        |
| $V_{THN}$ or $V_{THP}$                    | 0.83 V                          | 0.91 V                          | 0.7 V                                    | 0.9 V                                     |
| KP, $\mu\text{A}/\text{V}^2$              | 50                              | 17                              | 170                                      | 50                                        |
| $\lambda$ , $\text{V}^{-1}$               | 0.06 for $L \geq 5 \mu\text{m}$ | 0.06 for $L \geq 5 \mu\text{m}$ |                                          |                                           |
| $I_{drive}$ ( $\mu\text{A}/\mu\text{m}$ ) |                                 |                                 | 380 (typ.)                               | 190 (typ.)                                |
| $C'_{ox}$                                 | 800 aF/ $\mu\text{m}^2$         | 800 aF/ $\mu\text{m}^2$         | 3.7 fF/ $\mu\text{m}^2$                  | 3.7 fF/ $\mu\text{m}^2$                   |
| $R_n$ or $R_p$                            | 12 $\text{k}\Omega \cdot (L/W)$ | 36 $\text{k}\Omega \cdot (L/W)$ | (9 $\text{k}\Omega \cdot \mu\text{m})/W$ | (18 $\text{k}\Omega \cdot \mu\text{m})/W$ |
| $\tau_n$ or $\tau_p$                      | 38 ps                           | 114 ps                          | 20 ps                                    | 40ps                                      |

## Multipliers

| Name   | Symbol       | Value      |
|--------|--------------|------------|
|        |              |            |
| terra  | T            | $10^{12}$  |
| giga   | G            | $10^9$     |
| mega   | M or (MEG)   | $10^6$     |
| kilo   | k            | $10^3$     |
| milli  | m            | $10^{-3}$  |
| micro  | $\mu$ (or u) | $10^{-6}$  |
| nano   | n            | $10^{-9}$  |
| pico   | p            | $10^{-12}$ |
| fernto | f            | $10^{-15}$ |
| atto   | a            | $10^{-18}$ |

## Physical Constants

| Name                                 | Symbol                 | Value/Units                |
|--------------------------------------|------------------------|----------------------------|
| Vacuum dielectric constant           | $\epsilon_0$           | 8.85 aF/ $\mu\text{m}$     |
| Silicon dielectric constant          | $\epsilon_{\text{si}}$ | $11.7\epsilon_0$           |
| SiO <sub>2</sub> dielectric constant | $\epsilon_{\text{ox}}$ | $3.97\epsilon_0$           |
| SiN <sub>3</sub> dielectric constant | $\epsilon_{\text{Ni}}$ | $16\epsilon_0$             |
| Boltzmann's constant                 | k                      | $1.38 \times 10^{-23}$ J/K |
| Electronic charge                    | q                      | $1.6 \times 10^{-19}$ C    |
| Temperature                          | T                      | K                          |
| Thermal voltage                      | $V_T$                  | kT/q<br>= 26 mV @ 300 K    |

## Equations

| Parameter                                            | NMOS equations in terms of BSIM1 parameters                                            |
|------------------------------------------------------|----------------------------------------------------------------------------------------|
|                                                      | for PMOS use $V_{SD}$ , $V_{SG}$ , $V_{BS}$ , and $V_{THP}$                            |
| $V_{THN}$                                            | $V_{FB} + PHI + K1 \cdot \sqrt{PHI + V_{SB}} - K2 \cdot (PHI + V_{SB})$                |
| $C'_{ox}$                                            | $\epsilon_{ox}/TOX$                                                                    |
| KP                                                   | $MUZ \cdot C'_{ox}$                                                                    |
| $\beta$                                              | $KP \cdot (W/L)$                                                                       |
| $I_D$ (triode)<br>$V_{DS} \leq V_{GS} - V_{THN}$     | $\beta \left( (V_{GS} - V_{THN})V_{DS} - \frac{V_{DS}^2}{2} \right)$ (for Long L)      |
| $I_D$ (saturation)<br>$V_{DS} \geq V_{GS} - V_{THN}$ | $\frac{\beta}{2} (V_{GS} - V_{THN})^2 [1 + \lambda(V_{DS} - V_{DSsat})]$ (for Long L)  |
| $g_m$                                                | $\beta(V_{GS} - V_{THN}) = \sqrt{2\beta I_D}$ or $I_D / (V_T \cdot N0)$                |
| $\eta$                                               | $K1 \cdot \left( 2\sqrt{PHI + V_{SB}} \right)^{-1} - K2$ and $g_{mb} = g_m \cdot \eta$ |
| $r_o$                                                | $1/(\lambda I_D)$                                                                      |

IEEE Press  
445 Hoes Lane, P.O. Box 1331  
Piscataway, NJ 08855-1331

**Editorial Board**

John B. Anderson, *Editor in Chief*

|                 |                  |              |
|-----------------|------------------|--------------|
| P. M. Anderson  | R. Herrick       | R. S. Muller |
| M. Eden         | G. F. Hoffnagle  | W. D. Reeve  |
| M. E. El-Hawary | R. F. Hoyt       | D. J. Wells  |
| S. Furui        | S. Kartalopoulos |              |
| A. H. Haddad    | P. Laplante      |              |

Kenneth Moore, *Director of IEEE Press*

John Griffin, *Senior Editor*

Linda Matarazzo, *Assistant Editor*

IEEE Circuits & Systems Society, *Sponsor*

CAS-S Liaison to IEEE Press, Phillip Allen

IEEE Solid-State Circuits Society, *Sponsor*

SSC-S Liaison to IEEE Press, Stuart K. Tewksbury

**Technical Reviewers**

Jeff Bruce, *Micron Technology, Inc.*

Alan Buchholz, *Comlinear Corporation*

Ian Galton, *University of California, Irvine*

Bruce Johnson, *University of Nevada, Reno*

Joseph Karniewicz, *Micron Technology, Inc.*

Brent Keeth, *Micron Technology, Inc.*

William Kuhn, *Kansas State University*

Wen Li, *Micron Technology, Inc.*

Brian P. Lum Shue Chan, *Crystal Semiconductor Corporation*

Alan Mantooth, *Analogy, Inc.*

Adrian Ong, *Stanford University*

Terry Sculley, *Crystal Semiconductor Corporation*

Don Thelen, *Analog Interfaces*

Axel Thomsen, *Crystal Semiconductor Corporation*

Kwang Yoon, *Inha University, Korea*

# **CMOS**

## **Circuit Design, Layout, and Simulation**

**R. Jacob Baker, Harry W. Li and David E. Boyce**  
*Department of Electrical Engineering*  
*Microelectronics Research Center*  
*The University of Idaho*

IEEE Press Series on Microelectronic Systems  
Stuart K. Tewksbury, *Series Editor*

IEEE Circuits & Systems Society, *Sponsor*  
IEEE Solid-State Circuits Society, *Sponsor*



The Institute of Electrical and Electronics Engineers, Inc., New York

7571.49  
This book and other books may be purchased at a discount from the publisher when ordered in bulk quantities. Contact:

IEEE Press Marketing  
Attn: Special Sales  
Piscataway, NJ 08855-1331  
Fax: (732) 981-9334

For more information about IEEE PRESS products,  
visit the IEEE Home Page: <http://www.ieee.org/>

© 1998 by the Institute of Electrical and Electronics Engineers, Inc.  
345 East 47th Street, New York, NY 10017-2394

*All rights reserved. No part of this book may be reproduced in any form, nor may it be stored in a retrieval system or transmitted in any form, without written permission from the publisher.*

Printed in the United States of America

10 9 8 7 6 5 4 3

**ISBN 0-7803-3416-7**

**IEEE Order Number: PC5689**

#### **Library of Congress Cataloging-in-Publication Data**

Baker, R. Jacob (date)

CMOS circuit design, layout, and simulation / R. Jacob Baker,

Harry W. Li, and David E. Boyce.

p. cm. -- (IEEE Press series on microelectronic systems)

Includes bibliographical references and index.

ISBN 0-7803-3416-7

1. Metal oxide semiconductors, Complementary--Design and construction. 2. Integrated circuits--Design and construction.

3. Metal oxide semiconductor field-effect transistors. I. Li,

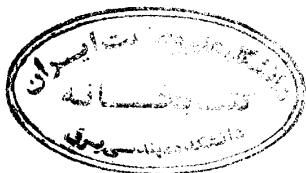
Harry W. II. Boyce, David E. III. Title. IV. Series

TK7871.99.M44B35 1997

621.3815--DC21

97-21906

CIP



*To*  
*Julie and Melanie*





# Contents

|                                      |            |
|--------------------------------------|------------|
| <b>Preface</b>                       | <b>xix</b> |
| <b>Part I CMOS Fundamentals</b>      | <b>1</b>   |
| <b>Chapter 1 Introduction</b>        | <b>3</b>   |
| 1.1 The CMOS IC Design Process ..... | 3          |
| 1.1.1 Fabrication .....              | 5          |
| 1.2 Using the LASI Program .....     | 5          |
| 1.2.1 Cells in LASI .....            | 7          |
| 1.2.2 Navigating LASI .....          | 10         |
| 1.2.3 Adding Objects .....           | 10         |
| 1.2.4 Editing Objects .....          | 11         |
| 1.2.5 Placing Cells .....            | 12         |
| 1.2.6 Common Problems .....          | 14         |
| 1.3 MOSIS .....                      | 15         |
| <b>Chapter 2 The Well</b>            | <b>23</b>  |
| 2.1 The Substrate .....              | 23         |
| 2.1.1 Patterning .....               | 24         |
| 2.1.2 Patterning the N-well .....    | 27         |
| 2.2 Laying out the N-well .....      | 27         |

|                                                    |           |
|----------------------------------------------------|-----------|
| 2.2.1 Design Rules for the N-well                  | 28        |
| 2.2.2 Using the LASIDRC program                    | 30        |
| 2.3 Resistance Calculation                         | 30        |
| 2.3.1 The N-well Resistor                          | 32        |
| 2.4 The N-well/Substrate Diode                     | 32        |
| 2.4.1 Depletion Layer Capacitance                  | 33        |
| 2.4.2 Storage Capacitance                          | 35        |
| 2.4.3 SPICE Modeling                               | 37        |
| 2.5 The RC Delay Through an N-well                 | 38        |
| <b>Chapter 3 The Metal Layers</b>                  | <b>45</b> |
| 3.1 The Bonding Pad                                | 45        |
| 3.1.1 Laying out the Pad                           | 45        |
| 3.1.2 Design Rules for Pads                        | 48        |
| 3.2 Design and Layout Using the Metal Layers       | 50        |
| 3.2.1 Design Rules for the Metal Layers            | 50        |
| 3.2.2 Parasitics Associated with the Metal Layers  | 50        |
| 3.2.3 Current Carrying Limitations                 | 54        |
| 3.2.4 Parasitics Associated with the Via           | 55        |
| 3.3 Crosstalk and Ground Bounce                    | 56        |
| 3.3.1 Ground Bounce                                | 57        |
| 3.4 Layout Using Cell Hierarchy                    | 59        |
| <b>Chapter 4 The Active and Poly Layers</b>        | <b>65</b> |
| 4.1 Design Rules                                   | 66        |
| 4.1.1 Design Rules for the n+/p+ Active Layers     | 66        |
| 4.1.2 Design Rules for Poly1                       | 69        |
| 4.2 Layout of a Standard Cell Frame                | 71        |
| 4.3 Patterning the Active Layers                   | 71        |
| 4.4 Layout of the MOSFET                           | 77        |
| 4.4.1 Parasitics Associated with the Active Layers | 77        |
| <b>Chapter 5 The MOSFET</b>                        | <b>83</b> |
| 5.1 The MOSFET Capacitances                        | 84        |
| 5.1.1 Case I: Accumulation                         | 84        |
| 5.1.2 Case II: Depletion                           | 85        |

|                                                                    |            |
|--------------------------------------------------------------------|------------|
| Contents                                                           | ix         |
| 5.1.3 Case III: Strong Inversion                                   | 86         |
| 5.1.4 Summary                                                      | 87         |
| 5.2 The Threshold Voltage                                          | 88         |
| 5.3 IV Characteristics of MOSFETs                                  | 93         |
| 5.3.1 MOSFET Operation in the Triode Region                        | 94         |
| 5.3.2 The Saturation Region                                        | 96         |
| 5.4 SPICE Modeling of the MOSFET                                   | 98         |
| 5.4.1 Level 1 Model Parameters Related to $V_{THN}$                | 98         |
| 5.4.2 Level 1 Model Parameters Related to Transconductance         | 99         |
| 5.4.3 SPICE Modeling of the Source and Drain Implants              | 100        |
| 5.4.4 Layout of the MOSFET                                         | 102        |
| <b>Chapter 6 The BSIM SPICE Model</b>                              | <b>107</b> |
| 6.1 BSIM1 Model Parameters                                         | 109        |
| 6.2 BSIM1 DC Equations                                             | 111        |
| 6.2.1 The Threshold Voltage                                        | 111        |
| 6.2.2 The Drain Current                                            | 113        |
| 6.2.3 The Subthreshold Current                                     | 116        |
| 6.3 Short Channel MOSFETs                                          | 119        |
| 6.3.1 MOSFET Scaling                                               | 120        |
| 6.3.2 Short Channel Effects                                        | 122        |
| 6.4 The BSIM3 SPICE Model                                          | 124        |
| 6.5 Convergence                                                    | 127        |
| <b>Chapter 7 CMOS Passive Elements</b>                             | <b>131</b> |
| 7.1 The Second Poly Layer (poly2)                                  | 131        |
| 7.1.1 Design Rules for Capacitor Formation                         | 131        |
| 7.1.2 Parasitics of the Poly Cap                                   | 132        |
| 7.1.3 Other Types of Capacitors                                    | 134        |
| 7.2 Temperature and Voltage Dependence of Capacitors and Resistors | 134        |
| 7.2.1 Resistors                                                    | 134        |
| 7.2.2 Capacitors                                                   | 140        |
| 7.3 Noise in Resistors                                             | 141        |
| <b>Chapter 8 Design Verification with LASICKT</b>                  | <b>149</b> |
| 8.1 Fundamentals of LASICKT                                        | 150        |

|                                                      |            |
|------------------------------------------------------|------------|
| 8.1.1 The Inverter                                   | 151        |
| 8.1.2 Running LASICKT                                | 157        |
| 8.1.3 Higher-ranking Cells; The OR Gate              | 160        |
| <b>Chapter 9 Analog MOSFET Models</b>                | <b>165</b> |
| 9.1 Low-Frequency MOSFET Model                       | 165        |
| 9.1.1 Small-Signal Model of the MOSFET in Saturation | 166        |
| 9.2 High-Frequency MOSFET Model                      | 171        |
| 9.2.1 Variation of Transconductance with Frequency   | 173        |
| 9.3 Temperature Effects in MOSFETs                   | 175        |
| 9.4 Noise in MOSFETs                                 | 178        |
| <b>Chapter 10 The Digital Model</b>                  | <b>185</b> |
| 10.1 The Digital MOSFET Model                        | 185        |
| 10.1.1 Capacitive Effects                            | 187        |
| 10.1.2 Process Characteristic Time Constant          | 188        |
| 10.1.3 Delay- and Transition-Times                   | 189        |
| 10.2 Series Connection of MOSFETs                    | 192        |
| 10.2.1 DC Behavior of Series-Connected MOSFETs       | 192        |
| 10.2.2 Delay Through Series-Connected MOSFETs        | 194        |
| Possible Student Projects                            | 196        |
| <b>Part II CMOS Digital Circuits</b>                 | <b>199</b> |
| <b>Chapter 11 The Inverter</b>                       | <b>201</b> |
| 11.1 DC Characteristics                              | 202        |
| 11.1.1 Noise Margins                                 | 203        |
| 11.1.2 Inverter Switching Point                      | 204        |
| 11.2 Switching Characteristics                       | 205        |
| 11.2.1 The Ring Oscillator                           | 209        |
| 11.2.2 Dynamic Power Dissipation                     | 210        |
| 11.3 Layout of the Inverter                          | 212        |
| 11.3.1 Latch-up                                      | 212        |
| 11.4 Sizing for Large Capacitive Loads               | 216        |
| 11.4.1 Distributed Drivers                           | 221        |

|                                                   |            |
|---------------------------------------------------|------------|
| Contents                                          | xi         |
| 11.4.2 Driving Long Lines                         | 223        |
| 11.5 Other Inverter Configurations                | 224        |
| 11.5.1 N-Channel Only Output Drivers              | 224        |
| 11.5.2 Inverters with Tri-State Outputs           | 226        |
| 11.5.3 The Bootstrapped NMOS inverter             | 226        |
| <b>Chapter 12 Static Logic Gates</b>              | <b>231</b> |
| 12.1 DC Characteristics of the NAND and NOR Gates | 231        |
| 12.1.1 DC Characteristics of the NAND Gate        | 231        |
| 12.1.2 DC Characteristics of the NOR Gate         | 234        |
| 12.2 Layout of the NOR and NAND Gates             | 235        |
| 12.3 Switching Characteristics                    | 236        |
| 12.3.1 NAND Gate                                  | 238        |
| 12.3.2 Number of Inputs                           | 242        |
| 12.4 Complex CMOS Logic Gates                     | 243        |
| 12.4.1 Cascode Voltage Switch Logic               | 247        |
| 12.4.2 Differential Split-Level Logic             | 247        |
| 12.4.3 Tri-State Outputs                          | 252        |
| <b>Chapter 13 The TG and Flip-Flops</b>           | <b>255</b> |
| 13.1 The Pass Transistor                          | 255        |
| 13.2 The CMOS TG                                  | 257        |
| 13.2.1 Layout of the CMOS TG                      | 258        |
| 13.2.2 Series Connection of Transmission Gates    | 260        |
| 13.3 Applications of the Transmission Gate        | 260        |
| 13.4 The Flip-Flop                                | 265        |
| 13.4.1 Clocked Flip-Flops                         | 266        |
| <b>Chapter 14 Dynamic Logic Gates</b>             | <b>275</b> |
| 14.1 Fundamentals of Dynamic Logic                | 275        |
| 14.1.1 Charge Leakage                             | 276        |
| 14.1.2 Simulating Dynamic Circuits                | 277        |
| 14.1.3 Nonoverlapping Clock Generation            | 279        |
| 14.1.4 CMOS TG in Dynamic Circuits                | 281        |
| 14.2 Clocked CMOS Logic                           | 281        |

|                                                    |            |
|----------------------------------------------------|------------|
| <b>Chapter 15 VLSI Layout</b>                      | <b>289</b> |
| 15.1 Chip Layout .....                             | 290        |
| 15.2 Layout Steps .....                            | 301        |
| <b>Chapter 16 BiCMOS Logic Gates</b>               | <b>313</b> |
| 16.1 Layout of the Junction-Isolated BJT .....     | 314        |
| 16.2 Modeling the NPN .....                        | 316        |
| 16.3 The BiCMOS Inverter .....                     | 320        |
| 16.4 Other BiCMOS Logic Gates .....                | 323        |
| 16.5 CMOS and ECL Conversions Using BiCMOS .....   | 326        |
| <b>Chapter 17 Memory Circuits</b>                  | <b>331</b> |
| 17.1 RAM Memory Cells .....                        | 332        |
| 17.1.1 The DRAM Cell .....                         | 333        |
| 17.1.2 The SRAM cell .....                         | 337        |
| 17.2 The Sense Amplifier .....                     | 338        |
| 17.3 Row/Column Decoders .....                     | 341        |
| 17.4 Timing Requirements for DRAMs .....           | 343        |
| 17.5 Modern DRAM Circuits .....                    | 345        |
| 17.5.1 DRAM Memory Cell Layout .....               | 345        |
| 17.5.2 Folded/Open Architectures .....             | 347        |
| 17.6 Other Memory Cells .....                      | 350        |
| <b>Chapter 18 Special-Purpose Digital Circuits</b> | <b>355</b> |
| 18.1 The Schmitt Trigger .....                     | 355        |
| 18.1.1 Design of the Schmitt Trigger .....         | 356        |
| 18.1.2 Switching Characteristics .....             | 359        |
| 18.1.3 Applications of the Schmitt Trigger .....   | 359        |
| 18.1.4 High-Speed Schmitt Trigger .....            | 361        |
| 18.2 Multivibrator Circuits .....                  | 362        |
| 18.2.1 The Monostable Multivibrator .....          | 362        |
| 18.2.2 The Astable Multivibrator .....             | 363        |
| 18.3 Voltage Generators .....                      | 364        |
| 18.3.1 Improving the Efficiency .....              | 366        |
| 18.3.2 Generating Higher Voltages .....            | 367        |
| 18.3.3 Example .....                               | 368        |

|                                              |            |
|----------------------------------------------|------------|
| <b>Chapter 19 Digital Phase-Locked Loops</b> | <b>373</b> |
| 19.1 The Phase Detector .....                | 375        |
| 19.1.1 The XOR Phase Detector                | 375        |
| 19.1.2 Phase Frequency Detector              | 379        |
| 19.2 The Voltage-Controlled Oscillator ..... | 383        |
| 19.2.1 The Current-Starved VCO               | 384        |
| 19.2.2 Source Coupled VCOs                   | 387        |
| 19.3 The Loop Filter .....                   | 391        |
| 19.3.1 XOR DPLL                              | 391        |
| 19.3.2 PFD DPLL                              | 398        |
| 19.4 System Considerations .....             | 403        |
| 19.4.1 Clock Recovery from NRZ Data          | 407        |
| 19.5 Delay-Locked Loops .....                | 417        |

## **Part III CMOS Analog Circuits** **425**

|                                                  |            |
|--------------------------------------------------|------------|
| <b>Chapter 20 Current Sources and Sinks</b>      | <b>427</b> |
| 20.1 The Current Mirror .....                    | 427        |
| 20.1.1 The Cascode Connection                    | 433        |
| 20.1.2 Sensitivity Analysis                      | 438        |
| 20.1.3 Temperature Analysis                      | 439        |
| 20.1.4 Transient Response                        | 442        |
| 20.1.5 Layout of the Simple Current Mirror       | 444        |
| 20.1.6 Matching in MOSFET Mirrors                | 446        |
| 20.2 Other Current Sources/Sinks .....           | 450        |
| <b>Chapter 21 References</b>                     | <b>463</b> |
| 21.1 Voltage Dividers .....                      | 463        |
| 21.1.1 The Resistor-MOSFET Divider               | 464        |
| 21.1.2 The MOSFET-Only Voltage Divider           | 465        |
| 21.2 Current Source Self-Biasing .....           | 469        |
| 21.2.1 Threshold Voltage Referenced Self-Biasing | 469        |
| 21.2.2 Diode Referenced Self-Biasing             | 470        |
| 21.2.3 Thermal Voltage Referenced Self-Biasing   | 474        |

|                                                                |            |
|----------------------------------------------------------------|------------|
| 21.3 Bandgap Voltage References .....                          | 477        |
| 21.3.1 Bandgap Referenced Biasing .....                        | 478        |
| 21.4 Beta Multiplier Referenced Self-Biasing .....             | 480        |
| 21.4.1 A Voltage Reference .....                               | 481        |
| 21.4.2 Operation in the Subthreshold Region .....              | 483        |
| <b>Chapter 22 Amplifiers</b> .....                             | <b>489</b> |
| 22.1 Gate-Drain Connected Loads .....                          | 489        |
| 22.1.1 Common Source Amplifiers .....                          | 489        |
| 22.1.2 The Source Follower .....                               | 498        |
| 22.1.3 Common Gate Amplifiers .....                            | 500        |
| 22.2 Current Source Loads .....                                | 500        |
| 22.2.1 The Cascode Connection .....                            | 506        |
| 22.2.2 The Push-Pull Amplifier .....                           | 510        |
| 22.3 Noise and Distortion in Amplifiers .....                  | 512        |
| 22.3.1 Modeling Amplifier Noise .....                          | 515        |
| 22.4 A Class AB Amplifier .....                                | 517        |
| <b>Chapter 23 Feedback Amplifiers</b> .....                    | <b>525</b> |
| 23.1 The Feedback Equation .....                               | 526        |
| 23.2 Properties of Negative Feedback on Amplifier Design ..... | 527        |
| 23.2.1 Gain Desensitivity .....                                | 527        |
| 23.2.2 Bandwidth Extension .....                               | 528        |
| 23.2.3 Reduction in Nonlinear Distortion .....                 | 529        |
| 23.2.4 Input and Output Impedance Control .....                | 530        |
| 23.3 Recognizing Feedback Topologies .....                     | 532        |
| 23.3.1 Input Mixing .....                                      | 533        |
| 23.3.2 Output Sampling .....                                   | 533        |
| 23.3.3 The Feedback Network .....                              | 534        |
| 23.3.4 Calculating Open-Loop Parameters .....                  | 537        |
| 23.3.5 Calculating Closed-Loop Parameters .....                | 539        |
| 23.4 The Voltage Amp (Series-Shunt Feedback) .....             | 540        |
| 23.5 The Transimpedance Amp (Shunt-Shunt Feedback) .....       | 547        |
| 23.5.1 Simple Feedback Using a Gate-Drain Resistor .....       | 555        |
| 23.6 The Transconductance Amp (Series-Series Feedback) .....   | 558        |
| 23.7 The Current Amplifier (Shunt-Series Feedback) .....       | 562        |



|                                                     |            |
|-----------------------------------------------------|------------|
| 23.8 Stability .....                                | 564        |
| 23.8.1 The Return Ratio .....                       | 568        |
| <b>Chapter 24 Differential Amplifiers</b>           | <b>579</b> |
| 24.1 The Source Coupled Pair .....                  | 579        |
| 24.1.1 Current Source Load .....                    | 582        |
| 24.1.2 Common-Mode Rejection Ratio .....            | 591        |
| 24.1.3 Noise .....                                  | 592        |
| 24.1.4 Matching Considerations .....                | 594        |
| 24.2 The Source Cross-Coupled Pair .....            | 596        |
| 24.2.1 Current Source Load .....                    | 599        |
| 24.3 Cascode Loads .....                            | 602        |
| 24.4 Wide-Swing Differential Amplifiers .....       | 605        |
| 24.4.1 Current Differential Amplifier .....         | 607        |
| 24.4.2 Constant Transconductance Diff-Amp .....     | 608        |
| <b>Chapter 25 Operational Amplifiers</b>            | <b>617</b> |
| 25.1 Basic CMOS Op-Amp Design .....                 | 618        |
| 25.1.1 Characterizing the Op-Amp .....              | 630        |
| 25.1.2 Compensating the Op-Amp Without Buffer ..... | 634        |
| 25.1.3 The Cascode Input Op-Amp .....               | 636        |
| 25.2 Operational Transconductance Amplifiers .....  | 637        |
| 25.2.1 Wide-Swing OTA .....                         | 642        |
| 25.2.2 The Folded-Cascode OTA .....                 | 654        |
| 25.3 The Differential Output Op-Amp .....           | 664        |
| 25.3.1 Fully Differential Folded-Cascode OTA .....  | 666        |
| 25.3.2 Gain Enhancement .....                       | 674        |
| <b>Part IV Mixed-Signal Circuits</b>                | <b>683</b> |
| <b>Chapter 26 Nonlinear Analog Circuits</b>         | <b>685</b> |
| 26.1 Basic CMOS Comparator Design .....             | 685        |
| 26.1.1 Characterizing the Comparator .....          | 691        |
| 26.2 Adaptive Biasing .....                         | 699        |
| 26.3 Analog Multipliers .....                       | 704        |

|                                                       |            |
|-------------------------------------------------------|------------|
| 26.3.1 The Multiplying Quad                           | 705        |
| 26.3.2 Level Shifting                                 | 710        |
| 26.3.3 Multiplier Design Using Squaring Circuits      | 715        |
| <b>Chapter 27 Dynamic Analog Circuits</b>             | <b>719</b> |
| 27.1 The MOSFET Switch                                | 719        |
| 27.2 Switched-Capacitor Circuits                      | 728        |
| 27.2.1 Switched-Capacitor Integrator                  | 731        |
| 27.3 Circuits                                         | 745        |
| <b>Chapter 28 Data Converter Fundamentals</b>         | <b>755</b> |
| 28.1 Analog Versus Discrete Time Signals              | 755        |
| 28.2 Converting Analog Signals to Digital Signals     | 756        |
| 28.3 Sample-and-Hold (S/H) Characteristics            | 759        |
| 28.4 Digital-to-Analog Converter (DAC) Specifications | 762        |
| 28.5 Analog-to-Digital Converter (ADC) Specifications | 772        |
| 28.6 Mixed-Signal Layout Issues                       | 783        |
| <b>Chapter 29 Data Converter Architectures</b>        | <b>791</b> |
| 29.1 DAC Architectures                                | 791        |
| 29.1.1 Digital Input Code                             | 791        |
| 29.1.2 Resistor String                                | 792        |
| 29.1.3 R-2R Ladder Networks                           | 797        |
| 29.1.4 Current Steering                               | 800        |
| 29.1.5 Charge Scaling DACs                            | 805        |
| 29.1.6 Cyclic DAC                                     | 810        |
| 29.1.7 Pipeline DAC                                   | 811        |
| 29.2 ADC Architectures                                | 813        |
| 29.2.1 Flash                                          | 813        |
| 29.2.2 The Two-Step Flash ADC                         | 818        |
| 29.2.3 The Pipeline ADC                               | 823        |
| 29.2.4 Integrating ADCs                               | 827        |
| 29.2.5 The Successive Approximation ADC               | 832        |
| 29.2.6 The Oversampling ADC                           | 837        |
| <b>Appendix A Orbit's CN20 Process</b>                | <b>855</b> |
| A.1 Process Specifications                            | 857        |

|                                               |            |
|-----------------------------------------------|------------|
| Contents                                      | xvii       |
| A.1.1 Electrical Specifications               | 857        |
| A.1.2 N-Channel SPICE Models                  | 859        |
| A.1.3 P-Channel SPICE Models                  | 861        |
| A.2 Hand Calculations .....                   | 862        |
| A.2.1 The N-channel MOSFET Equations          | 862        |
| A.2.2 The P-channel MOSFET Equations          | 865        |
| A.3 Design Rules .....                        | 867        |
| <b>Appendix B MOSIS Scalable Design Rules</b> | <b>873</b> |
| <b>Appendix C HP's CMOS14TB</b>               | <b>885</b> |
| <b>Index</b>                                  | <b>893</b> |
| <b>About the Authors</b>                      | <b>903</b> |
| <b>LASI Software</b>                          | <b>904</b> |



# Preface

Over the last ten years the electronics industry has exploded. A recent report by the Semiconductor Industry Association (SIA) [1] proclaimed that in 1995 alone, world chip revenues increased by 41.7 percent and for the past five years the growth had been exponential. By the year 1999, the report estimates that world chip sales will surpass \$234.5 billion, up from \$154 billion in 1996. The largest portion of total worldwide sales is dominated by the MOS market. Composed primarily of memory, micro and logic sales, the total combined MOS revenue contributed approximately 75 percent of total world-wide sales (\$114.2 billion), illustrating the strength of CMOS technology. The percentage of MOS sales relative to all chip revenues is expected to remain constant through 1999, when MOS sales will total \$178 billion.

CMOS technology continues to mature, with minimum feature sizes now approaching 0.1  $\mu\text{m}$ . Texas Instruments recently announced a 0.18  $\mu\text{m}$  process [2] in which the equivalent of 20 high-performance microprocessors could exist on the same substrate, with a transistor density of 125 million transistors. This high density allows for true system-level integration on a chip, with digital signal processors, microprocessors or microcontroller cores, memory, analog or mixed-signal functions all residing on the same die.

As educators we are often asked by our students, "Isn't analog dead? I thought everything was going digital!" How untrue! The prediction of the future demise of analog electronics has been around since the mid-1970s. According to the SIA report [1], the revenues generated by analog products closely parallel the MOS logic market and achieved a 22.5 percent increase in 1995. The analog market expects to reach \$18.2 billion in 1996 (a 9.5 percent increase) with double-digit growth projected for the next three years. In 1999, the total revenues generated by analog sales is forecasted to peak at \$26.6 billion (11.3 percent of total chip sales!). However, while there is still

demand for analog designers, their role is definitely changing. As was communicated by Paul Gray in [3], the days of pure analog design are over, meaning that very few systems remain purely analog. More and more systems are integrated, with increased functionality being performed in the digital domain. He goes on to state that the analog designer should become broad-based, with analog transistor-level design as the core skill. This means that the analog designer should also

- Have a good understanding of digital very large scale integration (VLSI) and be competent at using the latest computer-aided design (CAD) tools.
- know how to apply digital signal processing (DSP), analog signal processing (ASP), and filtering concepts to system-level design.
- possess insight into system implications of component-level performance.

For example, DSP and transistor-level analog design skills are needed for oversampling applications such as data converters, filtering, and a host of relatively new circuit topologies based on sigma-delta modulation. Being able to design both analog and digital circuits, as well as understand the interactions between the two domains, will provide an added dimension to a designer's portfolio that is difficult to match. Analog designers are in demand more than ever, simply because the end limitations of digital electronics need to be examined under the "analog" microscope to fully understand the mechanisms that are occurring. Therefore, this text attempts to combine digital and analog IC design in one complete reference.

Layout is the process of physically defining the layers that compose an integrated circuit. Typically, layouts are constructed using a computer-aided design program. CAD companies such as Mentor Graphics, Synopsis, and Cadence specialize in providing extremely powerful CAD software for the entire integrated circuit (IC) design process, including design, synthesis, simulation, and layout tools within an integrated framework. These workstation-based software tools can literally cost millions of dollars, but provide convenient and powerful features found nowhere else. CAD tools also exist for the PC. Tanner Tool's L-Edit provides a complete IC design CAD program for the personal computer. The program discussed in this book, Layout System for Individuals (LASI) (pronounced "LAZY"), also provides the student with the ability to lay out ICs on a PC and includes design rule checking and design verification capability. It is distributed as shareware, free for educational purposes.

With decreases in feature size come added complexities in the design. Layouts must now be considered heavily in the design process as matching and parasitic effects become the limiting factors in many precision and high-speed applications. The more the designer knows about the process with respect to layout and modeling, the more performance the engineer can "squeeze" out the design. However, performance is not the only reason to consider the layout. The economic impact of IC layouts can be detrimental to the circuit's marketing potential. In some cases a 20 percent increase in chip area can reduce the profits of a chip by several hundreds of thousands of dollars. Chip area should be considered as premium real estate. Therefore, much of the first ten

chapters of this book is devoted to fundamental layout issues, with other issues presented as the need arises.

Modeling is also a key issue. A simulation is only as accurate as its model. Although the Berkeley Short-channel IGFET Model (BSIM) model has become the industry standard its relatively nonintuitive structure makes hand analysis using BSIM model parameters an intimidating process. To many students (and engineers), the BSIM parameters are nothing more than sets of numbers at the end of their SPICE decks. However, some very useful information can be gleaned from the BSIM model which helps make the hand analysis more closely resemble the simulated result. Chapter 6 provides a great deal of information that relates the BSIM model to first-order hand-analysis equations.

A successful CMOS integrated circuit design engineer has knowledge in the areas of device operation, circuit design, layout, and simulation. Students learning CMOS IC design should be trained at a fundamental level in these areas. In the past, courses on CMOS integrated circuits dealt mainly with circuit design or analysis. Little to no time was spent on layout of the integrated circuits. This may have been justified. It is difficult to find a reason to lay out an entire chip and then not have the chip fabricated. However, through the use of the MOSIS<sup>1</sup> program, students can submit their chip designs for fabrication through one of the MOSIS contracted vendors. In approximately ten weeks the chips are returned to the university for evaluation. The MOSIS program is an outstanding way of introducing students to the design of ICs.

Although many texts [4-32] are available covering some aspects of CMOS analog or digital circuit design, none integrates the coverage of both topics with layout and includes layout software as is done in this text. Our focus, when writing this text, was on the fundamentals of custom CMOS integrated circuit design. It was our goal that a student who studies and masters the material in this text will possess the fundamental skills needed to design high-performance analog and digital CMOS circuits and have the basic understanding and problem-solving skills needed to enhance the performance of an IC or to determine why an IC doesn't function as simulated.

#### *Use of This Text*

This text can be used for two courses. Both courses can be offered at the senior/first-year graduate level. The first course concentrates on the physical design of CMOS digital integrated circuits with prerequisites of junior level Electronics I and a course on digital logic design. A possible semester course outline is as follows.

|        |                                                                                                                         |
|--------|-------------------------------------------------------------------------------------------------------------------------|
| Week 1 | Chs. 1 & 2, introduction, course requirements, layout and SPICE demonstrations, the n-well, sheet resistance.           |
| Week 2 | Chs. 2 & 3, the n-well, pn junction, capacitance, resistance, delay through the well, introduction to the metal layers. |

---

<sup>1</sup> MOSIS - MOS Implementation System through the Information Sciences Institute at the University of Southern California, (310) 822-1511 or <http://www.mosis.org>

|         |                                                                                                                                                  |
|---------|--------------------------------------------------------------------------------------------------------------------------------------------------|
| Week 3  | Chs. 3 & 4, the metal layers, parasitics, electromigration, layout of the padframe, active/poly layers, layout of the MOSFET and standard frame. |
| Week 4  | Ch. 5, MOSFET operation                                                                                                                          |
| Week 5  | Chs. 5 & 6, completion of MOSFET operation, discussion of modeling using the BSIM model.                                                         |
| Week 6  | Chs. 6 & 7, completion of BSIM model, layout of a capacitor, MOS temp dependence.                                                                |
| Week 7  | Chs. 10 & 11, digital models and the inverter.                                                                                                   |
| Week 8  | Ch. 11, the inverter, switching point voltage and switching times, layout, latch-up, and design.                                                 |
| Week 9  | Ch. 12, static logic gates, switching point voltages, speed, and layout.                                                                         |
| Week 10 | Chs. 13 & 14, the transmission gate, flip-flops, and dynamic logic gates.                                                                        |
| Week 11 | Chs. 15 & 16, VLSI layout and BiCMOS logic.                                                                                                      |
| Week 12 | Ch. 17, memory circuits, basic memory cells, and organization.                                                                                   |
| Week 13 | Ch. 18, special-purpose digital circuits.                                                                                                        |
| Week 14 | Ch. 19, introduction to digital phase locked loops, phase detectors, VCOs.                                                                       |
| Week 15 | Ch. 19, digital PLL design.                                                                                                                      |

The second course concentrates on CMOS analog circuit design. A possible semester course outline is as follows.

|             |                                                       |
|-------------|-------------------------------------------------------|
| Weeks 1 & 2 | Review of Chs. 1-6.                                   |
| Week 3      | Chs. 7, CMOS passive elements, noise characteristics. |
| Week 4      | Ch. 9, analog MOSFET models.                          |
| Week 5      | Ch. 20, current sources and sinks.                    |
| Week 6      | Ch. 21, references.                                   |
| Week 7      | Ch. 22, amplifiers.                                   |
| Week 8      | Ch. 23, selected topics in feedback amplifier design. |
| Week 9      | Ch. 24, differential amplifiers.                      |
| Weeks 10-12 | Ch. 25, operational amplifiers.                       |
| Week 13     | Ch. 26, nonlinear analog circuits.                    |
| Week 14     | Ch. 27, dynamic analog circuits.                      |



Week 15 Chs. 28 & 29, selected topics in data converter design.

This text can also be used as an accompanying text in a VLSI systems course that focuses on the implementation of systems rather than circuits. Use of the text in this manner is benefited by inclusion of the LASI layout software.

## REFERENCES

- [1] *Revised Forecast for World Chip Market Shows Growth of 6.7% in 1996, 19% by 1999*, Semiconductor Forecast Summary 1995-1998, Semiconductor Industry Association.
- [2] "New TI Technology Doubles Transistor Density," *Texas Instruments Integration Newsletter*, Vol. 13, No. 5, July 1995.
- [3] P. Gray, "Possible Analog IC Scenarios for the 90's," <http://kabuki.eecs.berkeley.edu/slides.html>

### *Digital Circuits and VLSI System Design*

- [4] C. Mead and L. Conway, *Introduction to VLSI Systems*, Addison-Wesley, 1980.
- [5] Glasser and Dopferpuhl, *The Design and Analysis of VLSI Circuits*, Addison Wesley, 1985.
- [6] M. Annaratone, *Digital CMOS Circuit Design*, Kluwer, 1986.
- [7] A. Mukherjee, *Introduction to NMOS and CMOS VLSI Systems Design*, Prentice-Hall Publishers, 1986. ISBN 0-13-490947-X
- [8] D. A. Hodges and H. G. Jackson, *Analysis and Design of Digital Integrated Circuits*, McGraw-Hill, 2nd ed., 1988. ISBN 0 - 07 - 029158 - 6.
- [9] M. Shoji, *CMOS Digital Circuit Technology*, Prentice-Hall, 1988. ISBN 0-13-138850-9.
- [10] J. P. Uyemura, *Fundamentals of MOS Digital Integrated Circuits*, Addison-Wesley, 1988. ISBN 0-201-13318-0.
- [11] R. L. Geiger, P. E. Allen, and N. R. Strader, *VLSI - Design Techniques for Analog and Digital Circuits*, McGraw-Hill, 1990. ISBN 0-07-023253-9.
- [12] J. Y. Chen, *CMOS Devices and Technology for VLSI*, Prentice-Hall, 1990. ISBN 0-13-138082-6.
- [13] Fabricius, *Introduction to VLSI Design*, McGraw-Hill, 1990.
- [14] M. I. Elmasry, *Digital MOS Integrated Circuits II*, IEEE Press, 1992. ISBN 0-87942-275-0.
- [15] N.H.E. Weste and K. Eshraghian, *Principles of CMOS VLSI Design*, Addison Wesley, 2nd ed., 1993. ISBN 0-201-53376-6.

- [16] J. P. Uyemura, *Circuit Design for Digital CMOS VLSI*, Kluwer, 1992.
- [17] D. A. Pucknell and K. Eshraghian, *Basic VLSI Design*, 3rd ed., Prentice Hall Publishers, 1994. ISBN 0-13-079153-9
- [18] D. Pucknell, *Basic VLSI Design: Systems and Circuits*, 3rd ed., Prentice Hall, 1994. ISBN: 0-13-079153-9
- [19] W. Wolf, *Modern VLSI Design: A Systems Approach*, Prentice Hall, 1994. ISBN: 0-13-588377-6
- [20] S. Kang and Y. Leblebici, *CMOS Digital Integrated Circuits - Analysis and Design*, McGraw-Hill, 1996. ISBN 0-07-038046-5.
- [21] K. Gopalan, *Introduction to Digital Microelectronic Circuits*, Irwin, 1996. ISBN 0-256-12089-7.
- [22] J. M. Rabaey, *Digital Integrated Circuits - A Design Perspective*, Prentice Hall, 1996, ISBN 0-13-178609-1.

#### *Analog Circuits*

- [23] A. B. Grebene, *Bipolar and MOS Analog Integrated Circuit Design*, John-Wiley, 1984. ISBN 0-471-08529-4
- [24] R. Gregorian and G. C. Temes, *Analog MOS Integrated Circuits for Signal Processing*, John Wiley, 1986. ISBN 0-471-09797-7.
- [25] P. E. Allen and D. R. Holberg, *CMOS Analog Circuit Design*, Holt, Rinehart and Winston, 1987. ISBN 0-03-006587-9.
- [26] P. R. Gray, B. A. Wooley and R. W. Broderson, *Analog MOS Integrated Circuits II*, IEEE Press. ISBN 0-87942-246-7.
- [27] R. L. Geiger, P. E. Allen, and N. R. Strader, *VLSI - Design Techniques for Analog and Digital Circuits*, McGraw-Hill, 1990. ISBN 0-07-023253-9.
- [28] P. R. Gray and R. G. Meyer, *Analysis and Design of Analog Integrated Circuits*, 3rd ed., John Wiley, Inc., 1993. ISBN 0-471-57495-3.
- [29] M. Ismail and T. Fiez, *Analog VLSI - Signal and Information Processing*, McGraw-Hill, Inc. 1994. ISBN 0-07-032386-0.
- [30] K. R. Laker and W. Sansen, *Design of Analog Integrated Circuits and Systems*, McGraw-Hill, 1994. ISBN 0-07-036060-X.
- [31] G.A.S. Machado, *Low-Power HF Microelectronics a unified approach*, IEE, 1996. ISBN 0-85296-874-4.
- [32] D. Johns and K. Martin, *Analog Integrated Circuit Design*, John Wiley, 1997. ISBN 0-471-14448-7.

# Acknowledgments

We would like to thank the reviewers, contributors, and colleagues who helped make this book possible; Dr. Phil Allen, Savoula Amanatidis, Ben Ba, Jan Bissey, Dr. William Black, Jeff Bruce, Alan Buchholz, Dr. Joseph Cavallaro, Brian P. Lum Shue Chan, Irfan Chaudhry, Lisa Dayne, Dr. Ian Galton, Dr. Randall Geiger, John Griffin, Wes Hansford, Aaron Huntsinger, Dr. Bruce Johnson, David Kao, Dr. Joe Karniewicz, Brent Keeth, Dr. William Kuhn, Wen Li, Dr. H. Alan Mantooth, Dr. Richard Marks, Dean Moriarty, Dr. Ken Noren, Dr. Adrian Ong, Dr. James Rochelle, Dr. Terry Sculley, Dr. Stuart Tewksbury, Dr. Don Thelen, Dr. Axel Thomsen, Dr. Vance Tyree, Scott Ward, Veronica Wilson, Dr. Jeff Wu, and Dr. Kwang S. Yoon.

The authors would also like to thank Orbit Semiconductor, MOSIS and HP for allowing us to include their process information and design rules with the text.

*R. Jacob Baker*

*Harry W. Li*

*David E. Boyce*

---

# Chapter

# 2

---

## The Well

In order to develop a fundamental understanding of CMOS integrated circuit layout and design, we begin with a study of the n-well. This approach will build a solid foundation for understanding the performance limitations and parasitics (the pn junctions, capacitances, and resistances inherent in a CMOS circuit) of the CMOS process.

### 2.1 The Substrate

CMOS circuits are fabricated on and in a silicon wafer as was discussed in Ch. 1. This wafer is doped with donor atoms, such as phosphorus for an n-type wafer, or acceptor atoms, such as boron for a p-type wafer. Our discussion will center around a p-type wafer. When designing CMOS integrated circuits with a p-type wafer, n-channel MOSFETs (NMOS for short) are fabricated directly in the p-type wafer, while the p-channel transistors, PMOS, are fabricated in an "n-well." The substrate or well are sometimes referred to as the bulk or body of a MOSFET. CMOS processes that fabricate MOSFETs in the bulk are known as "bulk CMOS processes." The well and the substrate are illustrated in Fig. 2.1, though not to scale.

Often an epitaxial layer is grown on the wafer. We will not make a distinction between this layer and the substrate. Some processes use a p-well or both n- and p-wells (sometimes called twin tub processes). A process that uses a p-type substrate with an n-well is called an "n-well process." The process described in Appendix A, CN20, is an n-well process. The n-well acts as the substrate or body of the p-channel transistors.

Another important consideration is that the n-well and the p-substrate form a diode (Fig. 2.2). In CMOS circuits, the substrate is usually tied to the lowest voltage in the circuit to keep this diode from forward biasing. Ideally, zero current flows through the substrate connection.

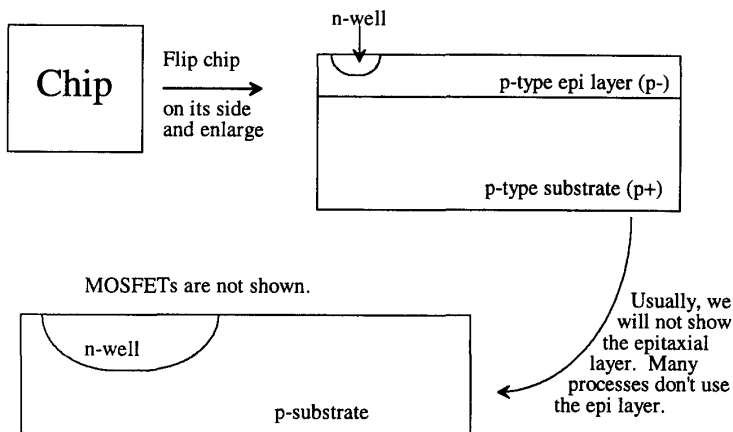


Figure 2.1 Illustration of the top and side view of a die.

Besides being used for fabricating p-channel transistors, the n-well can be used as a resistor. The voltage on either side of the resistor must be large enough to keep the substrate/well diode from forward biasing.

2.1.1 Patterning

CMOS integrated circuits are formed by patterning different layers on and in the CMOS wafer. Consider the following sequence of events that apply, in a fundamental way, to any layer we need to pattern. We start out with a clean, bare wafer as shown in Fig. 2.3a. The distance given by the line A to B will be used as a reference in Figs. 2.3b-j. Figures 2.3b-j are cross-sectional views of the dashed line shown in (a).

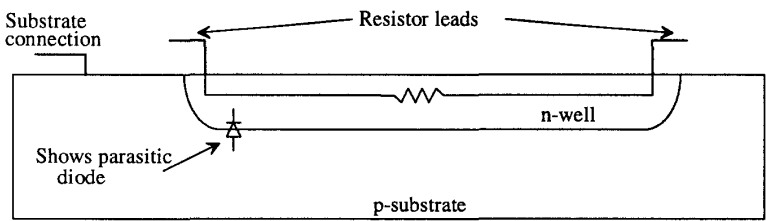
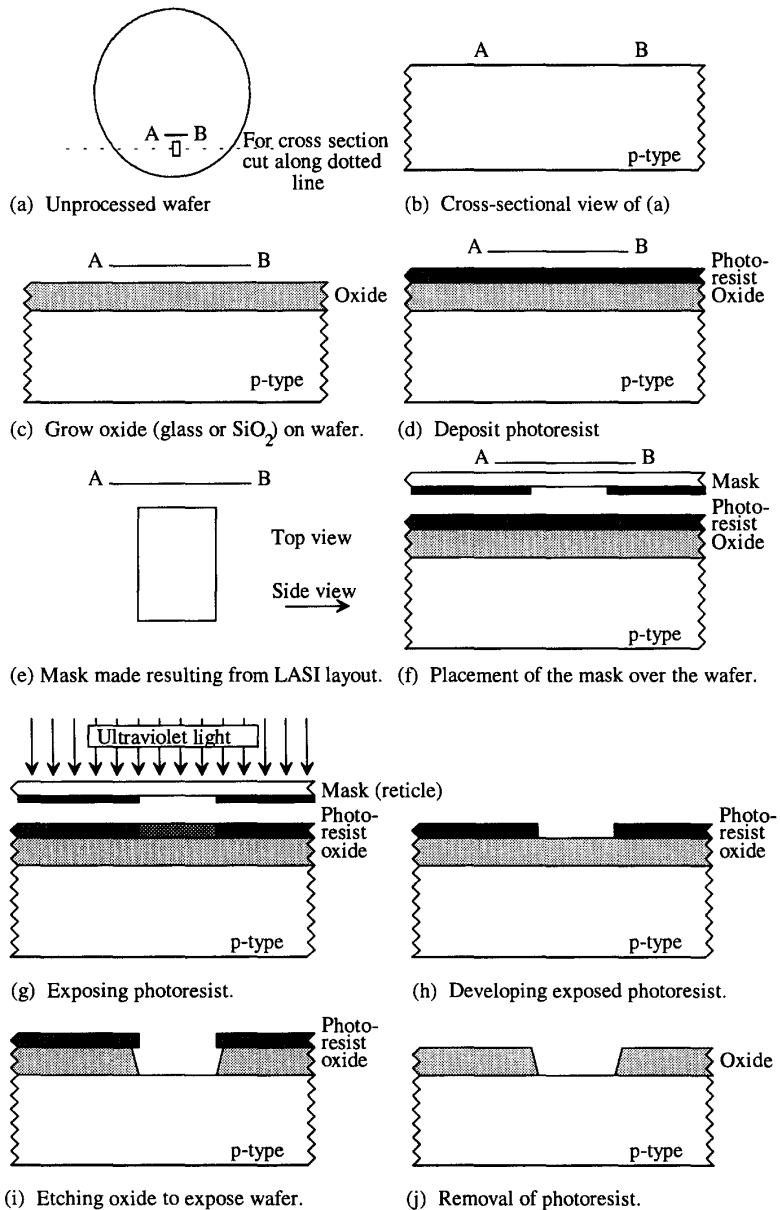


Figure 2.2 The n-well can be used as a resistor.

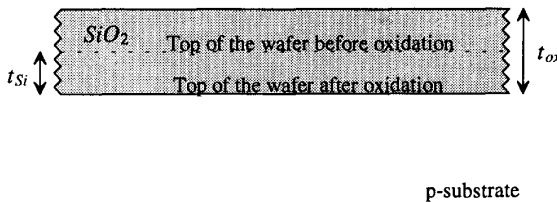


**Figure 2.3** Sequence of events used in patterning.

The first step in our generic patterning discussion is to grow an oxide,  $\text{SiO}_2$ , or glass, a very good insulator, on the wafer. Simply exposing the wafer to air yields the reaction  $\text{Si} + \text{O}_2 \rightarrow \text{SiO}_2$ . However, semiconductor processes must have tightly controlled conditions to precisely set the thickness and purity of the oxide. Therefore, we can grow the oxide using a reaction with steam,  $\text{H}_2\text{O}$ , or with  $\text{O}_2$  alone. The oxide resulting from the reaction with steam is called a wet oxide, while the reaction with  $\text{O}_2$  is a dry oxide. Both oxides are called thermal oxides due to the increased temperature used during oxide growth. The growth rate increases with temperature. The main benefit of the wet oxide is fast growing time. The main drawback of the wet oxide is the hydrogen byproduct. In general terms, the oxide grown using the wet techniques is not as pure as the dry oxide. The dry oxide, as we can conclude, generally takes a considerably longer time to grow. Both methods of growing oxide are found in CMOS processes.

An important observation we should make when looking at Fig. 2.3c is that the oxide growth actually consumes silicon. This is illustrated in Fig. 2.4. The overall thickness of the oxide is related to thickness of the consumed silicon by

$$t_{\text{Si}} = 0.45 \cdot t_{\text{ox}} \quad (2.1)$$



**Figure 2.4** How growing oxide consumes silicon.

The next step of the CMOS patterning process is to deposit a photosensitive resist layer across the wafer (see Fig. 2.3d). Keep in mind that the dimensions of the layers, that is, oxide, resist, and the wafer, are not drawn to scale. The thickness of a wafer is typically  $500\text{ }\mu\text{m}$ , while the thickness of a grown oxide or a deposited resist may be only a few  $\mu\text{m}$  or even less. After the resist is baked, the mask derived from the layout program, Figs. 2.3e and f, is used to selectively illuminate areas of the wafer, Fig. 2.3g. In practice, a single mask called a reticle, with openings several times larger than the final illuminated area on the wafer, is used to project the pattern and is stepped across the wafer with a machine called a stepper to generate the patterns needed to create multiple copies of a single chip. The light passing through the opening in the reticle is photographically reduced to illuminate the correct size area on the wafer.

The photoresist is developed (Fig. 2.3h), removing the areas that were illuminated. This process is called a positive resist process because the area that was illuminated was removed. A negative resist process removes the areas of resist that were not exposed to the light. Using both types of resist allows the process designer to cut down on the number of masks needed to define a CMOS process. Since creating the masks is expensive, lowering the number of masks is equated with lowering the cost of a process. This is also important in large manufacturing plants where fewer steps are equated with lower cost.

The next step in the patterning process is to remove the exposed oxide areas (Fig. 2.3i). Notice that the etchant etches under the resist, causing the opening in the oxide to be larger than what was specified by the mask. Some manufacturers intentionally bloat (make larger) or shrink (make smaller) the masks as specified by the layout program. Figure 2.3j shows the cross-sectional view of the opening after the resist has been removed.

### 2.1.2 Patterning the N-well

At this point we can make an n-well by diffusing donor atoms, those with 5 valence electrons, as compared to 4 for silicon, into the wafer. Referring to our generic patterning discussion given in Fig. 2.3, we begin by depositing a layer of resist directly on the wafer, Fig. 2.3d (without oxide). This is followed by exposing the resist to light through a mask (Figs. 2.3f and g) and developing or removing the resist (Fig. 2.3h). The mask used here can be generated with the LASI program. The next step in fabricating the n-well is to expose the wafer to donor atoms. The resist will block the diffusion of the atoms, while the openings will allow the donor atoms to penetrate into the wafer. This is shown in Fig. 2.5a. After a certain amount of time, dependent on the depth of the n-well desired, the diffusion source is removed (Fig. 2.5b). Notice that the n-well "outdiffuses" under the resist; that is, the final n-well size is not the same as the mask size. Again, the foundry where the chips are fabricated may bloat or shrink the mask to compensate for this *lateral diffusion*. The final step in making the n-well is the removal of the resist (Fig. 2.5c).

## 2.2 Laying out the N-well

When we lay out the n-well, we are viewing the chip from the top. The following example illustrates how to lay out an n-well of size 10  $\mu\text{m}$  square.

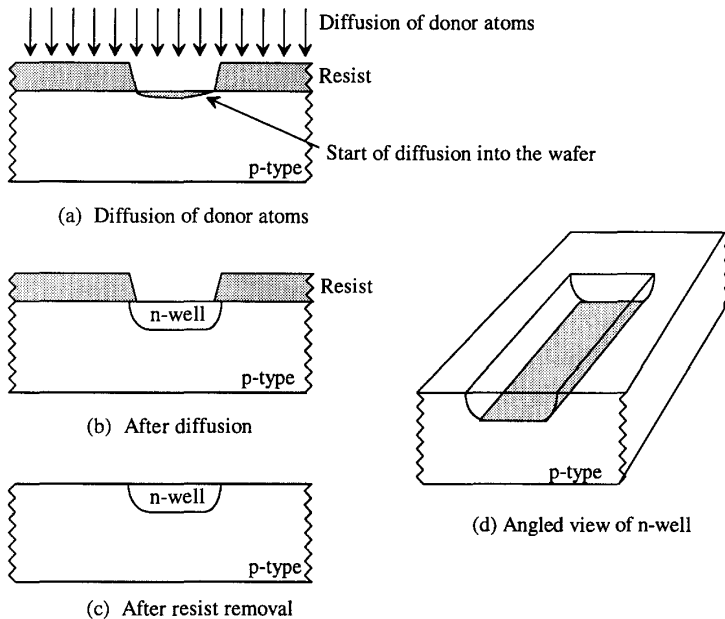
---

### Example 2.1

Using the LASI program, lay out an n-well that is 10  $\mu\text{m}$  square. Sketch the cross-sectional view of the layout. Assume we are using the CN20 setups given in the last chapter and using the layer table select **Layr** from the drawing screen commands and the layer "nwel." If the origin marker is not showing, type "r." Now select **Add**. The bottom of the display should show that the object is a box and that the grids are 1  $\mu\text{m}$ . Place the cursor in the display area on the origin



and press and release the LEFT mouse button. Move the mouse until the distance shows  $ux = 10\ \mu\text{m}$  and  $uy = 10\ \mu\text{m}$ . The resulting display is shown in the top of Fig. 2.6. Notice that the drawn size, both width and length, of the n-well differs from the actual size because of the lateral diffusion. ■



**Figure 2.5** Formation of the n-well.

### 2.2.1 Design Rules for the N-well

Now that we know how to lay out the n-well, we might ask the question, "Are there any limitations or constraints on the size and spacing of the n-wells?" That is to say, can we make the n-well  $2\ \mu\text{m}$  square? Can we make the distance between the n-wells  $1\ \mu\text{m}$ ? As we might expect, there are minimum spacing and size requirements for all layers in a CMOS process. Process engineers, who design the integrated circuit process, specify the design rules. A complete listing of the CN20 design rules can be found in Appendix A.

Figure 2.7 shows the design rules for the n-well. The minimum width of any n-well is  $3\ \mu\text{m}$ , while the minimum spacing between different n-wells is  $9\ \mu\text{m}$ . As the layout becomes complicated, the need for a program that ensures that the design rules are not violated is needed.

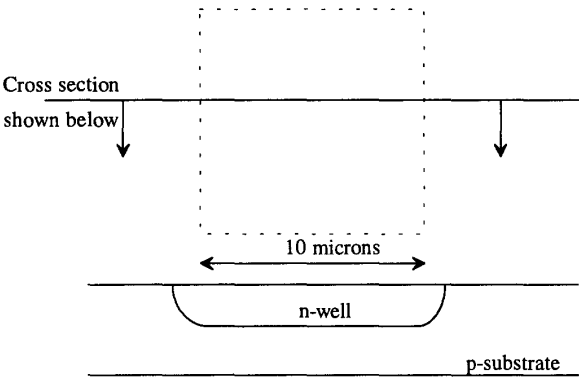


Figure 2.6 Layout and cross-sectional view of a 10 μm n-well.

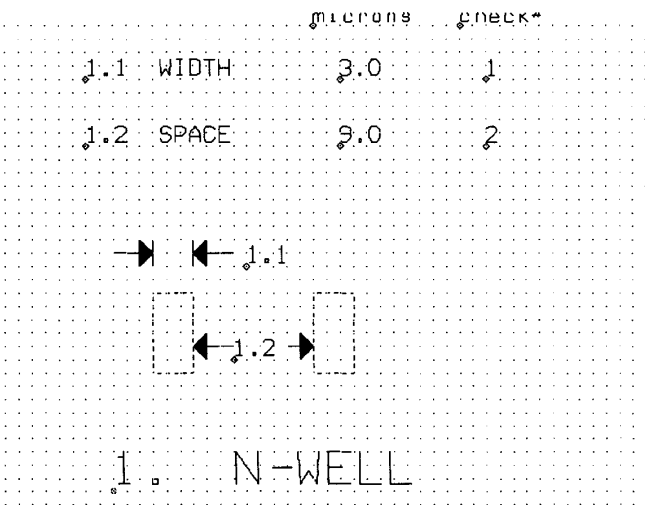


Figure 2.7 Design rules for layout of the n-well.

## 2.2.2 Using the LASIDRC Program

To perform a design rule check (DRC) of a layout, simply select the LASIDRC at the LASI system menu. Enter the name of the cell (see below) to be checked and type cn20.drc as the name of the check file. Since we only know two design rules at this point, that is, from Fig. 2.7 checks 1 and 2, the beginning check should be 1 and the ending check should be 2. To start the program, select Go on the top of the LASIDRC screen. If there is an error, a bit map of the n-well in blue will be displayed with the error in white. You can exit LASIDRC and go to LASI to fix the problem. When you return to LASIDRC, you can continue your checking at the check you failed.

### DRC Area

The design rule check will be performed on the *section* of layout shown in the drawing display just prior to calling the LASIDRC program. To DRC the entire cell, press the **Init** command button on the LASIDRC screen shown below. This feature can be used to decrease the time it takes to perform a DRC by DRCing only the specific areas of interest.

LASI Layer Logic Design Rule Checker Version 5.2 X

---

Help Read DOS Init Mem Go Quit

---

Checking Setup

Name of Cell to DRC **TEST2**

Name of Check File **CN20.DRC**

Starting Check Number **1**      Finishing Check Number **32**

Resolution=      Check Area Left      Distance=      Check Area Bottom

Scan Window Left **-48.55um**      Scan Window Right **118.05um**  
 Scan Window Bottom **-4.14um**      Scan Window Top **129.14um**

Enable HCPY Operator **N**      Enable PAUSE Operator **Y**  
 HCPY Command Line

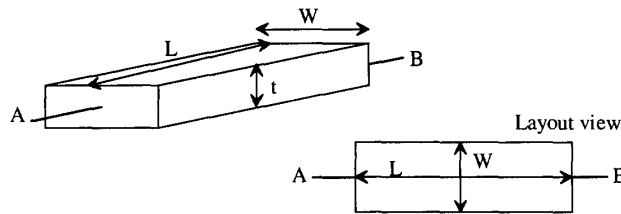
Name of Report File or Printer

↑ ↓ TAB=Move

## 2.3 Resistance Calculation

In addition to serving as a region in which to build PMOS transistors, n-wells are sometimes used to create integrated resistors. The resistance of a material is a function of the materials resistivity,  $\rho$ , and the materials dimensions. For example, the slab of material in Fig. 2.8 between the two leads has a resistance given by

$$R = \frac{\rho}{t} \cdot \frac{L}{W} \quad (2.2)$$



**Figure 2.8** Calculation of the resistance of a rectangular block of material.

In semiconductor processing, all of the fabricated thicknesses, such as the n-well, are constant. We only have control over  $W$  and  $L$ . Also notice that the  $W$  and  $L$  are what we see from the top view, that is, the layout view. We can rewrite Eq. (2.2) as

$$R = R_{\text{square}} \cdot \frac{L}{W} \quad (2.3)$$

$R_{\text{square}}$  is the sheet resistance of the material in  $\Omega/\text{square}$ .

### Example 2.2

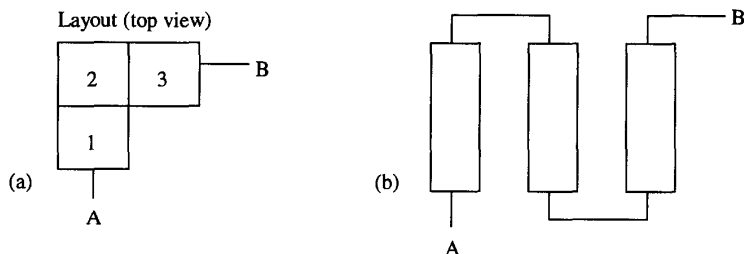
Calculate the resistance of an n-well that is  $10\ \mu\text{m}$  wide and  $100\ \mu\text{m}$  long.

From the Orbit electrical parameters in Appendix A for CN20 we see that the sheet resistance of the n-well varies from a minimum of  $2,000\ \Omega/\text{square}$  to a maximum of  $3,000\ \Omega/\text{square}$ , with a typical number of  $2,500\ \Omega/\text{square}$ . So the typical resistance between the ends of the n-well is

$$R = 2,500 \cdot \frac{100\ \mu\text{m}}{10\ \mu\text{m}} = 25\ \text{k}\Omega \quad \blacksquare$$

When laying out resistors using the "path" object, n-well, poly, or some other layer, LASI has a resistor calculator that will help in the calculation of nonrectangular resistances. Often, to minimize space, resistors are laid out in a serpentine pattern. The corners, that is, where the layer bends, are not rectangular. This is shown in Fig. 2.9a. All sections in Fig. 2.9a are square, so the resistance of sections 1 and 3 is  $R_{\text{square}}$ . The equivalent resistance of section 2 between the adjacent sides, however, is approximately  $0.6 R_{\text{square}}$ . The overall resistance between points A and B is therefore  $2.6 \cdot R_{\text{square}}$ .

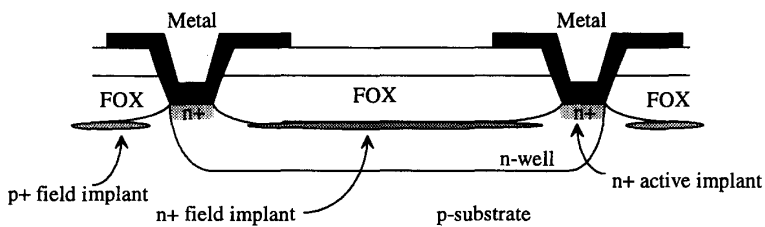
The layout shown in Fig. 2.9b uses wires to connect separate sections of a resistor avoiding corners. Avoiding corners in a resistor is the generally preferred method of layout in analog circuit design where the ratio of two resistors is important.



**Figure 2.9** (a) Figure for the calculation of the resistance of a corner section and (b) layout to avoid corners.

### 2.3.1 The N-well Resistor

At this point, it is appropriate to show the actual cross-sectional view of the n-well after all processing steps are completed (Fig. 2.10). The n+ and p+ implants are used to increase the threshold voltage of the field devices; more will be said on this later in Ch. 4. Notice in Appendix A, the Orbit electrical parameters, that the sheet resistance of the n-well is measured with the field implant in place, that is, with the n+ implant between the two metal connections in Fig. 2.10. Not shown in Fig. 2.10 is the connection to substrate. The field oxide (FOX; also known as ROX or recessed oxide) will be discussed in Ch. 4 when we discuss the active and poly layers.



**Figure 2.10** Cross-sectional view of n-well showing field implant. The field implantation is sometimes called the "channel stop implant".

### 2.4 The N-well/substrate Diode

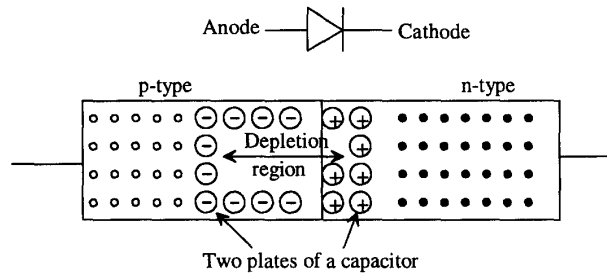
Placing an n-well in the p-substrate forms a diode. Therefore, it is important to understand how to model a diode for hand calculations and in SPICE simulations. In particular, let's discuss diodes using the n-well/substrate pn junction as an example [2]. The DC characteristics of the diode are given by the Shockley diode equation, or

$$I_D = I_S \left( e^{\frac{V_d}{nV_T}} - 1 \right) \quad (2.4)$$

The current  $I_D$  is the diode current;  $I_S$  is the scale (saturation) current;  $V_d$  is the voltage across the diode where the anode (p-type material) is assumed positive with respect to the cathode (n-type); and  $V_T$  is the thermal voltage which is given by  $\frac{kT}{q}$  where  $k$  = Boltzmann's constant ( $1.3806 \times 10^{-23}$  joules per degree kelvin),  $T$  is temperature in kelvin,  $n$  is the emission coefficient (a term that is related to the doping profile and affects the exponential behavior of the diode), and  $q$  is the electron charge of  $1.6022 \times 10^{-19}$  coulombs. The scale current and thus the overall diode current are related in SPICE by an area factor. The SPICE circuit simulation program assumes that the value of  $I_S$  supplied in the model statement was measured for a device with a reference area of 1. If an area factor of 2 is supplied for a diode, then  $I_S$  is doubled in Eq. (2.4).

### 2.4.1 Depletion Layer Capacitance

N-type silicon has a number of mobile electrons, while p-type silicon has a number of mobile holes (a vacancy of electrons in the valence band). Formation of a pn junction results in a depleted region at the p-n interface (Fig. 2.11). A depletion region is an area depleted of mobile holes or electrons. The mobile electrons move across the junction, leaving behind fixed donor atoms and thus a positive charge. The movement of holes across the junction, to the right in Fig. 2.11, occurs for the p-type semiconductor as well with a resulting negative charge. The fixed atoms on each side of the junction within the depleted region exert a force on the electrons or holes that have crossed the junction. This equalizes the charge distribution in the diode, preventing further charges from crossing the diode junction and also gives rise to a parasitic (depletion) capacitance.



**Figure 2.11** Simple illustration of depletion region formation in a pn junction.

The depletion capacitance,  $C_j$ , of a pn junction is given by

$$C_j = \frac{C_{j0}}{\left[ 1 - \left( \frac{V_d}{\Phi_0} \right) \right]^m} \quad (2.5)$$

$C_{j0}$  is the zero-bias capacitance of the pn junction, that is, the capacitance when the voltage across the diode is zero.  $V_d$  is the voltage across the diode,  $m$  is the grading coefficient (showing how the silicon changes from n- to p-type), and  $\phi_0$  is the built-in potential given by

$$\phi_0 = V_T \cdot \ln \left( \frac{N_A N_D}{n_i^2} \right) \quad (2.6)$$

where  $N_A$  and  $N_D$  are the dopings for the p- and n-type semiconductors, respectively,  $V_T$  is the thermal equivalent voltage  $\frac{kT}{q}$  (26 mV @ room temperature), and  $n_i$  is the intrinsic carrier concentration of silicon ( $n_i = 14.5 \times 10^9$  atoms/cm<sup>3</sup>).

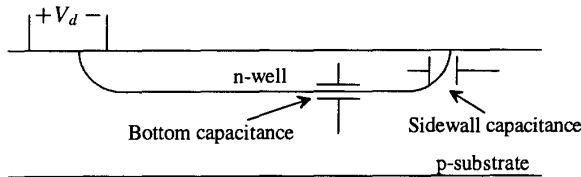
### Example 2.3

Sketch schematically the depletion capacitance of an n-well/p-substrate diode  $100 \times 100 \mu\text{m}^2$  square given that the substrate doping is  $10^{16}$  atoms/cm<sup>3</sup> and the well doping is  $10^{16}$  atoms/cm<sup>3</sup>. The measured zero-bias depletion capacitance of the junction is  $100 \text{ aF}/\mu\text{m}^2$ , and the grading coefficient is 0.333. Assume the depth of the n-well is  $3 \mu\text{m}$ .

We can begin this problem by calculating the built-in potential using Eq. (2.6):

$$\phi_0 = (.026) \cdot \ln \frac{10^{16} \cdot 10^{16}}{(14.5 \times 10^9)^2} = 0.7 \text{ V}$$

The depletion capacitance is made up of a *bottom* component and a *sidewall* component as shown in Fig. 2.12.



**Figure 2.12** A pn junction on the bottom and sides of the junction.

The bottom zero-bias depletion capacitance,  $C_{j0b}$ , is given by

$$C_{j0b} = (\text{Capacitance per Area}) \cdot (\text{Bottom Area}) \text{ which, for this example, is}$$

$$C_{j0b} = (100 \text{ aF}/\mu\text{m}^2) \cdot (100 \mu\text{m})^2 = 1 \text{ pF}$$

The sidewall zero-bias depletion capacitance,  $C_{j0s}$ , is given by

$$C_{j0s} = (\text{Capacitance per Area}) \cdot (\text{Depth of the Well}) \cdot (\text{Perimeter of the Well})$$

or

$$C_{j0s} = (100 \text{ aF}/\mu\text{m}^2) \cdot (3 \mu\text{m}) \cdot (400 \mu\text{m}) = 120 \text{ fF}$$

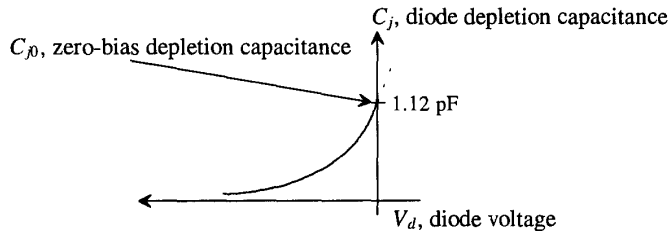
The total diode depletion capacitance between the n-well and the p-substrate is the parallel combination of the bottom and sidewall capacitances, or

$$C_j = \frac{C_{j0b}}{\left[1 - \left(\frac{V_d}{\Phi_0}\right)\right]^m} + \frac{C_{j0s}}{\left[1 - \left(\frac{V_d}{\Phi_0}\right)\right]^m} = \frac{C_{j0b} + C_{j0s}}{\left[1 - \left(\frac{V_d}{\Phi_0}\right)\right]^m}$$

Substituting in the numbers, we get

$$C_j = \frac{1 \text{ pF} + 0.120 \text{ pF}}{\left(1 - \left(\frac{V_d}{0.7}\right)\right)^{0.33}}$$

A sketch of how this capacitance changes with reverse potential is given in Fig. 2.13. Notice that when we discuss the depletion capacitance of a diode, it is usually with regard to a reverse bias. When the diode becomes forward-biased minority carriers, electrons in the p material and holes in the n material, injected across the junction, form a stored charge in and around the junction and give rise to a storage capacitance. This capacitance is usually much larger than the depletion capacitance. Furthermore, the time it takes to remove this stored charge can be significant. ■

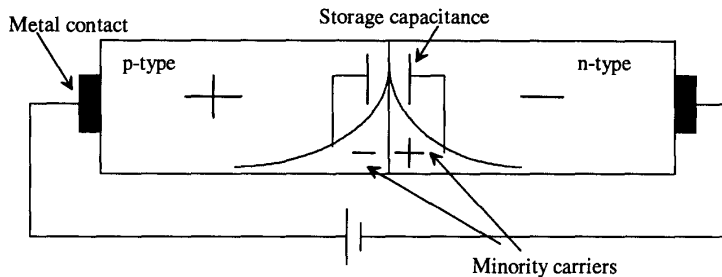


**Figure 2.13** Sketch of diode depletion capacitance against diode reverse voltage.

## 2.4.2 Storage Capacitance

Consider the charge distribution of a forward-biased diode shown in Fig. 2.14. When the diode becomes forward biased, electrons from the n-type side of the junction are attracted to the p-type side (and vice versa for the holes). After an electron drifts across the junction, it starts to diffuse toward the metal contact. If the electron recombines, that is, falls into a hole, before it hits the metal contact, the diode is called a "long base diode." The time it takes an electron to diffuse from the junction to the point the electron recombines is called the carrier lifetime. For silicon this lifetime is on the order of 10  $\mu\text{s}$ . If the distance between the junction and the metal contact is short, such that the electrons make it to the metal contact before recombining, the diode is said to be a "short base diode." In either case, the time between crossing the junction and





**Figure 2.14** Charge distribution in a forward-biased diode.

recombining is called the transit time,  $\tau_T$ . A capacitance is formed between the electrons diffusing into the p-side and the holes diffusing into the n-side, that is, formed between the minority carriers. (Electrons are the minority carriers in the p-type semiconductor.) This capacitance is often called a diffusion capacitance or storage capacitance (due to the presence of the stored minority carriers around the junction).

We can characterize the storage capacitance,  $C_S$ , in terms of the minority carrier lifetime. Under DC operating conditions, the storage capacitance is given by

$$C_S = \frac{I_D}{nV_T} \cdot \tau_T \quad (2.7)$$

$I_D$  is the DC current flowing through the forward-biased junction given by Eq. (2.4). Looking at the diode capacitance in this way is very useful for analog AC small-signal analysis. However, for digital applications we are more interested in the large-signal switching behavior of the diode. It should be pointed out that in general, for a CMOS process, it is undesirable to have a forward-biased pn junction. If we do have a forward-biased junction, it usually means there is a problem, for example, electrostatic protection, capacitive feedthrough possibly causing latch-up, and so on. These topics are discussed in more detail later in this chapter.

Consider Fig. 2.15. In the following diode switching analysis, we will assume that  $V_F \gg 0.7$ ,  $V_R < 0$  and that the voltage source has been at  $V_F$  long enough to reach steady-state condition; that is, the minority carriers have diffused out to an equilibrium condition. At the time  $t_1$  the input voltage source makes an abrupt transition from  $V_F$  to  $V_R$ , causing the current to change from  $\frac{V_F}{R}$  to  $\frac{V_R}{R}$ . The diode voltage remains at 0.7 V since the diode contains a stored charge that must be removed. At time  $t_2$  the stored charge is removed. At this point, the diode basically looks like a voltage-dependent capacitor that follows Eq. (2.5). In other words for  $t > t_2$  the diode depletion capacitance is charged through  $R$  until the current in the circuit goes to zero and the voltage across the diode is  $V_R$ . This accounts for the exponential decay of the current and voltage shown in Fig. 2.15.

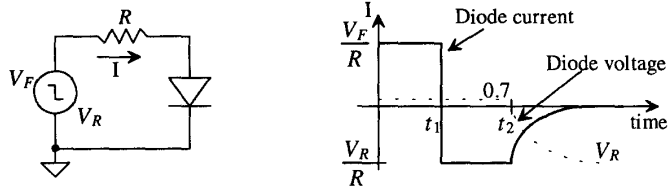


Figure 2.15 Diode test circuit.

The diode storage time, the time it takes to remove the stored charge,  $t_s$ , is simply the difference in  $t_2$  and  $t_1$ , or

$$t_s = t_2 - t_1 \quad (2.8)$$

This time is also given by

$$t_s = \tau_T \cdot \ln \frac{i_F - i_R}{-i_R} \quad (2.9)$$

where  $\frac{V_F}{R} = i_F$  and  $\frac{V_R}{R} = i_R$  is a negative number in this discussion. Note that it is quite easy to determine the minority carrier lifetime using this test setup.

Defining a time  $t_3$ , where  $t_3 > t_2$ , when the current in the diode becomes 10 percent of  $\frac{V_R}{R}$ , we can define the diode reverse recovery time, or

$$t_{rr} = t_3 - t_1 \quad (2.10)$$

### 2.4.3 SPICE Modeling

The SPICE (simulation program with IC emphasis) diode model parameters are listed in Table 2.1. The series resistance,  $R_s$ , deserves some additional comment. This resistance results from the finite resistance of the semiconductor used in making the diode and the contact resistance, the resistance resulting from a metal contact to the semiconductor. At this point, we are only concerned with the resistance of the semiconductor. For a reverse-biased diode, the depletion layer width changes, increasing for larger reverse voltages (decreasing both the capacitance and series resistance, of the diode). However, when we model the series resistance, we use a constant value. In other words, SPICE will not show us the effects of a varying  $R_s$ .

#### Example 2.4

Using SPICE, explain what happens when a diode with a carrier lifetime of 30 ns is taken from the forward-biased region to the reverse-biased region. Use the circuit shown in Fig. Ex2.4 to illustrate your understanding.

We will assume a zero-bias depletion capacitance of 1 pF. The SPICE netlist for this circuit is shown below.

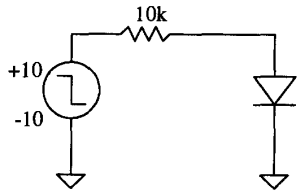


Figure Ex2.4

```
Diode storage time.
D1      1 0 TRR
R1      3 1 10k
Vin     3 0      DC 0 PULSE(10 -10 50n .1n .1n 50n 100n)
.MODEL  TRR  D
+ IS=1.0E-15 TT=30E-9 CJO=1E-12 VJ=.7 M=0.33
.probe
.tran 1n 100n
.end
```

Figure 2.16 shows the current through the diode (scaled), the input voltage step, and the voltage across the diode. One of the interesting things to notice about this circuit is that current actually flows through the diode in the negative direction, even though the diode is forward biased. During this time, the stored minority carrier charge is removed from the junction. The storage time is given by

$$t_s = 30 \text{ ns} \cdot \ln \frac{0.93 \text{ mA} + 1.07 \text{ mA}}{1.07 \text{ mA}} = 18.8 \text{ ns}$$

which is close to the simulation results. Note that the input pulse doesn't change until 50 ns after the simulation starts. This ensures a steady-state condition when the input changes from 10 to -10 V. ■

| Name     | SPICE |                                     |
|----------|-------|-------------------------------------|
| $I_s$    | IS    | Saturation current                  |
| $R_s$    | RS    | Series resistance                   |
| $n$      | N     | Emission coefficient                |
| $V_{bd}$ | BV    | Breakdown voltage                   |
| $I_{bd}$ | IBV   | Current which flows during $V_{bd}$ |
| $C_{j0}$ | CJ0   | Zero-bias pn junction capacitance   |
| $\phi_0$ | VJ    | Built-in potential                  |
| $m$      | M     | Grading coefficient                 |
| $\tau_T$ | TT    | Carrier transit time                |

Table 2.1 SPICE parameters related to diode.

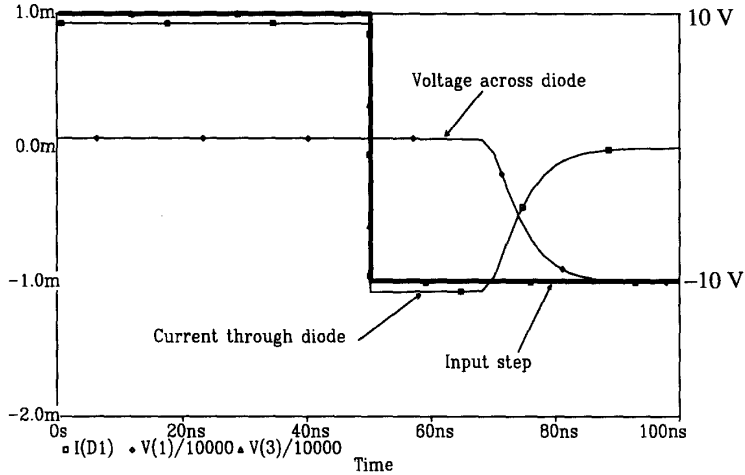


Figure 2.16 Results of Example 2.4 showing current and scaled voltages.

## 2.5 The RC Delay Through an N-well

At this point, we know that the n-well can be used as a resistor and as a diode when used with the substrate. Figure 2.17a shows the parasitic capacitance and resistance associated with the n-well. Since there is a depletion capacitance from the n-well to the substrate, we could sketch the equivalent symbol for the n-well resistor as shown in Fig. 2.17b. This is the basic form of an RC transmission line. If we put a step into one side of the n-well resistor a finite time later, called the delay time and measured at the 50 percent points of the pulses, the pulse will appear.

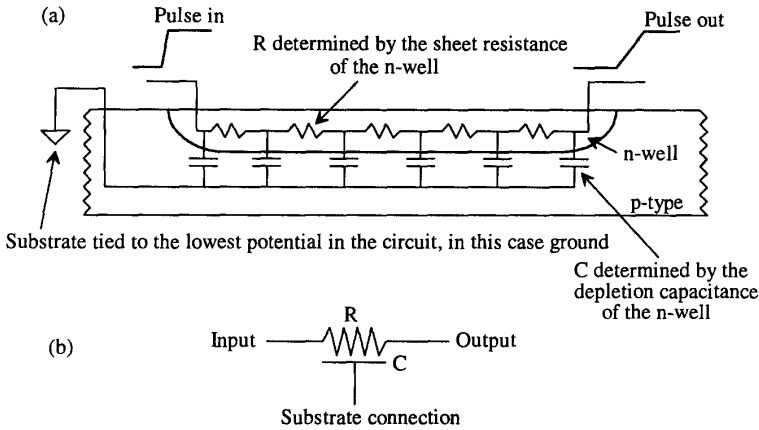
The delay can be calculated by knowing the resistance,  $r$ , per unit length, the capacitance,  $c$ , per unit length, and  $l$ , the number of unit lengths using the following [3]:

$$t_d = 0.35rc l^2 \quad (2.11)$$

### Example 2.5

Estimate the delay through a 250 k $\Omega$  resistor made using an n-well with a width of 3  $\mu\text{m}$  and a length of 300  $\mu\text{m}$ . Verify your answer with SPICE.

If we divide the resistor up into 100 squares each 3  $\mu\text{m}$  wide and 3  $\mu\text{m}$  long, we can define the number of unit lengths,  $l$ , as 100. The resistance of one of these squares is 2.5 k $\Omega$  ( $= r$ ) per unit length. Now we are faced with determining the capacitance to substrate of one of these squares. Since the capacitance is a function of the voltage, we can get a worst-case estimate by selecting the zero-bias depletion capacitance (see Fig. 2.13). From Ex. 2.3 the zero-bias



**Figure 2.17** (a) Parasitic resistance and capacitance of the n-well and (b) schematic symbol.

depletion capacitance between the n-well and substrate is  $100 \text{ aF}/\mu\text{m}^2$ . The capacitance per unit length is the sum of the bottom and sidewall capacitances. However, since each square, except for the first and last squares, has only two sides contributing to the depletion capacitance and it is desirable to keep the number of calculations to a minimum in an estimate, we will neglect the sidewall capacitance. The capacitance per unit length is given by

$$c = C_{\text{job}} = 100 \frac{\text{aF}}{\mu\text{m}^2} (3 \cdot 3) \mu\text{m}^2 = 900 \text{ aF}$$

The delay is now estimated by

$$t_d = 0.35 \cdot rcl^2 = 0.35 \cdot 2.5\text{k} \cdot 900 \text{ aF} \cdot 100^2 = 7.88 \text{ ns}$$

The SPICE netlist and the resulting output are shown in Fig. 2.18. Note that this is a SPICE3 netlist and not a PSPICE netlist. Also note that the nodes were labeled with names (i.e., Vin and Vout) rather than numbers. ■

We can simplify Eq. (2.11) by realizing that the products  $r \cdot l$  and  $c \cdot l$  are the total resistance and capacitance to substrate of the n-well resistor. Using this result on the previous example gives  $R = r \cdot l = 2,500 \cdot 100 = 250 \text{ k}\Omega$  and  $C = c \cdot l = 900 \text{ aF} \cdot 100 = 90 \text{ fF}$  and therefore

$$t_d = 0.35 \cdot RC = 0.35 \cdot 250\text{k} \cdot 90 \text{ fF} = 7.88 \text{ ns} \quad (2.12)$$

which is the same result given the example. The important thing to notice here is that we can totally avoid the unit length parameter  $l$ . The resistance  $R$  is the resistance of

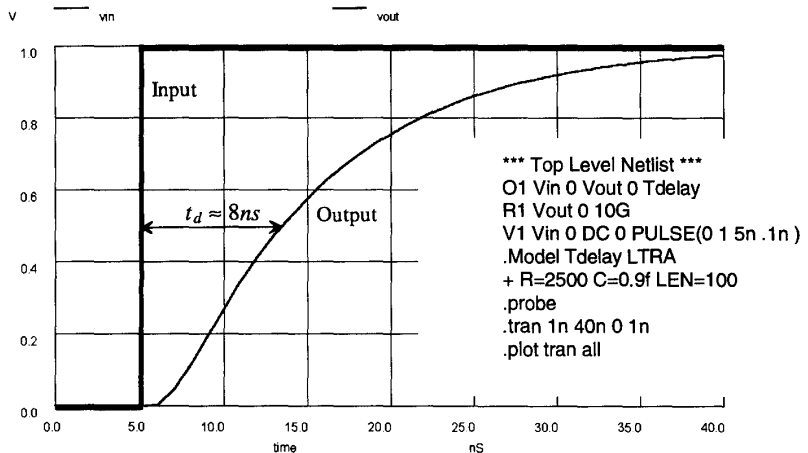


Figure 2.18 Simulation results from Example 2.5.

the resistor ( $= 250 \text{ k}\Omega$  above), and the capacitance  $C$  is simply the product of the bottom area of the resistor with the zero-bias depletion capacitance ( $= 3 \cdot 300 \cdot 100 \text{ aF} = 90 \text{ fF}$ ).

## REFERENCES

- [1] D. E. Boyce, *LASI Users Manual*, available by pressing F1 while running LASI or by running the MANUAL.EXE program included with LASI.
- [2] D. A. Hodges and H. G. Jackson, *Analysis and Design of Digital Integrated Circuits*, McGraw-Hill Publishing Company, 2nd ed., 1988. ISBN 0-07-029158-6.
- [3] N. H. E. Weste and K. Eshraghian, *Principles of CMOS VLSI Design*, Addison-Wesley, 2nd ed., 1993. ISBN 0-201-53376-6.

## PROBLEMS

- 2.1 Figure P2.1 is a section of n-well laid out using the path object with a width of  $4 \mu\text{m}$ . Sketch the cross-sectional view (see Ex. 2.1, at the positions indicated in the figure). Copy this layout using the LASI program. Using the **Res** command (before using the **Res** command, use **fget** on the path) with a sheet resistance of  $2,500 \Omega/\text{square}$  (the typical sheet resistance of the n-well) and an end correction of 0.6, determine the resistance of the section. How does this compare with the  $2.6R_{\text{square}}$  used for the layout in Fig. 2.9a?

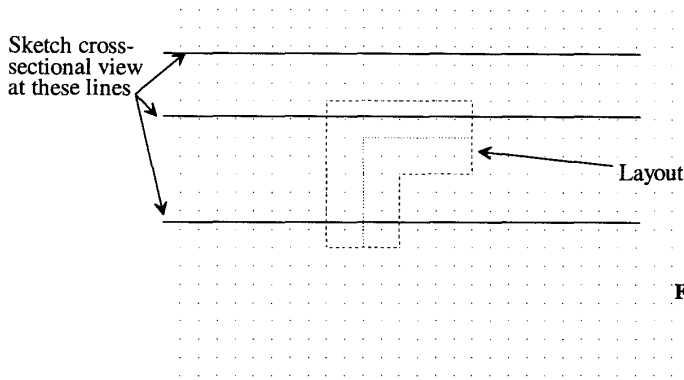


Figure P2.1

- 2.2 Add two boxes to the layout of Problem 1 (see Fig. P2.2). Using the LASIDRC program with checks 1 and 2, show the design rule violations in the layout.
- 2.3 Lay out a nominally 250 k $\Omega$  resistor using the n-well and the serpentine pattern shown in Fig. P2.3. Assume that the maximum length of a segment is 100  $\mu\text{m}$ . Also design rule check the finished resistor.
- 2.4 Assuming the n-well depth is 3  $\mu\text{m}$  in the CN20 process, what are the minimum, typical, and maximum values of the n-well resistivity?
- 2.5 Normally, the scale current of a pn junction is specified in terms of a scale current density,  $J_s$  (A/m<sup>2</sup>) and the width and length of a junction (i.e.,  $I_s = J_s \cdot L \cdot W$  neglecting the sidewall component). Estimate the scale current for the diode of Ex. 2.3 if  $J_s = 10^{-8}$  A/m<sup>2</sup>.
- 2.6 Repeat Problem 5 including the sidewall component ( $I_s = J_s \cdot L \cdot W + J_s \cdot (2L + 2W) \cdot \text{depth}$ ).
- 2.7 Using the diode of Ex. 2.3 in the circuit of Fig. P2.7, estimate the frequency of the input signal when the AC component of  $v_{out}$  is 707  $\mu\text{V}$  (i.e., estimate the 3 dB frequency of the  $|v_{out}/v_{in}|$ ).
- 2.8 Verify the answer given in problem 7 with SPICE.
- 2.9 Using SPICE, show that a diode can conduct current from its cathode to its anode when the diode is forward biased.
- 2.10 Estimate the delay through a 1 M $\Omega$  resistor (5  $\mu\text{m}$  by 2,000  $\mu\text{m}$ ) made in the CN20 process using the n-well. Verify with SPICE.
- 2.11 If one end of the resistor in Problem 10 is tied to +5 V and the other end is tied to the substrate that is tied to ground, estimate the depletion capacitance (F/m<sup>2</sup>) between the n-well and the substrate in the middle of the resistor. Assume that the resistance does not vary with position along the resistor.





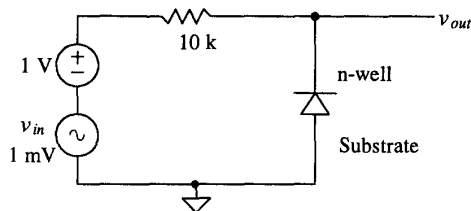


Figure P2.7

- 2.12** The diode reverse breakdown current, that is, the current that flows when  $|V_d| < BV$  (breakdown voltage), is modeled in SPICE by

$$I_D = IBV \cdot e^{-(V_d + BV)/V_T}$$

Assuming  $10 \mu\text{A}$  of current flows when the junction starts to break down at  $100 \text{ V}$ , simulate, using a SPICE DC sweep, the reverse breakdown characteristics of the diode. (The breakdown voltage,  $BV$ , is a positive number. When the diode starts to break down  $-BV = V_d$ . For this diode, breakdown occurs when  $V_d = -100 \text{ V}$ .)

- 2.13** Repeat Ex. 2.3 if the n-well/p-substrate diode is  $50 \mu\text{m}$  square and the acceptor doping concentration is changed to  $10^{15} \text{ atoms/cm}^3$ .
- 2.14** Estimate the storage time, that is, the time it takes to remove the stored charge in a diode (see Fig. 2.15), when  $\tau_T = 5 \text{ ns}$ ,  $V_F = 5 \text{ V}$ ,  $V_R = -5 \text{ V}$ ,  $C_{j0} = 0.5 \text{ pF}$ , and  $R = 1 \text{ k}$ . Verify your results with SPICE.

## Data Converter Architectures

Applications such as wireless communications and digital audio and video have created the need for cost-effective data converters that will achieve higher speed and resolution. The needs required by digital signal processors continually challenge analog designers to improve and develop new ADC and DAC architectures. There are many different types of architectures, each with unique characteristics and different limitations. This chapter presents a basic overview of the more popular data converter architectures and discusses the advantages and disadvantages of each along with their limitations.<sup>1</sup>

Now that we have defined the operating characteristics of ADCs, a more detailed examination of the basic architectures will be discussed using a top-down approach. Since many of the converters use op-amps, comparators, and resistor and capacitor arrays, the top-down approach will allow a broader discussion of the key component limitations in later sections.

### 29.1 DAC Architectures

A wide variety of DAC architectures exist, ranging from very simple to complex. Each, of course, has its own merits. Some use voltage division, whereas others employ current steering and even charge scaling to map the digital value into an analog quantity.

#### 29.1.1 Digital Input Code

In many cases, the digital signal is not provided in binary code but is any one of a number of codes: binary, BCD, thermometer code, Gray code, sign-magnitude, two's complement, offset binary, and so on. (See Fig. 29.1 for a comparison of some of the more commonly used digital input codes.) For example, it may be desirable to allow only one bit to change value when changing from one code to the next. If that is the case, a Gray code will suffice. The thermometer code is used quite frequently and can

---

<sup>1</sup> Much of the material in this chapter is based on course notes provided by Dr. Terry Sculley, of Crystal Semiconductor Corporation, Austin, Texas.

| Decimal | Binary | Thermometer | Gray | Two's Complement |
|---------|--------|-------------|------|------------------|
| 0       | 000    | 0000000     | 000  | 000              |
| 1       | 001    | 0000001     | 001  | 111              |
| 2       | 010    | 0000011     | 011  | 110              |
| 3       | 011    | 0000111     | 010  | 101              |
| 4       | 100    | 0001111     | 110  | 100              |
| 5       | 101    | 0011111     | 111  | 011              |
| 6       | 110    | 0111111     | 101  | 010              |
| 7       | 111    | 1111111     | 100  | 001              |

**Figure 29.1** Comparison of digital input codes.

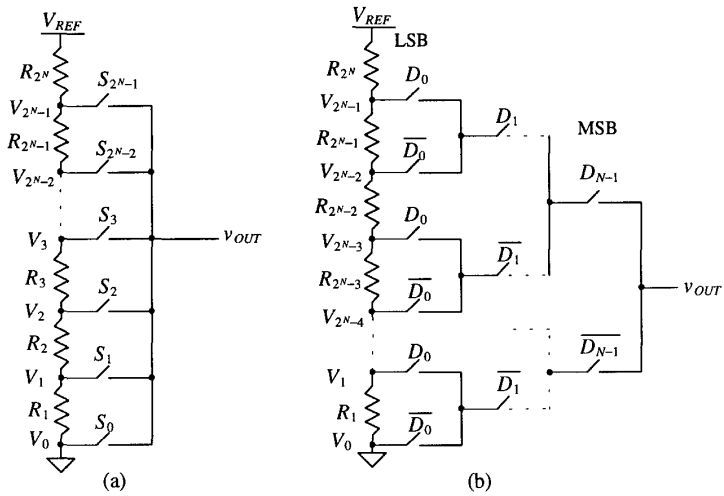
also be seen in Fig. 29.1. Notice that it requires  $2^N - 1$  bits to represent an  $N$ -bit word. The code used is purely dependent on the application, and the reader should be aware that many types of codes are available.

**29.1.2 Resistor String**

The most basic DAC is seen in Fig. 29.2a. Comprised of a simple resistor string of  $2^N$  identical resistors and switches, the analog output is simply the voltage division of the resistors at the selected tap. Note that a  $N:2^N$  decoder will be required to provide the  $2^N$  signals controlling the switches. This architecture typically results in good accuracy, provided that no output current is required and that the values of the resistors are within the specified error tolerance of the converter. One big advantage of a resistor string is that the output will always be guaranteed to be monotonic.

One problem with this converter is that the converter output is always connected to  $2^N - 1$  switches that are off and one switch that is on. For larger resolutions, a large parasitic capacitance appears at the output node, resulting in slower conversion speeds. A better alternative for the resistor string DAC is seen in Fig. 29.2b. Here, a binary switch array ensures that the output is connected to at most  $N$  switches that are on and  $N$  switches that are off, thus increasing the conversion speed. The input to this switch array is a binary word since the decoding is inherent in the binary tree arrangement of the switches.

Another problem with the resistor string DAC is the balance between area and power dissipation. An integrated version of this converter will lead to a large chip area for higher bit resolutions because of the large number of passive components needed. Active resistors such as the  $N$ -well resistor can be used for low-resolution applications. However, as the resolution increases, the relative accuracy of the resistors becomes an important factor. Although the value of  $R$  could always be made small to minimize the



**Figure 29.2** (a) A simple resistor string DAC and (b) use of a binary switch array to lower the output capacitance.

chip area required, power dissipation would then become the critical issue as current flows through the resistor string at all times.

### Example 29.1

Design a 3-bit resistor string ladder using a binary switch array. Assume that  $V_{REF} = 5$  V and that the maximum power dissipation of the converter is to be 5 mW (not including the power required by the digital logic). Determine the value of the analog voltage for each of the possible digital input codes.

The power dissipation will determine the current flowing through the resistor string by

$$I_{MAX} = \frac{5 \times 10^{-3} \text{ W}}{5 \text{ V}} = 1 \text{ mA}$$

Since a 3-bit converter will have eight resistors, the value of  $R$  is

$$R = \frac{1}{8} \cdot \frac{5 \text{ V}}{1 \text{ mA}} = 625 \Omega$$

The converter can be seen in Fig. 29.3. Examine the switch array if the input code is  $D_2D_1D_0 = 100$  or  $4_{10}$ . Since  $D_2$  is high, the top switch will be closed and the lower switch,  $\overline{D_2}$ , will be open. In the row corresponding to  $D_1$ , since  $D_1 = 0$ , both of the switches marked  $\overline{D_1}$  will be closed and the other two will be open. The LSB controls the largest number of switches; therefore, since  $D_0$  is low, all

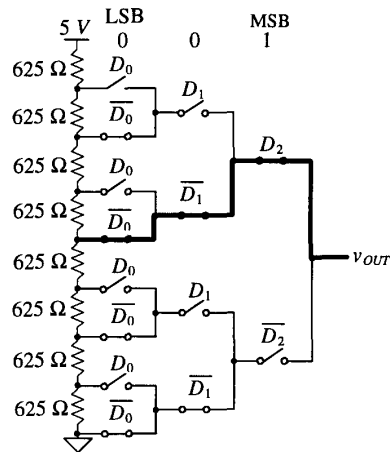


Figure 29.3 A 3-bit resistor string DAC used in Ex. 29.1

the  $\overline{D_0}$  switches will be closed and all the  $D_0$  switches will be open. There should be only one path connecting a single tap on the resistor string to the output. This is bolded, with the resistor string tapped in the middle of the string. Therefore,  $v_{OUT} = \frac{1}{2} V_{REF} = 2.5 \text{ V}$ . The remaining outputs can be seen in Fig. 29.4. ■

| $D_2D_1D_0$ | $v_{OUT}$ |
|-------------|-----------|
| 000         | 0         |
| 001         | 0.625     |
| 010         | 1.25      |
| 011         | 1.875     |
| 100         | 2.5       |
| 101         | 3.125     |
| 110         | 3.75      |
| 111         | 4.375     |

Figure 29.4 Output voltages generated from the 3-bit DAC in Ex. 29.1.

### Mismatch Errors Related to the Resistor String DAC

The accuracy of the resistor string is obviously related to matching between the resistors, which ultimately determines the INL and DNL for the entire DAC. Suppose that the  $i$ -th resistor,  $R_i$ , has a mismatch error associated with it so that

$$R_i = R + \Delta R_i \quad (29.1)$$

where  $R$  is the ideal value of the resistor and  $\Delta R_i$  is the mismatch error. Also suppose that the mismatches were symmetrical about the string so that the sum of all the mismatch terms were zero, or

$$\sum_{i=1}^{2^N} \Delta R_i = 0 \quad (29.2)$$

The value of the voltage at the tap associated with the  $i$ -th resistor should ideally be

$$V_{i,ideal} = \frac{(i)V_{REF}}{2^N}, \text{ for } i = 0, 1, 2, \dots, 2^N - 1 \quad (29.3)$$

However, including the mismatch, the actual value of the  $i$ -th voltage will be the sum of all the resistances up to and including resistor  $i$ , divided by the sum of all the resistances in the string. This can be represented by

$$V_i = V_{REF} \cdot \frac{\sum_{k=1}^i R_k}{\sum_{k=1}^{2^N} R_k} = V_{REF} \cdot \frac{\sum_{k=1}^i R + \Delta R_k}{2^N R} \quad (29.4)$$

The denominator does not include any mismatch error since it was assumed that the mismatches sum to zero as defined in Eq. (29.2). Notice that there is no resistor,  $R_0$ , corresponding to  $V_0$  (see Fig. 29.2), and it is assumed that  $V_0$  is ground. Equation (29.4) can be rewritten as

$$V_i = \frac{V_{REF}}{2^N R} \left[ (i)R + \sum_{k=1}^i \Delta R_k \right] = \frac{(i)V_{REF}}{2^N} + \frac{V_{REF}}{2^N R} \sum_{k=1}^i \Delta R_k \quad (29.5)$$

or finally, the value of the voltage at the  $i$ -th tap is

$$V_i = V_{i,ideal} + \frac{V_{REF}}{2^N} \cdot \sum_{k=1}^i \frac{\Delta R_k}{R} \quad (29.6)$$

Equation (29.6) is not of much importance by itself, but it can be used to help determine the nonlinearity errors.

### Integral Nonlinearity of the Resistor String DAC

Integral nonlinearity (INL) is defined as the difference between the actual and ideal switching points, or

$$INL = V_i - V_{i,ideal} \quad (29.7)$$

and plugging in Eqs. (29.6) and (29.3) into (29.7) yields,

$$INL = \frac{V_{REF}}{2^N} \cdot \sum_{k=1}^i \frac{\Delta R_k}{R} \quad (29.8)$$

Equation (29.8) is a general expression for the INL for a given resistor,  $R_i$ , and requires that the mismatch of all the resistances used in the summation are known. However, this equation does not illuminate how to determine the *worst-case* or maximum INL for a resistor string.

Intuitively, one would think that the worst-case INL would occur at the top of the resistor string ( $i=2^N$ ) with all the  $\Delta R_k$ 's at their maximum values. However, the previous derivation was performed with the assumption that the mismatches summed to zero. With this restriction, the maximum INL will occur at the midpoint of the string corresponding to  $i = 2^{N-1}$ , corresponding to the case where the MSB was a one and all other bits were zero. Another condition that will ensure a worst-case scenario will be to consider the lower half resistors at their maximum positive mismatch value and upper half resistors at their maximum negative mismatch value, or vice versa.

If the resistors on a string were known to have 2 percent matching, then  $\Delta R_k$  would be constrained to the bounds of

$$-0.02R \leq \Delta R_k \leq 0.02R \quad (29.9)$$

and the worst-case INL (again, assuming 2 percent matching) using Eq. (29.8) would be,

$$|INL|_{max} = \frac{V_{REF}}{2^N} \cdot \sum_{k=1}^{2^{N-1}} \frac{\Delta R_k}{R} = \frac{V_{REF}}{2^N} \cdot \frac{2^{N-1} \cdot 0.02R}{R} = 0.01 \cdot V_{REF} \quad (29.10)$$

Note that Eq. (29.10) has units of volts, which can easily be converted to LSBs. We can use Eq. (29.10) to determine the effective resolution of a DAC using a resistor string, as will be seen in the next example.

Because the worst-case analysis was performed, the maximum INL occurs at the middle of the string. We can improve this specification on paper by using the "best-fit" approach to measuring INL. In this case, the reference line is simply shifted up slightly (refer to Ch. 28) so that it no longer passes through the end points, but instead minimizes the INL.

---

### Example 29.2

Determine the effective number of bits for a resistor string DAC, which is assumed to be limited by the INL. The resistors are passive poly resistors with a known relative matching of 1 percent and  $V_{REF} = 5$  V.

Using Eq. (29.10), the maximum INL will be

$$|INL|_{max} = 0.005 \cdot V_{REF} = 0.025 \text{ V}$$

Since we know that this maximum INL should be equal to  $\frac{1}{2}$  LSB in the worst case,

$$\frac{1}{2}LSB = \frac{5}{2^{N+1}} = 0.025 \text{ V}$$

and solving for  $N$  yields

$$N = \log_2\left(\frac{5}{0.025}\right) - 1 = 6.64 \text{ bits}$$

This means that the resolution for a DAC containing a resistor string matched to within 1 percent will be, at most 6 bits. ■

#### *Differential Nonlinearity of the Worst-Case Resistor String DAC*

Resistor string matching is not as critical when determining the DNL. Remembering that the definition of DNL is simply the actual height of the stair-step in the DAC transfer curve minus the ideal step height, we can write this in terms of the voltages at the taps of adjacent resistors on the string. Using Eq. (29.5), we can express this as,

$$|V_i - V_{i-1}| = \left| \left[ \frac{(i)V_{REF}}{2^N} + \frac{V_{REF}}{2^N} \cdot \sum_{k=1}^i \frac{\Delta R_k}{R} \right] - \left[ \frac{(i-1)V_{REF}}{2^N} + \frac{V_{REF}}{2^N} \cdot \sum_{k=1}^{i-1} \frac{\Delta R_k}{R} \right] \right|$$

which can be simplified to

$$|V_i - V_{i-1}| = \left| \frac{V_{REF}}{2^N} \left( 1 + \frac{\Delta R_i}{R} \right) \right| \quad (29.11)$$

The DNL can then be determined by subtracting the ideal step height from Eq. (29.11),

$$DNL_i = \left| \frac{V_{REF}}{2^N} \left( 1 + \frac{\Delta R_i}{R} \right) - \frac{V_{REF}}{2^N} \right| = \left| \frac{V_{REF}}{2^N} \cdot \frac{\Delta R_i}{R} \right| \quad (29.12)$$

and the maximum DNL will occur at the value of  $i$  for which  $\Delta R$  is at its maximum value. If it is assumed once again that the resistors are matched to within 2 percent, the worst-case DNL will be

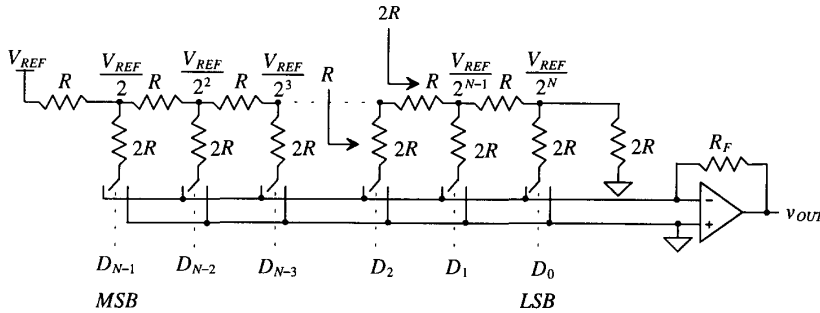
$$DNL_{max} = \left| 0.02 \cdot \frac{V_{REF}}{2^N} \right| = 0.02 \text{ LSB} \quad (29.13)$$

which is well below the  $\frac{1}{2}$  LSB limit. The INL is obviously the limiting factor in determining the resolution of a resistor string DAC since its maximum value is  $2^N$  times larger than the DNL.

#### **29.1.3 R-2R Ladder Networks**

Another DAC architecture that incorporates fewer resistors is called the  $R$ - $2R$  ladder network [1]. This configuration consists of a network of resistors alternating in value of  $R$  and  $2R$ . Figure 29.5 illustrates an  $N$ -bit  $R$ - $2R$  ladder. Starting at the right end of the network, notice that the resistance looking to the right of any node to ground is  $2R$ . The digital input determines whether each resistor is switched to ground (noninverting input) or to the inverting input of the op-amp. Each node voltage is related to  $V_{REF}$ , by a binary-weighted relationship caused by the voltage division of the ladder network. The total current flowing from  $V_{REF}$  is constant, since the potential at the bottom of each





**Figure 29.5** An  $R$ - $2R$  digital-to-analog converter.

switched resistor is always zero volts (either ground or virtual ground). Therefore, the node voltages will remain constant for any value of the digital input.

The output voltage,  $v_{OUT}$ , is dependent on currents flowing through the feedback resistor,  $R_F$ , such that

$$v_{OUT} = -i_{TOT} \cdot R_F \quad (29.14)$$

where  $i_{TOT}$  is the sum of the currents selected by the digital input by

$$i_{TOT} = \sum_{k=0}^{N-1} D_k \cdot \frac{V_{REF}}{2^{N-k}} \cdot \frac{1}{2R} \quad (29.15)$$

where  $D_k$  is the  $k$ -th bit of the input word with a value that is either a 1 or a 0.

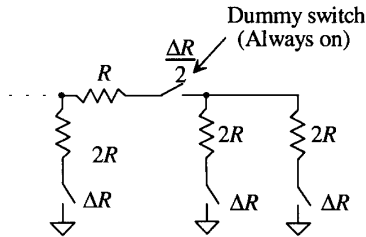
This architecture, like the resistor string architecture, requires matching to within the resolution of the converter. Therefore, the switch resistance must be negligible, or a small voltage drop will occur across each switch, resulting in an error. One way to eliminate this problem is to add dummy switches. Assume that the resistance of each switch connected to the  $2R$  resistors is  $\Delta R$ , as seen in Fig. 29.6. Dummy switches with one-half the resistance of the real switches are "hard-wired" so that they are always on and placed in series with each of the horizontal resistors. The total resistance of any horizontal branch,  $R'$ , is

$$R' = R + \frac{\Delta R}{2} \quad (29.16)$$

The resistance of any vertical branch is  $2R + \Delta R$ , which is twice the value of the horizontal branch. Therefore, a  $R' - 2R'$  relationship is maintained. Of course, a dummy switch equal to the switch size of a  $2R$  switch will have to be placed in series with the terminating resistor as well.

### Example 29.3

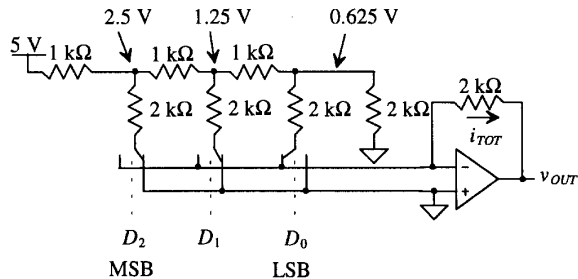
Design a 3-bit DAC using a  $R$ - $2R$  architecture with  $R = 1 \text{ k}\Omega$ ,  $R_F = 2 \text{ k}\Omega$ , and  $V_{REF} = 5 \text{ V}$ . Assume that the resistances of the switches are negligible.



**Figure 29.6** Use of dummy switches to offset switch resistance.

Determine the value of  $i_{TOT}$  for each digital input and the corresponding output voltage,  $v_{OUT}$ .

Figure 29.7 shows the 3-bit DAC for a digital input of 001. The voltages at each node in the resistor network are labeled. For each switch, if the digital input bit is a 0, then the resistor is attached to the ground. If the bit is a 1, then the resistor is attached to the virtual ground of the inverting input and current flows to the output of the op-amp. Therefore, for  $D_2D_1D_0 = 000$ , all the switches are connected to ground and no current flows through the feedback resistor and the output voltage,  $v_{OUT}$ , is zero.



**Figure 29.7** A 3-bit  $R$ - $2R$  digital-to-analog converter used in Ex. 29.3.

When  $D_2D_1D_0 = 001$ , the rightmost resistor is switched to the op-amp inverting input and the other two resistors remain attached to ground. Therefore, the total current flowing through the feedback resistor will simply be the current through the rightmost resistor, which is defined by Eq. (29.15) as

$$\frac{V_{REF}}{8} \cdot \frac{1}{2000} = 0.3126 \text{ mA}$$

and the output voltage, by Eq. (29.14) becomes,

$$v_{OUT} = -(0.3126 \text{ mA})(2000 \Omega) = -0.625 \text{ V}$$

which is to be expected. The other values for the output voltage can be calculated using Eqs. (29.14) and (29.15) and are seen in Fig. 29.8. ■

| $D_2D_1D_0$ | $i_{TOT}$ (mA)                   | $v_{OUT}$ (V) |
|-------------|----------------------------------|---------------|
| 000         | 0                                | 0             |
| 001         | 0.3125                           | - 0.625       |
| 010         | 0.625                            | - 1.25        |
| 011         | $0.625 + 0.3125 = 0.9375$        | - 1.875       |
| 100         | 1.25                             | - 2.5         |
| 101         | $1.25 + 0.3125 = 1.5625$         | - 3.125       |
| 110         | $1.25 + 0.625 = 1.875$           | - 3.75        |
| 111         | $1.25 + 0.625 + 0.3125 = 2.1875$ | - 4.375       |

**Figure 29.8** Output voltages generated from the 3-bit DAC in Example 29.3.

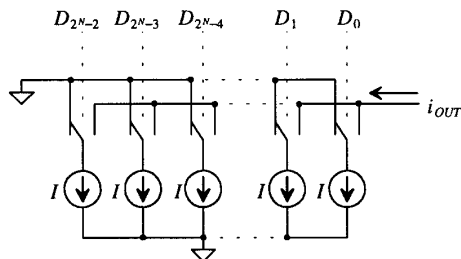
### 29.1.4 Current Steering

In the previous section, a voltage was converted into a current, which then generated a voltage at the output. Another DAC method uses current throughout the conversion. Known as *current steering*, this type of DAC requires precision current sources that are summed in various fashions.

Figure 29.9 illustrates a generic current steering DAC. This configuration requires a set of current sources, each having a unit value of current,  $I$ . Since there are no current sources generating  $i_{OUT}$  when all the digital inputs are zero, the MSB,  $D_{2^N-2}$ , is offset by two bits instead of one. For example, for a 3-bit converter, seven current sources will be needed, labeled from  $D_0$  to  $D_6$ . The binary signal controls whether or not the current sources are connected to either  $i_{OUT}$  or some other summing node (in this case ground). The output current,  $i_{OUT}$ , has the range of

$$0 \leq i_{OUT} \leq (2^N - 1) \cdot I \quad (29.17)$$

and can be any integer multiple of  $I$  in between. An interesting issue to note is the format of the digital code required to drive the switches. Since there are  $2^N - 1$  current sources, the digital input will be in the form of a *thermometer code*. This code will be all 1's from the LSB up to the value of the  $k$ -th bit,  $D_k$ , and all 0's above it. The point at

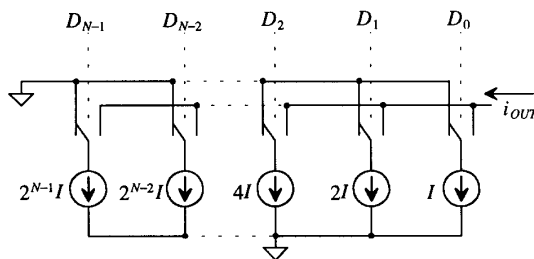


**Figure 29.9** A generic current steering DAC.

which the input code changes from all 1's to all 0's "floats" up or down and resembles the action of a thermometer, hence the name. Typically, a thermometer encoder is used to convert binary input data into a thermometer code.

Another current steering architecture is seen in Fig. 29.10. This architecture uses binary-weighted current sources, thus requiring only  $N$  current sources of various sizes versus  $2^N - 1$  sources in the previous example. Since the current sources are binary weighted, the input code can be a simple binary number with no thermometer encoder needed.

One advantage of the current steering DACs is the high-current drive inherent in the system. Since no output buffers are necessary to drive resistive loads, these DACs typically are used in high-speed applications. Traditionally, high-speed current steering DACs have been fabricated using bipolar technology. However, the ability to generate matched current mirrors makes CMOS an enticing alternative. Of course, the precision needed to generate high resolutions is dependent on how well the current sources can be matched or the degree to which they can be made binary weighted. For example, if a 13-bit DAC was designed using these architectures, there would have to be 8,191 current sources resident on the chip - not an insignificant amount. For the



**Figure 29.10** A current steering DAC using binary-weighted current sources.

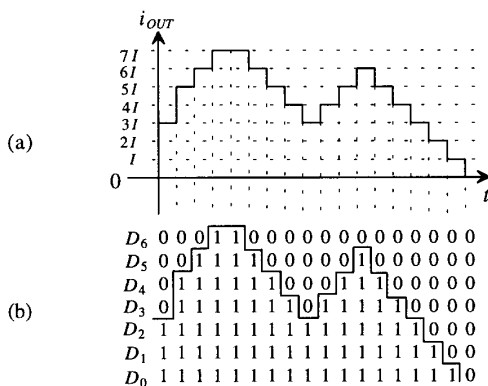
binary-weighted sources, only 13 current sources would be needed. Yet the size of the largest current source would have to be 4096 or  $2^{N+1}$  times larger than the smallest. Even if the unit current,  $I$ , was chosen to be  $5\text{ }\mu\text{A}$ , the largest current source would be 20.48 mA!

Another problem associated with this architecture is the error due to the switching. Since the current sources are in parallel, if one of the current sources is switched off and another is switched on, a "glitch" could occur in the output if the timing was such that both of them were on or both were off for an instant. While this may not seem significant, if the converter is switching from 011111 to 1000000, the output will spike toward ground and then back to the correct value if all the switches turn off for an instant. If the DAC is driving a resistive load and the output current is converted to a voltage, a substantial voltage spike will occur at the output.

#### Example 29.4

Construct a table showing the thermometer code necessary to generate the output shown in Fig. 29.11a for a 3-bit current steering DAC using unit current sources.

The thermometer code can be seen in Fig. 29.11b. When the code is all zeros, the output is zero volts. Therefore, only 7 bits are needed to represent the  $2^N$  or 8 states of a 3-bit DAC. Note how the interface between all 1's and all 0's actually resembles the output signal itself. ■



**Figure 29.11** (a) Output of a 3-bit current steering DAC and (b) the thermometer code input.

#### Mismatch Errors Related to Current Steering DACs

Analysis of the mismatch associated with the current sources is similar to the resistor string analysis. It is assumed that each current source in Fig. 29.9 is

$$I_k = I + \Delta I_k \text{ for } k = 1, 2, 3, \dots, 2^N - 1 \quad (29.18)$$

where  $I$  is the ideal value of the current and  $\Delta I_k$  is the error due to mismatch. If it is again assumed that the  $\Delta I_k$  terms sum to zero and that one-half of the current sources contain the maximum positive mismatch,  $\Delta I_{max}$ , and the other half contains the maximum negative mismatch,  $-\Delta I_{max}$ , (or vice versa), then the worst-case condition will occur at midscale with the actual output current being

$$I_{out} = \sum_{k=1}^{2^{N-1}} (I + \Delta I_k) = 2^{N-1} \cdot I + 2^{N-1} \cdot |\Delta I|_{max} = I_{out,ideal} + 2^{N-1} \cdot |\Delta I|_{max} \quad (29.19)$$

Since the INL is simply the actual output current minus the ideal, the worst-case INL will be

$$|INL|_{max} = 2^{N-1} \cdot |\Delta I|_{max,INL} \quad (29.20)$$

The term,  $|\Delta I|_{max,INL}$  represents the maximum current source mismatch error that will keep the INL less than  $\frac{1}{2}$  LSB. Each current source represents the value of 1 LSB; therefore,  $\frac{1}{2}$  LSB is equal to  $0.5 I$ . Since the maximum INL should correspond to the  $\frac{1}{2}$  LSB, equating Eq. (29.20) to  $\frac{1}{2} I$  results in the value for  $|\Delta I|_{max,INL}$ ,

$$|\Delta I|_{max,INL} = \frac{0.5I}{2^{N-1}} = \frac{I}{2^N} \quad (29.21)$$

Equation (29.21) illustrates the difficulty of using this architecture at high resolutions. If the value of  $I$  is set to be  $5 \mu A$ , and the  $N$  is desired to be 12 bits, then the value of  $|\Delta I|_{max,INL}$  becomes

$$|\Delta I|_{max,INL} = \frac{5 \times 10^{-6}}{2^{12}} = 1.221 \text{ nA!} \quad (29.22)$$

which means that each of the  $5 \mu A$  current sources must lie between the bounds of,

$$4.99878 \mu A \leq I_k \leq 5.001221 \mu A \quad (29.23)$$

to achieve a worst-case INL, which is within  $\frac{1}{2}$  LSB error.

The DNL is easily obtained since the step height in the transfer curve is equivalent to the value of the ideal current source,  $I$ . The maximum difference between any two adjacent values of output current will be simply the value of the single source,  $I_k$ , which contains the largest mismatch error for which the DNL will be less than  $\frac{1}{2}$  LSB,  $|\Delta I|_{max,DNL}$ :

$$I_{out(x)} - I_{out(x-1)} = I_k + |\Delta I|_{max,DNL} \quad (29.24)$$

Therefore, the DNL is simply

$$|DNL|_{max} = I_k + |\Delta I|_{max,DNL} - I_k = |\Delta I|_{max,DNL} \quad (29.25)$$

Equating the maximum DNL to the value of  $\frac{1}{2}$  LSB,

$$|\Delta I|_{max,DNL} = \frac{1}{2} \text{ LSB} = \frac{1}{2} I \quad (29.26)$$

which is much easier to attain than the requirement for the INL.

For the binary-weighted current sources seen in Fig. 29.10, a slightly different analysis is needed to determine the requirements for INL and DNL. In this case, it will be assumed that the current source corresponding to the MSB ( $D_{N-1}$ ) has a maximum positive mismatch error value and the remainder of the bits ( $D_0$  to  $D_{N-2}$ ) contain a maximum negative mismatch error, so that the sum of all the errors equals zero. Therefore, the INL is

$$|INL|_{max} = 2^{N-1}(I + |\Delta I|_{max,INL}) - 2^{N-1} \cdot I = 2^{N-1} \cdot |\Delta I|_{max,INL} \quad (29.27)$$

which is equivalent to the value of the current steering array in Fig. 29.9.

The DNL is slightly different because of the binary weighting of the current sources. One cannot add a single current source with each incremental increase in the digital input code. However, the worst-case condition for binary-weighted arrays tends to occur at midscale when the code transitions from 011111...111 to 100000...000. The worst-case DNL at this point is

$$DNL_{max} = \left[ 2^{N-1} \cdot (I + |\Delta I|_{max,DNL}) - \sum_{k=1}^{N-1} 2^{k-1} \cdot (I - |\Delta I|_{max,DNL}) \right] - I \quad (29.28)$$

which can be written as

$$DNL_{max} = 2^{N-1} \cdot (I + |\Delta I|_{max,DNL}) - (2^{N-1} - 1) \cdot (I - |\Delta I|_{max,DNL}) - I = (2^N - 1) \cdot |\Delta I|_{max,DNL} \quad (29.29)$$

and setting this value equal to  $\frac{1}{2}$  LSB and solving for  $\Delta I_{max}$ ,

$$|\Delta I|_{max,DNL} = \frac{0.5I}{2^N - 1} = \frac{I}{2^{N+1} - 2} \quad (29.30)$$

Therefore, the DNL requirements for the binary-weighted current source array is more stringent than the INL requirements.

One interesting issue regarding the previous derivation is that the challenging accuracy requirements in Eq. (29.30) are placed only on the MSB current source. For each of the remaining binary-weighted sources, the DNL requirements become more relaxed. This is simply because the size of the MSB source is equivalent to all the other sources combined, and so its value plays the most important role in the DAC's accuracy.

### Example 29.5

Determine the tolerance of the MSB current source on a 10-bit binary-weighted current source array with a unit current source of  $1 \mu\text{A}$ , which will result in a worst-case DNL that is less than  $\frac{1}{2}$  LSB.

Since Eq. (29.30) defines the maximum  $|\Delta I|$  needed to keep the DNL less than  $\frac{1}{2}$  LSB, we must first use this equation,

$$|\Delta I|_{\max, DNL} = \frac{1 \times 10^{-6}}{2^{11} - 2} = 0.4888 \text{ nA}$$

For a 10-bit DAC, the MSB current source will have a value that is  $2^9$  times larger than the unit current source, or 0.512 mA. Therefore, the range of values for which this array will have a DNL that is less than  $\frac{1}{2}$  LSB is

$$0.51199995 \text{ mA} \leq I_{MSB} \leq 0.5120004888 \text{ mA} \quad \blacksquare$$

### 29.1.5 Charge Scaling DACs

A very popular DAC architecture used in CMOS technology is the charge scaling DAC. Shown in Fig. 29.12a, a parallel array of binary-weighted capacitors, totaling  $2^N C$ , is connected to an op-amp. The value,  $C$ , is a unit capacitance of any value. After initially being discharged, the digital signal switches each capacitor to either  $V_{REF}$  or ground, causing the output voltage,  $v_{OUT}$ , to be a function of the voltage division between the capacitors.

Since the capacitor array totals  $2^N C$ , if the MSB is high and the remaining bits are low, then a voltage divider occurs between the MSB capacitor and the rest of the array. The analog output voltage,  $v_{OUT}$ , becomes

$$v_{OUT} = V_{REF} \cdot \frac{2^{N-1} C}{(2^{N-1} + 2^{N-2} + 2^{N-3} + \dots + 4 + 2 + 1 + 1)C} = V_{REF} \cdot \frac{2^{N-1} C}{2^N C} = \frac{V_{REF}}{2} \quad (29.31)$$

which confirms the fact that the MSB changes the output of a DAC by  $\frac{1}{2} V_{REF}$ . Figure 29.12b shows the equivalent circuit under this condition. The ratio between  $v_{OUT}$  and  $V_{REF}$  due to each capacitor can be generalized to

$$v_{OUT} = \frac{2^k C}{2^N C} \cdot V_{REF} = 2^{k-N} \cdot V_{REF} \quad (29.32)$$

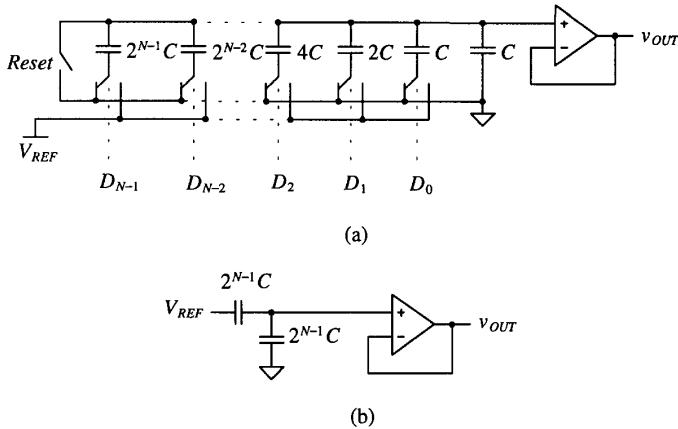
where it is assumed that the  $k$ -th bit,  $D_k$ , is 1 and all other bits are zero. Superposition can then be used to find the value of  $v_{OUT}$  for any digital input word by

$$v_{OUT} = \sum_{k=0}^{N-1} D_k 2^{k-N} \cdot V_{REF} \quad (29.33)$$

One limitation of this architecture as shown in Fig. 29.12a is the existence of a parasitic capacitance at the top plate of the capacitor array due to the op-amp. This will prohibit its use as a high-resolution data converter. A better implementation would include the use of a parasitic insensitive, switched-capacitor integrator (see Ch. 27) as the driving circuit. However, the capacitor array itself is the critical component of this data converter and is used in charge redistribution ADCs (Sec. 29.2.5).

The INL and DNL calculations for the binary-weighted capacitor array are identical to those for the binary-weighted current source array, except the unit current source,  $I$ , and its corresponding error term,  $\Delta I$ , are replaced by  $C$  and  $\Delta C$  in Eqs. (29.27) – (29.30).





**Figure 29.12** (a) A charge scaling DAC, (b) the equivalent circuit with the MSB = 1, and all other bits set to zero.

### Example 29.6

Design a 3-bit charge scaling DAC and find the value of the output voltage for  $D_2D_1D_0 = 010$  and  $101$ . Assume that  $V_{REF} = 5$  V and  $C = 0.5$  pF.

The 3-bit DAC can be seen in Fig. 29.13a. The equivalent circuits for the capacitor array can be seen in Fig. 29.13b and c. The value of the output voltage can be calculated by either using Eq. (29.32) or the equivalent circuits and performing the voltage division. For  $D = 010$ , the equivalent circuit in Fig. 29.13b yields

$$v_{OUT} = V_{REF} \cdot \left(\frac{1}{4}\right) = 1.25 \text{ V}$$

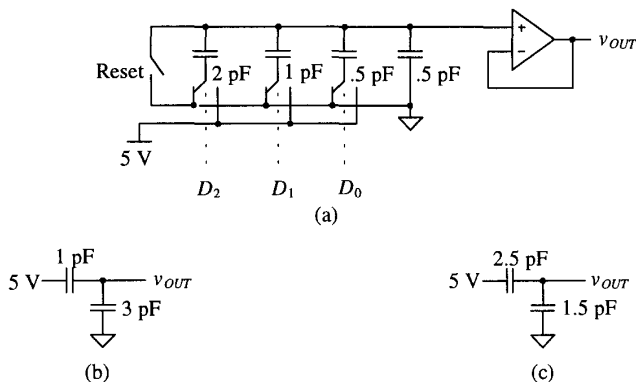
Using Eq. (29.33) to calculate  $v_{OUT}$  for  $D = 101$  yields

$$v_{OUT} = \sum_{k=0}^{N-1} D_k 2^{k-N} \cdot V_{REF} = [1 \cdot (2^{-3}) + 0 \cdot (2^{-2}) + 1 \cdot (2^{-1})] \cdot 5 = \left(\frac{1}{8} + \frac{1}{2}\right) \cdot 5 = 3.125 \text{ V}$$

which is the result expected. ■

### Layout Considerations for a Binary-Weighted Capacitor Array

One problem with this converter is the need for precisely ratioed capacitors. As the number of bits increase, the ratio of the MSB capacitor to the LSB capacitor becomes more difficult to control. For example, Fig. 29.14a shows a 3-bit binary capacitor array



**Figure 29.13** (a) A 3-bit charge scaling DAC used in Ex. 29.6 and the equivalent circuits inputs equal to (b) 010 (c) 101.

using three capacitors. When the capacitor is fabricated, *undercutting* of the mask [2, 3] will cause an error in the ratio of the capacitors, causing potentially large DNL and INL errors as  $N$  increases.

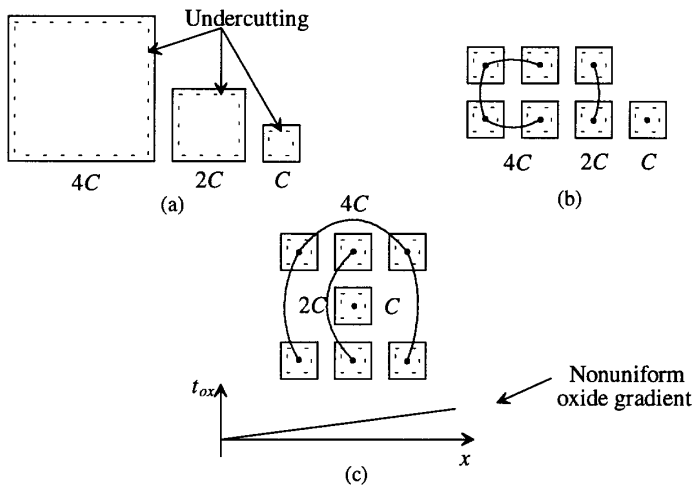
One solution to this problem is seen in Fig. 20.14b. Here, each capacitor in the array is constructed out of a unit capacitance. Undercutting then affects all the capacitors in the same way, and the ratio between capacitors is maintained. Another problem that affects even this layout strategy is a nonuniform oxide growth. Gradient errors will result in errors in the ratios of the capacitors. Figure 29.14c illustrates another layout strategy that overcomes this issue. The capacitors are laid out in a common-centroid scheme so that the first-order oxide errors average out to be the same for each capacitor.

#### The Split Array

The charge scaling architecture is very popular among CMOS designers because of its simplicity and relatively good accuracy. Although a linear capacitor is required using poly2, high resolutions in the 10 to 12-bit range can be achieved. Passive, double-poly capacitors have good matching accuracy as well. However, as the resolution increases, the size of the MSB capacitor becomes a major concern. For example if the unit capacitor,  $C$ , was chosen to be 0.5 pF, and a 16-bit DAC was to be designed, the MSB capacitor would need to be

$$C_{MSB} = 2^{N-1} \cdot 0.5 \text{ pF} = 16.384 \text{ nF} \quad (29.34)$$

Based on our 2  $\mu\text{m}$  process parameters, the capacitance between poly1 and poly2 is nominally 500 aF/ $\mu\text{m}^2$ . Therefore, the area required for this one capacitor is over 30 million square microns!



**Figure 29.14** Layout of a binary-weighted capacitor array using (a) single capacitors (b) unit capacitors to minimize undercutting effect, and (c) common-centroid to minimize oxide gradients.

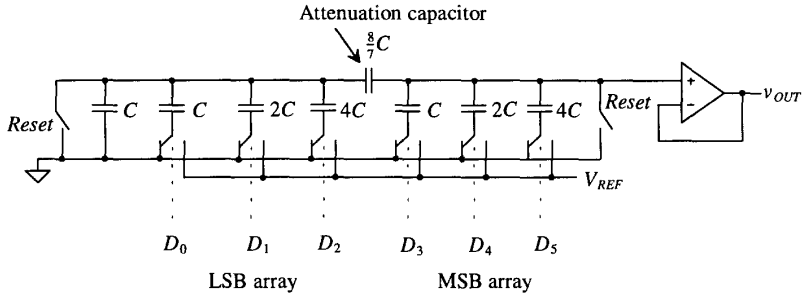
One method to reduce the size of the capacitors is to use a split array. A 6-bit example of the array is pictured in Fig. 29.15. This architecture [1] is slightly different from the charge scaling DAC pictured in Fig. 29.13 in that the output is taken off a different node and an additional attenuation capacitor is used to separate the array into a LSB array and a MSB array. Note that the LSB,  $D_0$ , now corresponds to the leftmost switch and that the MSB,  $D_5$ , corresponds to the rightmost switch. The value of the attenuation capacitor can be found by

$$C_{atten} = \frac{\text{sum of the LSB array capacitors}}{\text{sum of the MSB array capacitors}} \cdot C \quad (29.35)$$

where the sum of the MSB array is equal to the sum of LSB capacitor array minus  $C$ . The value of the attenuation capacitor should be such that the series combination of the attenuation capacitor and the LSB array, assuming all bits are zero, is equal to  $C$ .

#### Example 29.7

Using the 6-bit charge scaling DAC shown in Fig. 29.15, (a) show that the output voltage will be  $\frac{1}{2} \cdot V_{REF}$  if (a)  $D_5 D_4 D_3 D_2 D_1 D_0 = 100000$  and (b) the output will be  $\frac{1}{64} \cdot V_{REF}$  if  $D_5 D_4 D_3 D_2 D_1 D_0 = 000001$ .



**Figure 29.15** A charge scaling DAC using a split array.

(a) If  $D_5 = 1$  and the remaining bits are all zero, then the equivalent circuit for the DAC can be represented by Fig. 29.16a. The expression for the output voltage then becomes

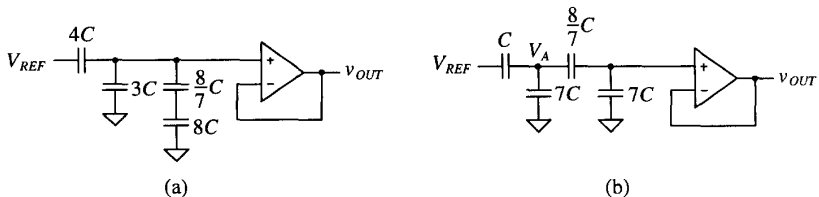
$$v_{OUT} = \frac{4}{\left(\frac{8}{7} \parallel 8\right) + 3 + 4} \cdot V_{REF} = \frac{1}{2} \cdot V_{REF}$$

(b) For the second case, the equivalent circuit can be seen in Fig. 29.16b. The intermediate node voltage,  $V_A$ , is simply the voltage division between  $C$  and the remainder of the circuit, or

$$V_A = V_{REF} \cdot \frac{1}{\left(7 + \frac{8}{7}\right) + 1} = \frac{1}{8 + \frac{56}{57}} \cdot V_{REF} \quad (29.36)$$

The output voltage can be written as

$$v_{OUT} = V_A \cdot \frac{\frac{8}{7}}{\frac{8}{7} + 7} = \frac{8}{57} \cdot V_A \quad (29.37)$$



**Figure 29.16** Equivalent circuits for Example 29.7.

Plugging Eq. (29.36) into Eq. (29.37) yields

$$v_{OUT} = V_{REF} \cdot \frac{8}{(8 \cdot 57) + 56} = \frac{V_{REF}}{64} \quad (29.38)$$

which is the desired result. ■

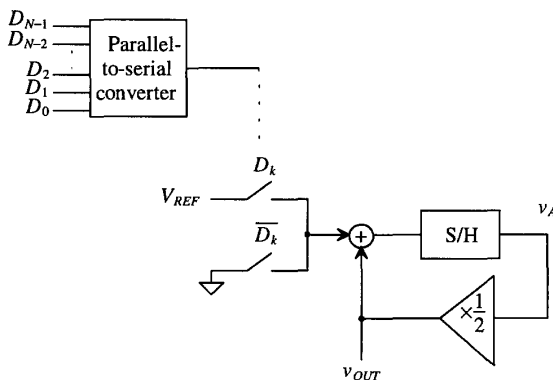
### 29.1.6 Cyclic DAC

The cyclic DAC uses only a couple of simple components to perform the conversion. As seen in Fig. 29.17, a summer adds  $V_{REF}$  or ground to the feedback signal depending on the input bits. An amplifier with a gain of 0.5 feeds the output voltage back to the summer such that the output at the end of each cycle is dependent on the value of the output during the cycle before. Notice that the input bits must be read in a serial fashion. Therefore, the conversion is performed one bit at a time, resulting in  $N$  cycles required for each conversion. The voltage output at the end of the  $n$ -th cycle of the conversion can be written as

$$v_{OUT}(n) = \left( D_{n-1} \cdot V_{REF} + \frac{1}{2} \cdot v_A(n-1) \right) \cdot \frac{1}{2} \quad (29.39)$$

with a condition such that the output of the S/H is initially zero [ $v_A(0) = 0$  V].

The accuracy of this converter is dependent on several factors. The gain of the 0.5 amplifier needs to be highly accurate (to within the accuracy of the DAC) and is usually generated with passive capacitors. Similarly, the summer and the sample-and-hold also need to be  $N$ -bit accurate. Limitations of the converter due to these fundamental building blocks will be discussed in more detail in Sec. 29.3. Since this converter uses a pseudo-"sampled-data" approach, implementing this architecture using switched capacitors is relatively easy.



**Figure 29.17** A cyclic digital-to-analog converter.

**Example 29.8**

Show the value of the output voltage at the end of each cycle for a 6-bit cyclic DAC with an input value of  $D_5D_4D_3D_2D_1D_0 = 110101$ . Assume that  $V_{REF} = 5$  V.

We can predict the value of the output based on our previous experience with DACs. The digital input 110101 corresponds to  $53_{10}$ . Therefore, the output voltage due to this input should be

$$v_{OUT} = \frac{53}{64} \cdot V_{REF} = 4.140625 \text{ V}$$

Now examine the cyclic converter in Fig. 29.17. By performing a 6-bit conversion and using Eq. (29.39), the outputs occurring at the end of each cycle can be seen in Fig. 29.18.

The output voltage at the end of the sixth cycle is precisely what was predicted. Note that had this been a 3-bit conversion, the output voltage at the end of cycle 3 would correspond to the value of the 3-bit DACs studied previously with an input of 101. ■

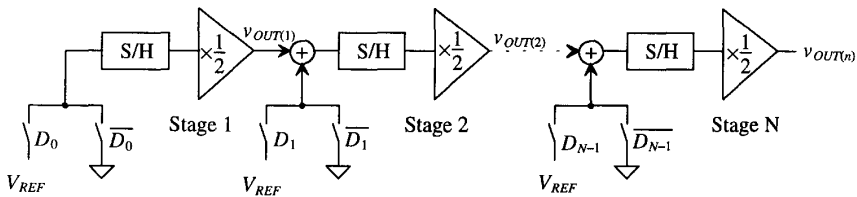
| Cycle Number, $n$ | $D_{n-1}$ | $v_A(n-1)$ | $v_{OUT}(n)$                                    |
|-------------------|-----------|------------|-------------------------------------------------|
| 1                 | 1         | 0          | $\frac{1}{2}(5 + 0) = 2.5 \text{ V}$            |
| 2                 | 0         | 5          | $\frac{1}{2}(0 + 2.5) = 1.25 \text{ V}$         |
| 3                 | 1         | 2.5        | $\frac{1}{2}(5 + 1.25) = 3.125 \text{ V}$       |
| 4                 | 0         | 6.25       | $\frac{1}{2}(0 + 3.125) = 1.5625 \text{ V}$     |
| 5                 | 1         | 3.125      | $\frac{1}{2}(5 + 1.5625) = 3.28125 \text{ V}$   |
| 6                 | 1         | 6.5625     | $\frac{1}{2}(5 + 3.28125) = 4.140625 \text{ V}$ |

**Figure 29.18** Output from the 6-bit cyclic DAC used in Ex. 29.8.

### 29.1.7 Pipeline DAC

The cyclic converter presented in the last section takes  $N$  clock cycles per  $N$ -bit conversion. Instead of recycling the output back to the input each time, we could extend the cyclic converter to  $N$  stages, where each stage performs one bit of the conversion. This extension of the cyclic converter is called a *pipeline* DAC and is seen in Fig. 29.19. Here, the signal is passed down the "pipeline," and as each stage works on one conversion, the previous stage can begin processing another. Therefore, an initial  $N$  clock cycle delay is experienced as the signal makes its way down the pipeline the very first time. However, after the  $N$  clock cycle delay, a conversion takes place at every clock cycle.

Besides the  $N$  clock cycle delay, this architecture can be very fast. However, the amplifier gains must be very accurate to produce high resolutions. Also, this



**Figure 29.19** A pipeline digital-to-analog converter.

architecture uses  $N$  times more circuitry than that of the cyclic, so there is a tradeoff between speed and chip area. The output voltage of the  $n$ -th stage in the converter can be written as

$$v_{OUT(n)} = [D_{n-1} \cdot V_{REF} + v_{OUT(n-1)}] \cdot \frac{1}{2} \quad (29.40)$$

The operation of each stage in the pipeline can be summarized as follows: if the input bit is a 1, add  $V_{REF}$  to the output of the previous stage, divide by two, and pass the value to the next stage. If the input bit is a 0, simply divide the output of the previous stage by two and pass along the resulting value.

### Example 29.9

Find the output voltage for a 3-bit pipeline DAC for three cases:  $D_A = 001$ ,  $D_B = 110$ , and  $D_C = 101$  and show that the conversion time to perform all three conversions is five clock cycles using the pipeline approach. Assume that  $V_{REF} = 5$  V.

The first stage will operate on the LSBs of each word; the second stage will operate on the middle bits; and the last stage, the MSBs. Based on the pipeline strategy, once the LSB of the first input word is performed and passed on, the LSB of the second word,  $D_B$ , can begin its conversion. Similarly, once the LSB of the second stage is completed and passed on, the LSB of the third word,  $D_C$ , can begin. The conversion cycle for all three input words will produce the output shown in Fig. 29.20. The items that are in **bold** are associated with the first input word,  $D_A$ , whereas the italicized numbers represent the values associated with  $D_B$  and the underlined items,  $D_C$ .

The first output of the DAC is not valid until the end of the third clock cycle and should look familiar as the 3-bit DAC output for an input word of  $D_2D_1D_0 = 001$ . The following two clock cycles that produce outputs for  $D_2D_1D_0$  equal 110 and 101, respectively. ■

| Clock Cycle | $v_{OUT(1)}$ | $v_{OUT(2)}$ | $v_{OUT(3)}$ | $D_0$    | $D_1$    | $D_2$    |
|-------------|--------------|--------------|--------------|----------|----------|----------|
| 1           | <b>2.5</b>   | 0            | 0            | <b>1</b> | 0        | 0        |
| 2           | 0            | <b>1.25</b>  | 0            | 0        | <b>0</b> | 0        |
| 3           | <u>2.5</u>   | 2.5          | <b>0.625</b> | <u>1</u> | <u>1</u> | <b>0</b> |
| 4           |              | <u>1.25</u>  | 3.75         |          | <u>0</u> | <u>1</u> |
| 5           |              |              | <u>3.125</u> |          |          | <u>1</u> |

**Figure 29.20** Output from the 3-bit pipeline DAC used in Example 29.9.

## 29.2 ADC Architectures

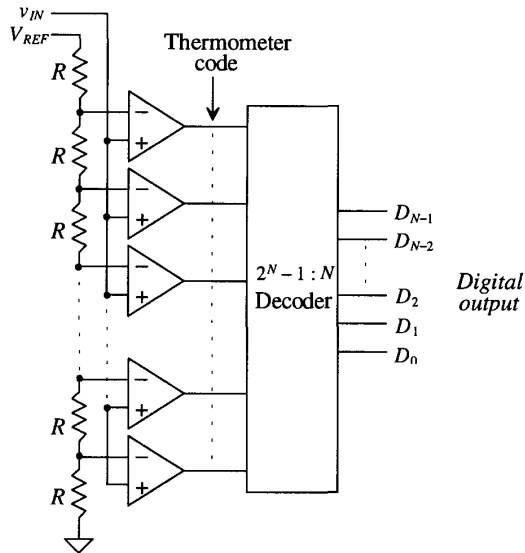
A survey of the field of current A/D converter research reveals that a majority of effort has been directed to four different types of architectures: pipeline, flash-type, successive approximation, and oversampled ADCs. Each has benefits that are unique to that architecture and span the spectrum of high speed and resolution.

Since the ADC has a continuous, infinite-valued signal as its input, the important analog points on the transfer curve x-axis for an ADC are the ones that correspond to changes in the digital output word. These input transitions determine the amount of INL and DNL associated with the converter.

### 29.2.1 Flash

Flash or parallel converters have the highest speed of any type of ADC. As seen in Fig. 29.21, they utilize one comparator per quantization level ( $2^N - 1$ ) and  $2^N$  resistors (a resistor string DAC). The reference voltage is divided into  $2^N$  values, each of which is fed into a comparator. The input voltage is compared with each reference value and results in a thermometer code at the output of the comparators. A thermometer code will exhibit all zeros for each resistor level if the value of  $v_{IN}$  is less than the value on the resistor string, and ones if  $v_{IN}$  is greater than or equal to voltage on the resistor string. A simple  $2^N - 1:N$  digital thermometer decoder circuit converts the compared data into an  $N$ -bit digital word. The obvious advantage of this converter is the speed with which one conversion can take place. Each clock pulse generates an output digital word. The advantage of having high speed, however, is counterbalanced by the doubling of area with each bit of increased resolution. For example, an 8-bit converter requires 255 comparators, but a 9-bit ADC requires 511! Flash converters have traditionally been limited to 8-bit resolution with conversion speeds of 10 – 40 Ms/s using CMOS technology [4–6]. The disadvantages of the Flash ADC are the area and power requirements of the  $2^N - 1$  comparators. The speed is limited by the switching of the comparators and the digital logic.





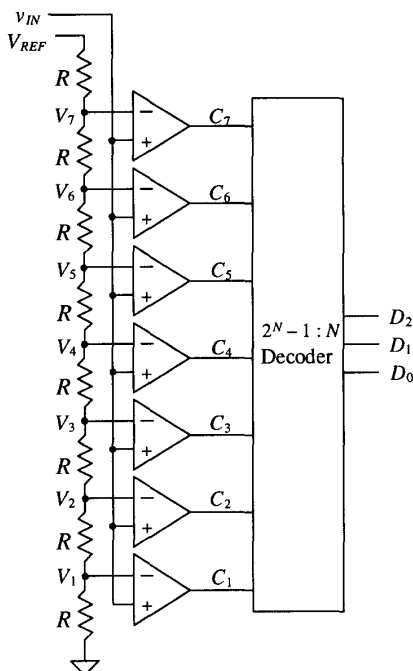
**Figure 29.21** Block diagram of a Flash ADC.

### Example 29.10

Design a 3-bit Flash converter, listing the values of the voltages at each resistor tap, and draw the transfer curve for  $v_{IN} = 0$  to 5 V. Assume  $V_{REF} = 5$  V. Construct a table listing the values of the thermometer code and the output of the decoder for  $v_{IN} = 1.5$ , 3.0, and 4.5 V.

The 3-bit converter can be seen in Fig. 29.22. Since the values of all the resistors are equal, the voltage of each resistor tap,  $V_i$ , will be  $V_i = V_{REF} \left( \frac{i}{8} \right)$  where  $i$  is the number of the resistor in the string for  $i = 1$  to 7. Obviously,  $V_1 = 0.625$  V,  $V_2 = 1.25$  V,  $V_3 = 1.875$  V,  $V_4 = 2.5$  V,  $V_5 = 3.125$  V,  $V_6 = 3.75$  V,  $V_7 = 4.375$  V. Therefore, when  $v_{IN}$  first becomes equal or greater than each of these values, a transition will occur in the transfer curve. The transfer curve can be seen in Fig. 29.23 and should look similar to those seen in Ch. 28. The quantization levels and their corresponding thermometer codes can be summarized as seen in Fig. 29.24.

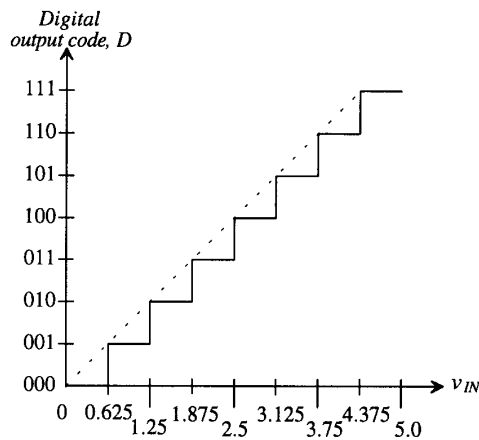
The transfer curve of this ADC corresponds to the ADC with quantization error centered about  $+\frac{1}{2}$  LSB, as discussed in Ch. 28 (Fig. 28.20). To shift the curve by  $\frac{1}{2}$  LSB so that the code transitions occur around the LSB values and the



**Figure 29.22** Three-bit Flash A/D converter to be used in Ex. 29.10.

quantization error is centered around 0 LSB, the value of the last resistor in the string would have to be adjusted to  $\frac{R}{2}$  and the value of the MSB resistor, closest to the reference voltage, would have to be made  $1.5R$ . Then the first code transition would occur at  $v_{IN} = 0.3125$  V, and the last code transition would occur at  $v_{IN} = 4.0625$  and so the transfer curve would exactly match that of Fig. 28.20.

Based on Fig. 29.24, when  $v_{IN} = 1.5$  V, only comparators  $C_1$  and  $C_2$  will have outputs of 1, since both  $V_1$  and  $V_2$  are less than  $1.5$  V. The remaining comparator outputs will be 0 since  $V_3$  through  $V_8$  will be greater than  $1.5$  V, thus generating the thermometer code, 0000011. The encoder must then convert this into a 3-bit digital word, resulting in 010. The same reasoning can be used to construct the data shown in Fig. 29.25. It should be obvious that if the polarity of the comparators were reversed, the thermometer code would be inverted. ■



**Figure 29.23** Transfer curve for the 3-bit Flash converter in Example 29.10.

| $v_{IN}$                                       | $C_7C_6C_5C_4C_3C_2C_1$ | $D_2D_1D_0$ |
|------------------------------------------------|-------------------------|-------------|
| $0 \leq v_{IN} < 0.625 \text{ V}$              | 0000000                 | 000         |
| $0.625 \text{ V} \leq v_{IN} < 1.25 \text{ V}$ | 0000001                 | 001         |
| $1.25 \text{ V} \leq v_{IN} < 1.875 \text{ V}$ | 0000011                 | 010         |
| $1.875 \text{ V} \leq v_{IN} < 2.5 \text{ V}$  | 0000111                 | 011         |
| $2.5 \text{ V} \leq v_{IN} < 3.125 \text{ V}$  | 0001111                 | 100         |
| $3.125 \text{ V} \leq v_{IN} < 3.75 \text{ V}$ | 0011111                 | 101         |
| $3.75 \text{ V} \leq v_{IN} < 4.375 \text{ V}$ | 0111111                 | 110         |
| $4.375 \leq v_{IN}$                            | 1111111                 | 111         |

**Figure 29.24** Code transitions for the Flash ADC used in Ex. 29.10.

*Accuracy Issues for the Flash ADC*

Accuracy is dependent on the matching of the resistor string and the input offset voltage of the comparators. From Sec. 26.1, we know that an ideal comparator should switch at the point at which the two inputs,  $v_+$  and  $v_-$ , are the same potential. However, the offset voltage  $V_{os}$ , prohibits this from occurring as the comparator output switches state as follows:

$$v_o = 1 \quad \text{when } v_+ \geq v_- + V_{os} \tag{29.41}$$

| $V_{IN}$ | $C_7C_6C_5C_4C_3C_2C_1$ | $D_2D_1D_0$ |
|----------|-------------------------|-------------|
| 1.5      | 0000011                 | 010         |
| 3.0      | 0001111                 | 100         |
| 4.5      | 1111111                 | 111         |

**Figure 29.25** Output for the Flash ADC used in Ex. 29.10.

$$v_o = 0 \quad \text{when } v_+ < v_- + V_{os} \quad (29.42)$$

The resistor string DAC was analyzed and presented in Sec. 29.1.2; the voltage on the  $i$ -th tap of the resistor string was found to be

$$V_i = V_{i,ideal} + \frac{V_{REF}}{2^N} \cdot \sum_{k=1}^i \frac{\Delta R_k}{R} \quad (29.43)$$

where  $V_{i,ideal}$  is the voltage at the  $i$ -th tap if all the resistors had an ideal value of  $R$ . The term,  $\Delta R_k$ , is the value of the resistance error (difference from ideal) due to the mismatch. Note that for the resistor string DAC, the sum of the mismatch terms plays an important factor in the overall voltage at each tap.

The switching point for the  $i$ -th comparator,  $V_{sw,i}$ , then becomes

$$V_{sw,i} = V_i + V_{os,i} \quad (29.44)$$

where  $V_{os,i}$  is the input referred offset voltage of the  $i$ -th comparator. The INL for the converter can then be described as

$$INL = V_{sw,i} - V_{sw,ideal} = V_{sw,i} - V_{i,ideal} \quad (29.45)$$

which becomes

$$INL = \frac{V_{REF}}{2^N} \cdot \sum_{k=1}^i \frac{\Delta R_k}{R} + V_{os,i} \quad (29.46)$$

The worst-case INL will occur at the middle of the string ( $i = 2^{N-1}$ ) as described in Sec. 29.1.2 and Eq. (29.10). Including the offset voltage, the maximum INL will be

$$|INL|_{max} = \frac{V_{REF}}{2^N} \cdot \sum_{k=1}^{2^{N-1}} \frac{\Delta R_k}{R} + |V_{os,i}|_{max} = V_{REF} \cdot \frac{2^{N-1}}{2^N R} \cdot |\Delta R_k|_{max} + |V_{os,i}|_{max} \quad (29.47)$$

which can be rewritten as

$$|INL|_{max} = \frac{V_{REF}}{2} \cdot \left| \frac{\Delta R_k}{R} \right|_{max} + |V_{os,i}|_{max} \quad (29.48)$$

where it is assumed that the maximum positive mismatch occurs in all the resistors in the lower half of the string and the maximum negative mismatch occurs in the upper

half (or vice versa) and that the comparator at the  $i$ -th tap contains the maximum offset voltage,  $|V_{os,i}|_{max}$ . Notice that the offset contributes directly to the maximum value for the INL. This explains another limitation to using Flash converters at high resolutions. The offset voltage alone can make the INL greater than  $\frac{1}{2}$  LSB.

### Example 29.11

If a 10-bit Flash converter is designed, determine the maximum offset voltage of the comparators which will make the INL less than  $\frac{1}{2}$  LSB. Assume that the resistor string is perfectly matched and  $V_{REF} = 5$  V.

Equation (29.48) requires that the offset voltage be equal to  $\frac{1}{2}$  LSB. Therefore,

$$|V_{os}|_{max} = \frac{5}{2^{11}} = 2.44 \text{ mV} \quad \blacksquare$$

The DNL calculation for the Flash converter is also attained using the analysis first presented in Sec. 29.1.2. Using the definition of DNL,

$$DNL = V_{sw,i} - V_{sw,i-1} - 1 \text{ LSB (in volts)} \quad (29.49)$$

Plugging in Eq. (29.44),

$$DNL = V_i + V_{os,i} - V_{i-1} - V_{os,i-1} - 1 \text{ LSB} \quad (29.50)$$

which can be written by using Eq. (29.6) as

$$DNL = V_{i,ideal} - V_{i-1,ideal} + \frac{V_{REF}}{2^N} \cdot \frac{\Delta R_i}{R} + V_{os,i} - V_{os,i-1} - 1 \text{ LSB} \quad (29.51)$$

which becomes

$$DNL = \frac{V_{REF}}{2^N} \cdot \frac{\Delta R_i}{R} + V_{os,i} - V_{os,i-1} \quad (29.52)$$

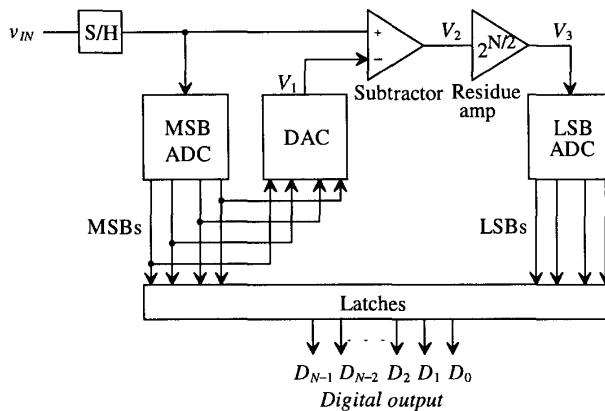
The maximum DNL will occur assuming  $\Delta R_i$  is at its maximum,  $V_{os,i}$  is at its maximum positive value, and  $V_{os,i-1}$  is at its maximum negative voltage. Thus,

$$|DNL|_{max} = \frac{V_{REF}}{2^N} \cdot \left| \frac{\Delta R_i}{R} \right|_{max} + 2|V_{os}|_{max} \quad (29.53)$$

which assumes that the maximum offset voltage in the positive and negative directions are symmetrical. Therefore, both resistor string matching and offset voltage affect the DNL of the converter.

## 29.2.2 The Two-Step Flash ADC

Another type of Flash converter is called the two-step Flash converter or a parallel, feed-forward ADC [7–10]. The basic block diagram of a two-step converter is seen in Fig. 29.26. The converter is separated into two complete Flash ADCs with feed-forward circuitry. The first converter generates a rough estimate of the value of the input, and the second converter performs a fine conversion. The advantages of this architecture are that the number of comparators is greatly reduced from that of the Flash converter -



**Figure 29.26** Block diagram of a two-step Flash ADC.

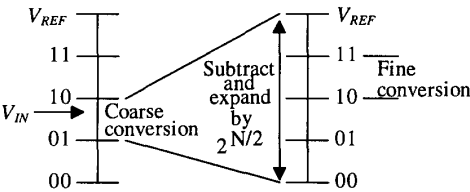
from  $2^N - 1$  comparators to  $2(2^{N/2} - 1)$  comparators. For example, an 8-bit Flash converter requires 255 comparators, while the two-step Flash requires only 30. The tradeoff is that the conversion process takes two steps instead of one, with the speed limited by the bandwidth and settling time required by the residue amplifier and the summer. The conversion process is as follows:

1. After the input is sampled, the most significant bits (MSBs) are converted by the first Flash ADC.
2. The result is then converted back to an analog voltage with the DAC and subtracted with the original input.
3. The result of the subtraction, known as the *residue*, is then multiplied by  $2^{N/2}$  and input into the second ADC. The multiplication not only allows the two ADCs to be identical, but also increases the quantum level of the signal input into the second ADC.
4. The second ADC produces the least significant bits through a Flash conversion.

Some architectures use the same set of comparators in order to perform both steps. The multiplication mentioned in step 3 can be eliminated if the second converter is designed to handle very small input signals. The accuracy of the two-step ADC is dependent primarily on the linearity of the first ADC.

Figure 29.27 illustrates the two-step nature of the converter. A more intuitive approach can be explained with this picture. The first conversion identifies the segment in which the analog voltage resides. This is also known as a *coarse conversion* of the MSBs. The results of the coarse conversion are then multiplied by  $2^{N/2}$  so that the segment within which  $V_{IN}$  resides will be scaled to the same reference as the first

conversion. The second conversion is known as the *fine conversion* and will generate the final LSBs using the same Flash approach. One can see why the accuracy of the first converter is so important. If the input value is close to the boundary between two coarse segments and the first ADC is unable to choose the correct coarse segment, then the second conversion will be completely erroneous. The following example further illustrates the two-step algorithm.



**Figure 29.27** Coarse and fine conversions using a two-step ADC.

**Example 29.12**

Assume that the two-step ADC shown in Fig. 29.26 has four bits of resolution. Make a table listing the MSBs,  $V_1$ ,  $V_2$ ,  $V_3$ , and the LSBs for  $V_{IN} = 2, 4, 9$ , and  $15$  V assuming that  $V_{REF} = 16$  V.

Since  $V_{REF}$  was conveniently made  $16$  V, each LSB will be  $1$  V. If  $V_{IN} = 2$  V, the output of the first 2-bit Flash converter will be  $00$  since  $V_{REF} = 16$  V and each resistor drops  $4$  V. The output of the 2-bit DAC,  $V_1$ , will therefore be  $0$ , resulting in  $V_2 = 2$  V. The multiplication of  $V_2$  by the  $4$  results in  $V_3 = 8$  V. Remember that each 2-bit Flash converter resembles that of Fig. 29.21. The thermometer code from the second Flash converter will be  $0011$ , which results in  $10$  as the LSBs. The other values can be calculated as seen in Fig. 29.28. ■

| $V_{IN}$ | $D_3D_2$ (MSBs) | $V_1$ | $V_2$ | $V_3$ | $D_1D_0$ (LSBs) |
|----------|-----------------|-------|-------|-------|-----------------|
| 2        | 00              | 0     | 2     | 8     | 10              |
| 4        | 01              | 4     | 0     | 0     | 00              |
| 9        | 10              | 8     | 1     | 4     | 01              |
| 15       | 11              | 12    | 3     | 12    | 11              |

**Figure 29.28** Output for the Flash ADC used in Ex. 29.12.

*Accuracy Issues Related to the Two-Step Flash Converters*

As stated previously, the overall accuracy of the converter is dependent on the first ADC. The second Flash must have only the accuracy of a stand-alone Flash converter. This means that if an 8-bit two-step Flash converter contains two 4-bit Flash converters, the second Flash needs only to have the resolution of a 4-bit Flash, which is not difficult

to achieve. However, the first 4-bit Flash must have the accuracy of an 8-bit Flash, meaning that the worst-case INL and DNL for the first bit Flash must be less than  $\pm\frac{1}{2}$  LSB for an 8-bit ADC. Thus, the resistor matching and comparators contained in the first ADC must possess the accuracy of the overall converter. Refer to Sec. 29.2.1 for derivations on INL and DNL for a Flash. The DAC must also be accurate to within the resolution of the ADC.

#### *Accuracy Issues Related to the Operational Amplifiers*

With the addition of the summer and the amplifier, other sources of accuracy errors are present in this converter. The summer and the amplifier must add and amplify the signal to within  $\pm\frac{1}{2}$  LSB of the ideal value. It is quite difficult to implement standard operational amplifiers within high-resolution data converters because of these accuracy requirements. The nonideal characteristics of the op-amp are well known and in many cases alone limit the accuracy of the data converter. In this case, the amplifier is required to multiply the residue signal by some factor of two. Although this may not seem difficult at first glance, a closer examination will reveal a dependency on the open-loop gain.

Suppose that the amplifier was being used in a 12-bit, two-step data converter. Remember that in order for a data converter to be  $N$ -bit accurate, the INL and DNL need to be kept below  $\pm\frac{1}{2}$  LSB and one-half of an LSB can be defined as

$$0.5 \text{ LSB} = \frac{V_{REF}}{2^{N+1}} \quad (29.54)$$

Since the output of the amplifier gets quantized to 6 bits, the amplifier would need to be 6-bit accurate, resulting in an accuracy of

$$\text{Accuracy} = \frac{1}{2^{6+1}} = \frac{1}{128} = 0.0078 = 0.78\% \quad (29.55)$$

And suppose that a feedback amplifier with a gain of 64, or  $2^{N/2}$ , is used as the residue amplifier. The gain would need to be within the following range:

$$63.5 \text{ V/V} < A_{CL} < 64.5 \text{ V/V} \quad (29.56)$$

where  $A_{CL}$  is the closed-loop gain of the amplifier. Already, one can see why operational amplifiers are not used for high-accuracy applications. Designing an op-amp with a high degree of gain accuracy is very difficult.

Generalizing this concept for an  $N$ -bit application will require knowledge of feedback theory discussed in Ch. 23. The closed-loop gain of the amplifier is expressed as

$$A_{CL} = \frac{v_o}{v_i} = \frac{A_{OL}}{1 + A_{OL}\beta} \quad (29.57)$$

where  $A_{OL}$  is the open-loop gain of the amplifier and  $\beta$  is the feedback factor. Also, from Ch. 23, it is known that as  $A_{OL}$  increases in value, the closed-loop gain,  $A_{CL}$ ,



approaches the value of  $\frac{1}{\beta}$ . Therefore, if it is assumed that the closed-loop gain of the amplifier will be equal to the ideal value of  $\frac{1}{\beta}$  plus some maximum deviation from the ideal,  $\Delta A$ , then,

$$A_{CL} = \frac{v_o}{v_i} = \frac{A_{OL}}{1 + A_{OL}\beta} = \frac{1}{\beta} + \Delta A \quad (29.58)$$

and the value of  $\Delta A$  will be equal to  $\frac{1}{2}$  LSB and  $\frac{1}{\beta}$  is the desired value of the closed-loop gain (usually some factor of  $2^N$ ). Therefore, the right two terms of Eq. (29.58) can be solved for the open-loop gain of the amplifier such that

$$|A_{OL}| = \frac{1}{\beta}(2^{N+1} + 1) \approx \frac{2^{N+1}}{\beta} \quad (29.59)$$

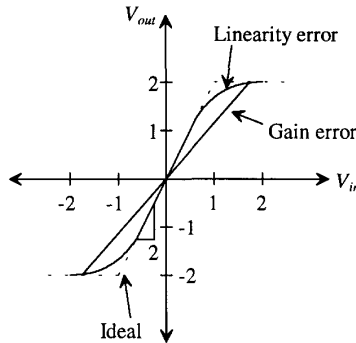
Equation (29.59) relates the open-loop gain of an amplifier to the resolution and closed-loop gain of the data converter. If the op-amp is to be used as a gain of 64, but is required to amplify signals with 6-bit accuracy, then the open-loop gain of the amplifier must be at least

$$|A_{OL}| \geq \frac{2^{N+1}}{\beta} = 128 \cdot 64 = 8,192 \text{ V/V} \quad (29.60)$$

This is certainly an achievable specification. However, notice that for every bit increase in resolution the open-loop gain requirement doubles. This is one reason why two-step Flash converters are limited in resolution to around 12 bits [9–12].

Linearity of the amplifier is another aspect of amplifier performance that must be considered when designing ADCs. The amplifier must be able to linearly amplify the input signal over an input voltage range to within  $\frac{1}{2}$  LSB of the number bits that its output is quantized. This means that if the amplifier is used in a 12-bit, two-step ADC, the slope of the transfer curve must be within 6-bit accuracy of the ideal. If the amplifier is not designed correctly, nonlinearity is introduced as devices in the amplifier go into nonsaturation. Harmonic distortion occurs, resulting in an error within the ADC.

Linearity is typically measured in terms of total harmonic distortion, or THD, (refer to Sec. 22.3). However, the transfer curve illustrates the limitation more effectively. Figure 29.29 shows a transfer curve of an op-amp with a gain of two. The ideal transfer curve is shown if the input range is known to be between  $-1$  and  $1$  V. The actual transfer curve shows nonlinearity introduced at both ends of the input range. In order for the amplifier to be  $N$ -bit accurate, the slope of the actual transfer curve may not vary from the ideal by more than the accuracy required at the output of the amplifier. Note also in Fig. 29.29 the subtle difference between a gain error and nonlinearity. However, a gain error is much less harmful to an ADC's performance than harmonic distortion.



**Figure 29.29** An op-amp transfer curve that distinguishes between gain error and linearity error.

### 29.2.3 The Pipeline ADC

After examining the two-step ADC, one might wonder whether there is such a converter as a three-step or four-step ADC. In actuality, one could divide the number of conversions into many steps. The pipeline ADC is an  $N$ -step converter, with 1-bit being converted per stage. Able to achieve high resolution (10–13 bits) at relatively fast speeds [11–15], the pipeline ADC consists of  $N$  stages connected in series (Fig. 29.30). Each stage contains a 1-bit ADC (a comparator), a sample-and-hold, a summer, and a gain of two amplifier. Each stage of the converter performs the following operation:

1. After the input signal has been sampled, compared it to  $\frac{V_{REF}}{2}$ . The output of each comparator is the bit conversion for that stage.
2. If  $V_{IN} > \frac{V_{REF}}{2}$  (comparator output is 1),  $\frac{V_{REF}}{2}$  is subtracted from the held signal and pass the result to the amplifier. If  $V_{IN} < \frac{V_{REF}}{2}$  (comparator output is 0), then pass the original input signal to the amplifier. The output of each stage in the converter is referred to as the *residue*.
3. Multiply the result of the summation by 2 and pass the result to the sample-and-hold of the next stage.

A main advantage of the pipeline converter is its high throughput. After an initial delay of  $N$  clock cycles, one conversion will be completed per clock cycle. While the residue of the first stage is being operated on by the second stage, the first stage is free to operate on the next samples. Each stage operates on the residue passed down from the previous stage, thereby allowing for fast conversions. The disadvantage is having the initial  $N$  clock cycle delay before the first digital output appears. The severity of this disadvantage is, of course, dependent on the application.

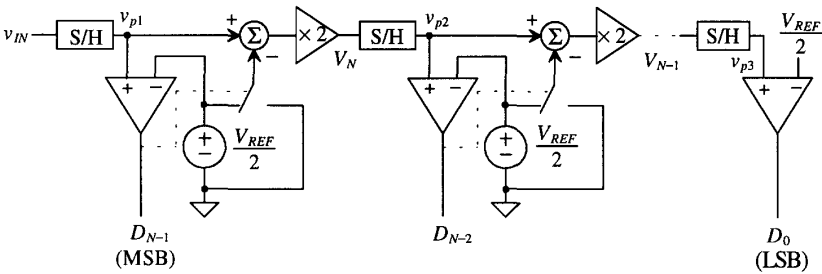


Figure 29.30 Block diagram of a pipeline ADC.

One interesting aspect of this converter is its dependency on the most significant stages for accuracy. A slight error in the first stage propagates through the converter and results in a much larger error at the end of the conversion. Each succeeding stage requires less accuracy than the one before, so special care must be taken when considering the first several stages.

**Example 29.13**

Assume the pipeline converter shown in Fig. 29.30 is a 3-bit converter. Analyze the conversion process by making a table of the following variables:  $D_2$ ,  $D_1$ ,  $D_0$ ,  $V_2$ ,  $V_1$ , for  $v_{IN} = 2, 3$ , and  $4.5$  V. Assume that  $V_{REF} = 5$  V. Where  $V_3$  is the residue voltage out of the first stage and  $V_2$  is the residue voltage out of the second stage.

The output of the first comparator,  $D_2 = 0$ , since  $v_{IN} < 2.5$  V. Since  $D_2 = 0$ ,  $V_3 = 2(2) = 4$  V. Passing this voltage down the pipeline, since  $V_3 > 2.5$  V,  $D_1 = 1$  and  $V_2$  becomes

$$V_1 = \left( V_2 - \frac{V_{REF}}{2} \right) \times 2 = 3 \text{ V}$$

The LSB,  $D_0 = 1$ , since  $V_2 > 2.5$  V, and the digital output corresponding to  $V_{IN} = 2$  V, is  $D_2D_1D_0 = 011$ . The actual digital outputs are simply the comparator outputs, and the data can be completed as seen in Fig. 29.31. ■

| $v_{IN}$ | $V_3$ (V) | $V_2$ (V) | Digital Out ( $D_2D_1D_0$ ) |
|----------|-----------|-----------|-----------------------------|
| 2.0      | 4.0       | 3.0       | 011                         |
| 3.0      | 1.0       | 2.0       | 100                         |
| 4.5      | 4.0       | 3.0       | 111                         |

Figure 29.31 Output for the pipeline ADC used in Ex. 29.13.

*Accuracy Issues Related to the Pipeline Converter*

The 1-bit per stage ADC can be analyzed by examining the switching point of each comparator for the ideal and nonideal case. Using Fig. 29.30, and assuming all the components are ideal, let  $v_{IN,1}$  represents the value of the input voltage when the first comparator switches. This occurs when

$$v_{IN,1} = \frac{1}{2} V_{REF} \quad (29.61)$$

The positive input voltage on the second comparator,  $v_{p2}$ , can be written in terms of the previous stage, or

$$v_{p2} = [v_{IN} - \frac{1}{2} \cdot D_{N-1} \cdot V_{REF}] \cdot 2 \quad (29.62)$$

where  $D_{N-1}$  is the MSB output from the first comparator and is either a 1 or a 0. The second comparator switches when  $v_{p2} = \frac{1}{2} V_{REF}$ . The value of  $v_{IN}$  at this point, denoted as  $v_{IN,2}$ , is

$$v_{IN,2} = \frac{1}{2} \cdot D_{N-1} \cdot V_{REF} + \frac{1}{4} V_{REF} \quad (29.63)$$

Continuing on in a similar manner, we can write the value of the voltage on the positive input of the third comparator in terms of the previous two stages as

$$v_{p3} = \left[ [v_{IN} - \frac{1}{2} \cdot D_{N-1} \cdot V_{REF}] \cdot 2 - [\frac{1}{2} \cdot D_{N-2} \cdot V_{REF}] \right] \cdot 2 \quad (29.64)$$

and the third comparator will switch when  $v_{p3} = \frac{1}{2} V_{REF}$ , which corresponds to the point at which  $v_{IN}$  becomes

$$v_{IN,3} = \frac{1}{2} \cdot D_{N-1} \cdot V_{REF} + \frac{1}{4} \cdot D_{N-2} \cdot V_{REF} + \frac{1}{8} V_{REF} \quad (29.65)$$

By now, a general trend can be recognized and the value of  $v_{IN}$  can be derived for the point at which the comparator of the  $N$ -th stage switches. This expression can be written as

$$v_{IN,N} = \frac{1}{2} \cdot D_{N-1} \cdot V_{REF} + \frac{1}{4} \cdot D_{N-2} \cdot V_{REF} + \frac{1}{8} \cdot D_{N-3} \cdot V_{REF} + \dots + \frac{1}{2^{N-1}} \cdot D_1 \cdot V_{REF} + \frac{1}{2^N} \cdot V_{REF} \quad (29.66)$$

Notice that the preceding equation does not include  $D_0$ . This is because  $D_0$  is the output of the  $N$ -th stage comparator.

Now that we have derived the switching points for the ideal case, the nonideal case can be considered. Only the major sources of error will be included in the analysis so as not to overwhelm the reader in the analysis. These include the comparator offset voltage,  $V_{COS,x}$ , and the sample-and-hold offset voltage,  $V_{SOS,x}$ . The variable,  $x$ , represents the number of the stage for which each of the errors is associated, and the "prime" notation will be used to distinguish between the ideal and nonideal case. The reader should also be aware that the offset voltages can be of either polarity. It will be assumed that all of the residue amplifiers will have the same gain, denoted as  $A$ .

The positive input to the first nonideal comparator,  $v'_{p1}$ , will include the offset from the first sample-and-hold, such that

$$v'_{p1} = v_{IN} + V_{SOS,1} \quad (29.67)$$

Now the first comparator will not switch until the voltage on the positive input overcomes the comparator offset as well. This occurs when

$$v'_{p1} = \frac{1}{2}V_{REF} + V_{COS,1} \quad (29.68)$$

Thus, equating Eqs. (29.67) and (29.68) and solving for the value of the input voltage when the switching occurs for the first comparator yields

$$v'_{IN,1} = \frac{1}{2}V_{REF} + V_{COS,1} - V_{SOS,1} \quad (29.69)$$

The input to the second comparator,  $v'_{p2}$ , can be written as

$$v'_{p2} = [v_{IN} + V_{SOS,1} - \frac{1}{2} \cdot D_{N-1} \cdot V_{REF}] \cdot A + V_{SOS,2} \quad (29.70)$$

and the value of input voltage at the point which the second comparator switches occurs when

$$v'_{IN,2} = \frac{1}{2} \cdot D_{N-1} \cdot V_{REF} + \frac{1}{2} \frac{V_{REF}}{A} - V_{SOS,1} - \frac{1}{A}(V_{SOS,2} - V_{COS,2}) \quad (29.71)$$

Continuing in the same manner, we can write the value of the input voltage that causes the third comparator to switch as

$$v'_{IN,3} = \frac{1}{2} \cdot D_{N-1} \cdot V_{REF} + \frac{1}{2} \cdot D_{N-2} \cdot \frac{V_{REF}}{A} - V_{SOS,1} - \frac{1}{A}V_{SOS,2} - \frac{1}{A^2}V_{SOS,3} - \frac{1}{A^2}[V_{COS,3} - \frac{1}{2}V_{REF}] \quad (29.72)$$

which can be generalized to the  $N$ -th switching point as

$$v'_{IN,N} = \frac{1}{2} \cdot D_{N-1} \cdot V_{REF} + \frac{1}{2} \cdot D_{N-2} \cdot \frac{V_{REF}}{A} + \dots + \frac{1}{2} \cdot D_1 \cdot \frac{V_{REF}}{A^{N-2}} + \frac{1}{2} \cdot \frac{V_{REF}}{A^{N-1}} + \frac{V_{COS,N}}{A^{N-1}} - \sum_{k=1}^N \frac{V_{SOS,k}}{A^{k-1}} \quad (29.73)$$

The INL can be calculated by subtracting switching point between the nonideal and ideal case. Therefore, the INL of the first stage is found by subtracting Eqs. (29.69) and (29.61).

$$INL_1 = v'_{IN,1} - v_{IN,1} = V_{COS,1} - V_{SOS,1} \quad (29.74)$$

The second stage INL is

$$INL_2 = v'_{IN,2} - v_{IN,2} = \frac{V_{REF}}{2} \left( \frac{1}{A} - \frac{1}{2} \right) - V_{SOS,1} - \frac{V_{SOS,2}}{A} + \frac{V_{COS,2}}{A} \quad (29.75)$$

and the INL for the  $N$ -th stage is

$$INL_N = \frac{1}{2} \cdot D_{N-2} \cdot V_{REF} \cdot \left( \frac{1}{A} - \frac{1}{2} \right) + \frac{1}{2} \cdot D_{N-3} \cdot V_{REF} \cdot \left( \frac{1}{A^2} - \frac{1}{4} \right) + \dots$$

$$+\frac{1}{2} \cdot D_1 \cdot V_{REF} \cdot \left( \frac{1}{A^{N-2}} - \frac{1}{2^{N-2}} \right) + \frac{1}{2} \cdot V_{REF} \cdot \left( \frac{1}{A^{N-1}} - \frac{1}{2^{N-1}} \right) + \frac{V_{COS,N}}{A^{N-1}} - \sum_{k=1}^N \frac{V_{SOS,K}}{A^{k-1}} \quad (29.76)$$

Equations (29.74) - (29.76) are very important to an understanding of the limitations of the pipeline ADC. Notice the importance of the comparator and summer offsets in Eq. (29.74). The worst-case addition of the offsets must be less than  $\frac{1}{2}$  LSB to keep the ADC  $N$ -bit accurate. The second stage is more dependent on the gain of the residue amplifier as seen in Eq. (29.75). The gain error discussed in the previous section plays an important role in determining the overall accuracy of the converter. Now examine the effects of the offsets on the INL of the  $N$ -th stage. In Eq. (29.76), both the comparator and summer offsets of the  $N$ -th stage (when  $k = N$ ) are divided by a large gain. Therefore, the latter stages in a pipeline ADC are not as critical to the accuracy as the first stages, and die area and power can be reduced by using less accurate designs for the least significant stages. The summation term in Eq. (29.76) also reveals that the summer offset of the first stage ( $k = 1$ ) has a large effect on the  $N$ -th stage. However, this point is inconsequential since  $V_{SOS,1}$  will have to be minimized to achieve  $N$ -bit accuracy for the first stage anyway. Typically, if the INL and DNL specifications can be made  $N$ -bit accurate in the first few stages, the latter stages will not adversely affect overall accuracy.

The DNL can be found by calculating the difference between the worst-case switching points and subtracting the ideal value for an LSB. As defined earlier, the worst-case will occur at midscale when the output switches from 0111...111 to 1000...000 as  $v_{IN}$  increase. Thus, the DNL is

$$DNL_{max} = v'_{IN,1} - v'_{IN,N} - \frac{V_{REF}}{2^N} \quad (29.77)$$

where  $v'_{IN,N}$  is calculated using Eq. (29.73) and assuming that  $D_{N-1}$  is a zero and all the other bits are ones. Plugging in Eqs. (29.69) and (29.73) into Eq. (29.77) yields

$$DNL_{max} = \frac{1}{2} V_{REF} \left( 1 - \sum_{k=1}^{N-1} \frac{1}{A^k} \right) + V_{COS,1} - \frac{V_{COS,N}}{A^{N-1}} + \sum_{k=2}^N \frac{V_{SOS,K}}{A^{k-1}} - \frac{V_{REF}}{2^N} \quad (29.78)$$

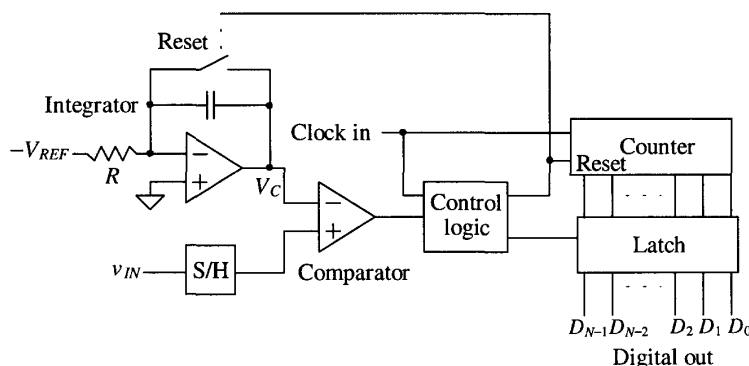
Again, the term that dominates this expression is the comparator offset associated with the first stage and the summer offset of the second stage. The entire expression in Eq. (29.78) must be less than  $\frac{1}{2}$  LSB for the ADC to have  $N$ -bit resolution.

#### 29.2.4 Integrating ADCs

Another type of ADC performs the conversion by integrating the input signal and correlating the integration time with a digital counter. Known as single- and dual-slope ADCs, these types of converters are used in high-resolution applications but have relatively slow conversions. However, they are very inexpensive to produce and are commonly found in slow-speed, cost-conscious applications.

### Single-Slope Architecture

Figure 29.32 illustrates the single-slope converter in block level form. A counter determines the number of clock pulses that are required before the integrated value of a reference voltage is equal to the sampled input signal. The number of clock pulses is proportional to the actual value of the input, and the output of the counter is the actual digital representation of the analog voltage.



**Figure 29.32** Block diagram of a single-slope ADC.

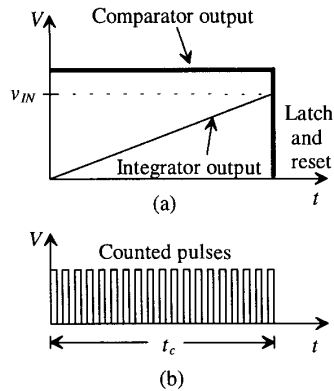
Since the reference is a DC voltage, the output of the integrator should start at zero and linearly increase with a slope that is dependent on the gain of the integrator. Notice that the reference voltage is defined as negative so that the output of the inverting integrator is positive. At the time when the output of the integrator surpasses the value of the S/H output, the comparator switches states, thus triggering the control logic to latch the value of the counter. The control logic also resets the system for the next sample. Figure 29.33 illustrates the behavior of the integrator output and the clock.

Note that if the input voltage is very small, the conversion time is very short, since the counter has to increment only a few times before the comparator latches the data. However, if the input voltage is at its full-scale value, the counter must increment to its maximum value of  $2^N$  clock cycles. Thus, the clock frequency must be many times faster than the bandwidth of the input signal. The conversion time,  $t_c$ , is dependent on the value of the input signal and can be described as

$$t_c = \frac{V_{IN}}{V_{REF}} \cdot 2^N \cdot T_{CLK} \quad (29.79)$$

where  $T_{CLK}$  is the period of the clock. The sampling rate is inversely proportional to the conversion time and can be written as

$$f_{Sample} = \frac{V_{REF}}{V_{IN} \cdot 2^N} \cdot f_{CLK} \quad (29.80)$$



**Figure 29.33** Single-slope ADC timing diagrams for (a) the comparator inputs and outputs and (b) the resulting counted pulses.

#### Example 29.14

Determine the clock frequency needed to form an 8-bit single-slope converter, if the analog signal bandwidth is 20 kHz.

Since the sampling rate required is 40 kHz, then the worst-case situation would occur for a full-scale input, in which event the integrator output would have to climb to its maximum value and the counter would increment  $2^N$  times during the corresponding 25  $\mu$ s period between samples. Therefore, the clock frequency would need to be  $2^N$  times faster than the sampling rate or 10.24 MHz. ■

#### Accuracy Issues Related to the Single-Slope ADC

Obviously, many potential error sources abound in this architecture. At the end of the conversion, the voltage across the integrating capacitor,  $V_C$ , (assuming no initial condition), will be

$$V_C = \frac{1}{C} \int_0^{t_c} \frac{V_{REF}}{R} dt = \frac{V_{REF} \cdot t_c}{RC} \quad (29.81)$$

where  $t_c$  is the conversion time. Plugging Eq. (29.79) into Eq. (29.81) yields

$$V_C = \frac{2^N \cdot T_{CLK} \cdot v_{IN}}{RC} = \frac{2^N \cdot v_{IN}}{f_{CLK} \cdot RC} \quad (29.82)$$

Equation (29.82) is a revealing one in that the final voltage on the integrator output is dependent not only on the value of the input voltage, which is to be expected, but also on the value of  $R$ ,  $C$ , and  $f_{CLK}$ . Therefore, any nonideal effects affecting these values will have an influence on the accuracy of the integrator output from sample to sample. For example, if an integrated diffused-resistor is used, then the voltage coefficient of the



resistor could limit the accuracy, since the resistor will be effectively nonlinear. Similarly, the capacitor may have charge leakage or aging effects associated with it. Also, any jitter in the clock will affect the overall accuracy. The integrator must have a linear slope to within the accuracy of the converter, which is dependent on the specifications of the op-amp (open-loop gain, settling time, offset, etc.) and must be considered accordingly.

Offset voltages on the comparator, the S/H, or the integrator will result in additional or fewer clock pulses, depending on the polarity of the offset. A delay also exists from the time that the inputs to the comparator are equal and the time that the output of the counter is actually latched. The reference voltage must also stay constant to within the accuracy of the converter.

#### *Dual-Slope Architecture*

A slightly more sophisticated design known as the dual-slope integrating ADC (Fig. 29.34) eliminates most of the problems encountered when using the single-slope converter. Here, two integrations are performed, one on the input signal and one on  $V_{REF}$ . The input voltage in this case is assumed to be negative, so that the output of the inverting integrator results in a positive slope during the first integration. Figure 29.35 illustrates the behavior for two separate samples. The first integration is of fixed length, dictated by the counter, in which the sample-and-held signal is integrated, resulting in the first slope. After the counter overflows and is reset, the reference voltage is connected to the input of the integrator. Since  $v_{IN}$  was negative and the reference voltage is positive, the inverting integrator output will begin discharging back down to zero at a constant slope. A counter again measures the amount of time for the integrator to discharge, thus generating the digital output.

For this figure, a 3-bit ADC is being used. Thus, the first integration period continues until the beginning of the eighth ( $2^3$ ) clock pulse, which corresponds to the counter's overflow bit. Note that the integrator's output corresponding to  $V_B$  is twice the value of the output corresponding to  $V_A$ . Thus, it requires twice as many clock pulses for the integrator to discharge back to zero from  $V_B$  than from  $V_A$ . The output of the counter at  $t_A$  is three or 011, while the counter output at  $t_B$  is twice that value or six (110) and the quantization is complete.

Notice that the first slope varies according to the value of the input signal, while the second slope, dependent only on  $V_{REF}$ , is constant. Similarly, the time required to generate the first slope is constant, since it is limited by the size of the counter. However, the discharging period is variable and results in the digital representation of the input voltage.

#### *Accuracy Issues Related to the Dual-Slope ADC*

One may wonder how the dual-slope converter is an improvement over the single-slope architecture, since a significantly longer conversion time is required. The first integration period requires a full  $2^N$  clock cycle and cannot be decreased, because the

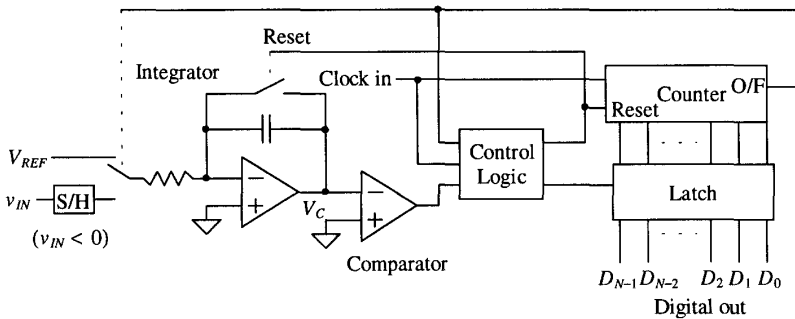


Figure 29.34 Block diagram of a dual-slope ADC.

second integration might require the full  $2^N$  clock cycles to discharge if the maximum value of  $v_{IN}$  is being converted. However, the dual slope is the preferred architecture because the same integrator and clock are used to produce both slopes. Therefore, any nonidealities will essentially be canceled. For example, assuming that the S/H is ideal, the gain of the integrator at the end of the first integration period,  $T_1$ , becomes

$$V_C = -\frac{1}{C} \int_0^{T_1} \frac{v_{IN}}{R} dt = \frac{|v_{IN}| \cdot T_1}{RC} \tag{29.83}$$

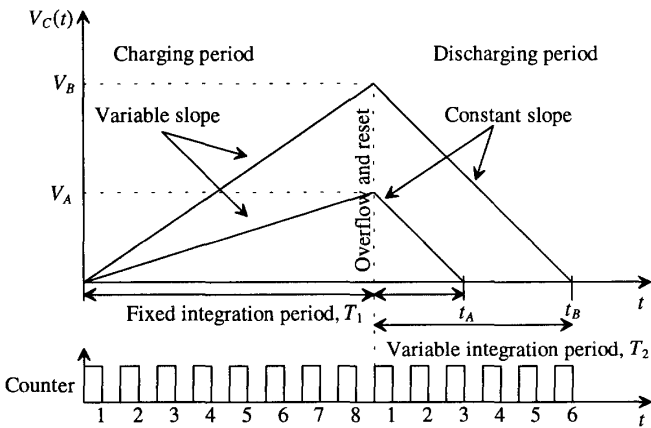


Figure 29.35 Integration periods and counter output for two separate samples of a 3-bit dual-slope ADC.

The output at the end of  $T_1$  is positive since the input voltage is considered to be negative and the integrator is inverting. After the clock has been reset, the discharging commences, with the initial condition defined by the value of the integrator output at the end of the charging period, or

$$V_C = \frac{|v_{IN}| \cdot T_1}{RC} - \frac{1}{C} \int_0^{T_2} \frac{V_{REF}}{R} dt \quad (29.84)$$

Once the value of the integrator output,  $V_C$ , reaches zero volts, Eq. (29.84) becomes

$$V_C = \frac{|v_{IN}| \cdot T_1}{RC} - \frac{V_{REF} \cdot T_2}{RC} = 0 \quad (29.85)$$

or,

$$|v_{IN}| \cdot T_1 = V_{REF} \cdot T_2 \quad (29.86)$$

At the end of the conversion, the dependencies on  $R$  and  $C$  have canceled out. Since we also know that the counter increments  $2^N$  times at time,  $T_1$ , and the counter increments  $D$  times at time,  $T_2$ , Eq. (29.86) can be rewritten as

$$\frac{D}{2^N} = \frac{|v_{IN}|}{V_{REF}} \quad (29.87)$$

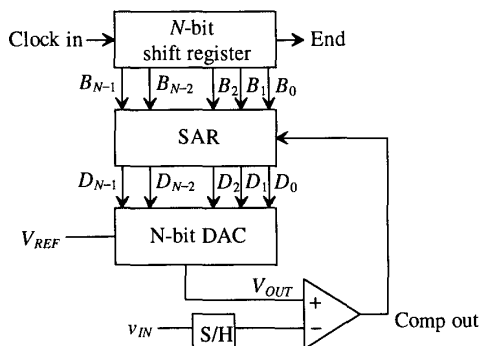
where  $D$  is the counter output that is actually the digital representation of the input voltage. Thus, it can be written that the ratio of the input voltage and the reference voltage is proportional to the ratio of the binary value of the digital word,  $D$ , and  $2^N$ . Therefore, since the same clock pulse is responsible for the charging and discharging times, any irregularities will also cancel out.

### 29.2.5 The Successive Approximation ADC

The successive approximation converter basically performs a binary search through all possible quantization levels before converging on the final digital answer. The block diagram is seen in Fig. 29.36. An  $N$ -bit register controls the timing of the conversion where  $N$  is the resolution of the ADC.  $V_{IN}$  is sampled and compared to the output of the DAC. The comparator output controls the direction of the binary search, and the output of the successive approximation register (SAR) is the actual digital conversion. The successive approximation algorithm is as follows.

1. A 1 is applied to the input to the shift register. For each bit converted, the 1 is shifted to the right 1-bit position.  $B_{N-1} = 1$  and  $B_{N-2}$  through  $B_0 = 0$ .
2. The MSB of the SAR,  $D_{N-1}$ , is initially set to 1, while the remaining bits,  $D_{N-2}$  through  $D_0$ , are set to 0.
3. Since the SAR output controls the DAC and the SAR output is 100...0, the DAC output will be set to  $\frac{V_{REF}}{2}$ .

4. Next,  $v_{IN}$  is compared to  $\frac{V_{REF}}{2}$ . If  $\frac{V_{REF}}{2}$  is greater than  $v_{IN}$ , then the comparator output is a 1 and the comparator resets  $D_{N-1}$  to 0. If  $\frac{V_{REF}}{2}$  is less than  $v_{IN}$ , then the comparator output is a 0 and the  $D_{N-1}$  remains a 1.  $D_{N-1}$  is the actual MSB of the final digital output code.
5. The 1 applied to the shift register is then shifted by one position so that  $B_{N-2} = 1$ , while the remaining bits are all 0.
6.  $D_{N-2}$  is set to a 1,  $D_{N-3}$  through  $D_0$  remain 0, while  $D_{N-1}$  remains the value from the MSB conversion. The output of the DAC will now either equal  $\frac{V_{REF}}{4}$  (if  $D_{N-1} = 0$ ) or  $\frac{3V_{REF}}{4}$  (if  $D_{N-1} = 1$ ).
7. Next,  $v_{IN}$  is compared to the output of the DAC. If the DAC output is greater than  $v_{IN}$ , the comparator the  $D_{N-2}$  is reset to 0. If  $v_{IN}$  is less than the DAC output,  $D_{N-2}$  remains a 1.
8. The process repeats until the output of the DAC converges to the value of  $v_{IN}$  within the resolution of the converter.

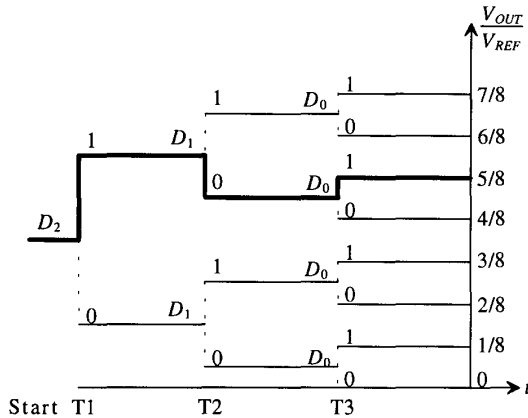


**Figure 29.36** Block diagram of the successive approximation ADC.

Figure 29.37 shows an example of the binary search nature of the converter. The bolded line shows the path of the conversion for 101, corresponding to  $\frac{5}{8} V_{REF}$ . All possible quantization levels are represented in the binary tree. With each bit decided, the search space decreases by one-half until the correct answer is converged upon.

### Example 29.15

Perform the operation of a 3-bit successive approximation ADC similar to Fig. 29.36 with  $V_{REF} = 8$ . Make a table that consists of  $D_2D_1D_0$ ,  $B_2B_1B_0$ ,  $V_{OUT}$  (the output from the DAC) and the comparator output, which shows the binary search algorithm of the converter for  $v_{IN} = 5.5$  V and 2.5 V.



**Figure 29.37** Binary search performed by a 3-bit successive approximation ADC for  $D=101$ .

We will designate  $D_2'D_1'D_0'$  as the initial output of the SAR before the comparator makes its decision. The final value is designated as  $D_2D_1D_0$ . Notice that if the comparator is a 1,  $D_2'D_1'D_0'$  differs from  $D_2D_1D_0$ , but if the comparator outputs a 0, then  $D_2'D_1'D_0' = D_2D_1D_0$ . The output of the shift register is designated as  $B_2B_1B_0$ .

Following the algorithm discussed previously, initially  $v_{IN} = 5.5$  V and is compared with 4 V. Since the comparator output is 0, the MSB remains a 1. The next bit is examined, and the output of the DAC is now 6 V. Since  $V_{OUT} > v_{IN}$ , the comparator output is 1, which resets the current SAR bit,  $D_1$ , to a 0 at the end of period T2. Lastly, the LSB is examined, and  $v_{IN}$  is compared with 5 V. Since  $v_{IN} > V_{OUT}$ , the comparator output is a 0, and the current SAR bit,  $D_0$ , remains a 1. The results can be examined in Fig. 29.38a. The final value for  $D_2D_1D_0$  is 101, which is what is expected considering that 101 in binary is equivalent to  $5_{10}$ . Figure 29.38b shows the data for the ADC using  $v_{IN} = 2.5$  V. The final value for  $v_{IN} = 2.5$  is 010, which again is what is expected for 3-bit resolution. ■

The successive approximation ADC is one of the most popular architectures used today. The simplicity of the design allows for both high speed and high resolution while maintaining relatively small area. The limit to the ADC's accuracy is dependent mainly on the accuracy of the DAC. If the DAC does not produce the correct analog voltage with which to compare the input voltage, the entire converter output will contain an error. Referring again to Fig. 29.37, we can see that if a wrong decision is made early, a massive error will result as the converter attempts to search for the correct quantization level in the wrong half of the binary tree.

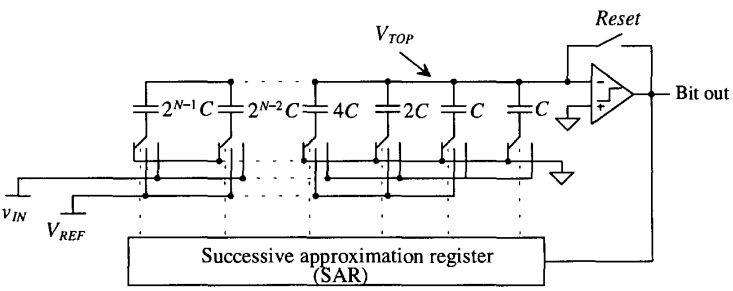
| Step | $v_{IN}$ | $B_2B_1B_0$ | $D_2'D_1'D_0'$ | $V_{OUT}$                        | Comp Out | $D_2D_1D_0$ |
|------|----------|-------------|----------------|----------------------------------|----------|-------------|
| T1   | 5.5      | 100         | 100            | $1/2 V_{REF} = 4 \text{ V}$      | 0        | 100         |
| T2   | 5.5      | 010         | 110            | $(1/2+1/4)V_{REF} = 6 \text{ V}$ | 1        | 100         |
| T3   | 5.5      | 001         | 101            | $(1/2+1/8)V_{REF} = 5 \text{ V}$ | 0        | 101         |
| (a)  |          |             |                |                                  |          |             |
| Step | $v_{IN}$ | $B_2B_1B_0$ | $D_2'D_1'D_0'$ | $V_{OUT}$                        | Comp Out | $D_2D_1D_0$ |
| T1   | 2.5      | 100         | 100            | $1/2 V_{REF} = 4 \text{ V}$      | 1        | 000         |
| T2   | 2.5      | 010         | 010            | $1/4 V_{REF} = 2 \text{ V}$      | 0        | 010         |
| T3   | 2.5      | 001         | 011            | $(1/4+1/8)V_{REF} = 3 \text{ V}$ | 1        | 010         |
| (b)  |          |             |                |                                  |          |             |

**Figure 29.38** Results from the 3-bit successive approximation ADC using (a)  $v_{IN} = 5.5$  and (b) 2.5 V.

*The Charge-Redistribution Successive Approximation ADC*

One of the most popular types of successive approximation architectures uses the binary-weighted capacitor array (analyzed in Sec. 29.1.5) as its DAC. Called a charge redistribution successive approximation ADC [2,16,17], this converter samples the input signal and then performs the binary search based on the amount of charge on each of the DAC capacitors. Figure 29.39 shows an  $N$ -bit architecture. A comparator has replaced the unity gain buffer used in the DAC architecture. The binary-weighted capacitor array also samples the input voltage, so no external sample-and-hold is needed.

The conversion process begins by discharging capacitor array, via the reset switch. Although this may appear to be an insignificant action, the converter is also performing automatic offset cancellation. Once the reset switch is closed, the comparator acts as a unity gain buffer. Thus, the capacitor array charges to the offset



**Figure 29.39** A charge redistribution ADC using a binary-weighted capacitor array DAC.

voltage of the comparator. This requires that the comparator be designed so as to be unity gain stable, which means that internal compensation may have to be switched in during the reset period. Next, the input voltage,  $v_{IN}$ , is sampled onto the capacitor array. The reset switch is still closed, for the top plate of the capacitor array needs to be connected to virtual ground of the unity gain buffer. The equivalent circuit is seen in Fig. 29.40a.

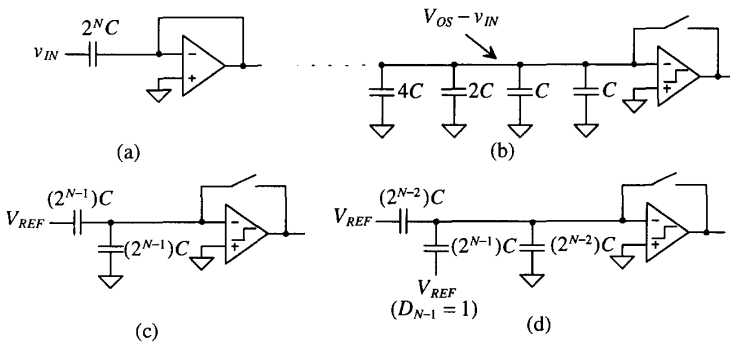
The reset switch is then opened, and the bottom plates of each capacitor in the array are switched to ground, so that the voltage appearing at the top plate of the array is now  $V_{OS} - v_{IN}$  (Fig. 29.40b). The conversion process begins by switching the bottom plate of the MSB capacitor to  $V_{REF}$  (Fig. 29.40c). If the output of the comparator is high, the bottom plate of the MSB capacitor remains connected to  $V_{REF}$ . If the comparator output is low, the bottom plate of the MSB is connected back to ground. The output of the comparator is  $D_{N-1}$ . The voltage at the top of the capacitor array,  $V_{TOP}$ , is now

$$V_{TOP} = -v_{IN} + V_{OS} + D_{N-1} \cdot \frac{V_{REF}}{2} \quad (29.88)$$

The next largest capacitor is tested in the same manner as seen in Fig. 29.40d. The voltage at the top plate of the capacitor after the second capacitor is tested becomes

$$V_{TOP} = -v_{IN} + V_{OS} + D_{N-1} \cdot \frac{V_{REF}}{2} + D_{N-2} \cdot \frac{V_{REF}}{4} \quad (29.89)$$

The conversion process continues on with the remaining capacitors so that the voltage on the top plate of the array,  $V_{TOP}$ , converges to the value of the offset voltage,  $V_{OS}$  (within the resolution of the converter), or



**Figure 29.40** The charge redistribution process: (a) Sampling the input while autozeroing the offset, (b) the voltage at the top plate after sampling, (c) the equivalent circuit while converting the MSB, and (d) the equivalent circuit while converting the next largest capacitor with the MSB result equal to one.

$$V_{TOP} = -V_{IN} + V_{OS} + D_{N-1} \cdot \frac{V_{REF}}{2} + D_{N-2} \cdot \frac{V_{REF}}{4} + \dots + D_1 \cdot \frac{V_{REF}}{2^{N-2}} + D_0 \cdot \frac{V_{REF}}{2^{N-1}} \approx V_{OS} \quad (29.90)$$

Note that the initial charge stored on the capacitor array is now redistributed onto only those capacitors that have their bottom plates connected to  $V_{REF}$ .

#### *Accuracy Issues Related to the Charge Redistribution Successive Approximation ADC*

Obviously, the limitation of this architecture is the capacitor matching. The mismatch is analyzed in the same manner as the binary-weighted current source array of Sec. 29.1.4. Thus, substituting the value of the unit capacitance,  $C$ , for the value of the unit current source,  $I$ , and using Eqs. (29.27) – (29.30),

$$|INL|_{max} = 2^{N-1}(C + |\Delta C|_{max,INL}) - 2^{N-1} \cdot C = 2^{N-1} \cdot |\Delta C|_{max,INL} \quad (29.91)$$

where the maximum  $\Delta C$  that will result in an INL that is less than  $\frac{1}{2}$  LSB is

$$|\Delta C|_{max,INL} = \frac{0.5C}{2^{N-1}} = \frac{C}{2^N} \quad (29.92)$$

and DNL is defined by

$$DNL_{max} = (2^N - 1) \cdot |\Delta C|_{max,DNL} \quad (29.93)$$

with the maximum  $\Delta C$  which will result in a DNL less than  $\frac{1}{2}$  LSB being

$$|\Delta C|_{max,DNL} = \frac{0.5C}{2^N - 1} = \frac{C}{2^{N+1} - 2} \quad (29.94)$$

### **29.2.6 The Oversampling ADC**

ADCs can be separated into two categories depending on the rate of sampling. The first category samples the input at the Nyquist rate, or  $f_N = 2F$  where  $F$  is the bandwidth of the signal and  $f_N$  is the sampling rate. The second type samples the signal at a rate much higher than the signal bandwidth. This type of converter is called an oversampling converter. Traditionally, successive approximation or dual-slope converters are used when high resolution is desired. However, trimming is required when attempting to achieve higher accuracy. Dual-slope converters require high-speed, high-accuracy integrators that are only available using a high  $f_T$  bipolar process. Having to design a high-precision sample-and-hold is another factor that limits the realization of a high-resolution ADC using these architectures.

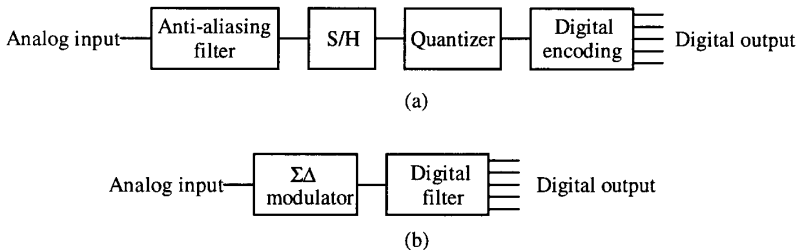
The oversampling ADC [18-20] is able to achieve much higher resolution than the Nyquist rate converters. This is because digital signal processing techniques are used in place of complex and precise analog components. The accuracy of the converter does not depend on the component matching, precise sample-and-hold circuitry, or trimming, and only a small amount of analog circuitry is required. Switched-capacitor implementations are easily achieved, and, as a result of the high sampling rate, only simplistic anti-aliasing circuitry needs to be used. However, because of the amount of



time required to sample the input signal, the throughput is considerably less than the Nyquist rate ADCs.

#### *Differences in Nyquist Rate and Oversampling ADCs*

The typical process used in analog-to-digital conversion is seen in Fig. 29.41a, while the block diagram for the oversampling ADC is seen in Fig. 29.41b. After filtering the signal to help minimize anti-aliasing effects, the signal is sampled, quantized, and encoded or decoded using simple digital logic to provide the digital data in the proper format. When using oversampling ADCs, little if any, anti-alias filtering is needed, no dedicated S/H is required, the quantization is performed with a modulator, and the encoding usually takes the form of a digital filter.

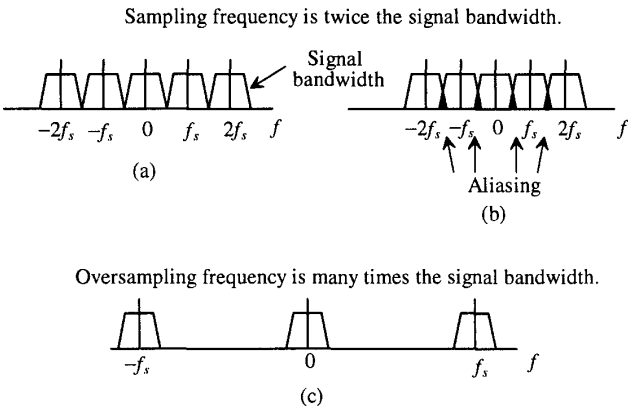


**Figure 29.41** Typical block diagram for (a) Nyquist rate converters and (b) oversampling ADCs.

Since the oversampling converter samples at many times the signal bandwidth, aliasing is not a serious problem. A discussion of the frequency characteristics of aliasing was presented in Ch. 28. Figure 29.42a shows that when using Nyquist rate converters, a sampled signal in the frequency domain appears as a series of band-limited signals at multiples of the sampling frequency (see Fig. 28.26 for more details). As the sampling frequency decreases, the frequency spectra begin to overlap, and aliasing (Fig. 29.42b) occurs. Complex, "brickwall" filters are needed to correct the problem.

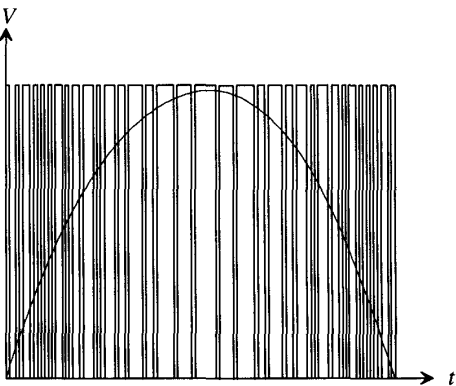
For oversampled ADCs, aliasing becomes much less of a factor. Since the sampling rate is much greater than the bandwidth of the signal, the frequency domain representation shows that the spectra are widely spaced, as seen in Fig. 29.42c. Therefore, overlapping of the spectra, and thus aliasing, will not occur, and only simple, first-order filters are required.

Oversampling converters typically employ switched-capacitor circuits and therefore do not need sample-and-hold circuits. The output of the modulator is a pulse-density modulated signal that represents the average of the input signal. The modulator is able to construct these pulses in real time, and so it is not necessary to hold the input value and perform the conversion.



**Figure 29.42** Frequency domain for (a) Nyquist rate converters (b) the aliasing that occurs; and (c) an oversampling converter.

As stated previously, the modulator actually provides the quantization in the form of a pulse-density modulated signal. Referred to as sigma-delta ( $\Sigma\Delta$ ) or delta-sigma ( $\Delta\Sigma$ ) modulation, the density of the pulses represents the average value of the signal over a specific period. Figure 29.43 illustrates the output of the modulator for the positive half of a sine wave input. Note that for the peak of the sine wave, most of the pulses are high. As the sine wave decreases in value, the pulses become distributed between high and low according to the sine wave value.



**Figure 29.43** Pulse-density output from a sigma-delta modulator for a sine wave input.

If the frequency of the sine wave represented the highest frequency component of the input signal, a Nyquist rate converter would take only two samples. The oversampling converter, however, may take hundreds of samples over the same period to produce this pulse-density signal.

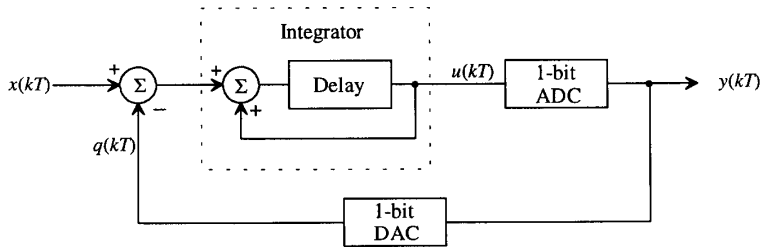
Digital signal processing is then utilized, which has two purposes: to filter any out-of-band quantization noise and to attenuate any spurious out-of-band signals. The output of the filter is then downsampled to the Nyquist rate so that the resulting output of the ADC is the digital data, which represents the average value of the analog voltage over the oversampling period. The effective resolution of oversampling converters is determined by the values of signal-to-noise ratio and dynamic range obtained.

*The First-Order  $\Sigma\Delta$  Modulator*

Now that the basic function of the  $\Sigma\Delta$  modulator has been described, it would be useful to examine its inner workings and determine why  $\Sigma\Delta$  modulation is so beneficial for generating high-resolution data. A basic first-order  $\Sigma\Delta$  modulator can be seen in Fig. 29.44. Here, an integrator and a 1-bit ADC are in the forward path, and a 1-bit DAC is in the feedback path of a single-feedback loop system. The variables labeled are in terms of time,  $T$ , which is the inverse of the sampling frequency and  $k$ , which is an integer. The 1-bit ADC is simply a comparator that converts an analog signal into either a high or a low. The 1-bit DAC uses the comparator output to determine if  $+V_{REF}$  or  $-V_{REF}$  is summed with the input.

While the benefits of  $\Sigma\Delta$  modulation are not obvious, a simple derivation of the output,  $y(kT)$ , will illuminate its distinct advantages. The output of the integrator,  $u(kT)$ , can be described as

$$u(kT) = x(kT - T) - q(kT - T) + u(kT - T) \tag{29.95}$$



**Figure 29.44** A first-order sigma-delta modulator.

where,  $x(kT - T) - q(kT - T)$  is equal to the integrator's previous input, and  $u(kT - T)$  is its previous output. The quantization error for the 1-bit ADC, as discussed in Ch. 28, is again defined as the difference between its output and input such that

$$Q_e(kT) = y(kT) - u(kT) \quad (29.96)$$

Plugging Eq. (29.95) into Eq. (29.96), the output response,  $y(kT)$  is

$$y(kT) = Q_e(kT) + x(kT - T) - q(kT - T) + u(kT - T) \quad (29.97)$$

An ideal 1-bit DAC has the following characteristic; if the input,  $y(kT) = 0$ , the output,  $q(kT) = -V_{REF}$ , and if  $y(kT) = 1$ , then  $q(kT) = V_{REF}$ . In reality, a 1-bit DAC consists of a couple of switches connecting  $V_{REF}$  or  $-V_{REF}$  to a common node, so it is not difficult to assume that the DAC is ideal. Therefore,

$$y(kT) = q(kT) \quad (29.98)$$

Utilizing Eq. (29.96) and Eq. (29.97), we find that Eq. (29.98) becomes

$$y(kT) = x(kT - T) + Q_e(kT) - Q_e(kT - T) \quad (29.99)$$

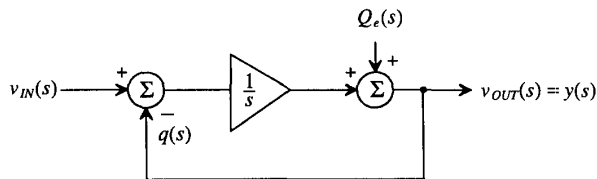
Therefore, the output of the modulator consists of a quantized value of the input signal delayed by one sample period, plus a differencing of the quantization error between the present and previous values. Thus, the real power of  $\Sigma\Delta$  modulation is that the quantization noise,  $Q_e$ , cancels itself out to the first order.

A frequency domain example will further illuminate this important fact. Suppose that the first-order modulator can be modeled in the  $s$  domain as seen in Fig. 29.45, with an ideal integrator represented with transfer function of  $\frac{1}{s}$ , the 1-bit ADC is modeled as a simple error source,  $Q_e(s)$ , and again the DAC is considered to be ideal, such that  $y(s)$  is equal to  $q(s)$ . It is also assumed that the bandwidth of the input signal is much less than the bandwidth of the modulator. Therefore, using simple feedback theory,  $v_{OUT}(s)$  becomes

$$v_{OUT}(s) = Q_e(s) + \frac{1}{s} \cdot [v_{IN}(s) - v_{OUT}(s)] \quad (29.100)$$

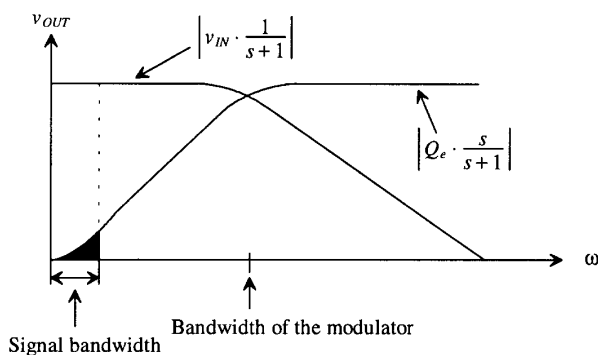
and solving for  $v_{OUT}$  yields,

$$v_{OUT}(s) = Q_e(s) \cdot \frac{s}{s+1} + v_{IN}(s) \frac{1}{s+1} \quad (29.101)$$



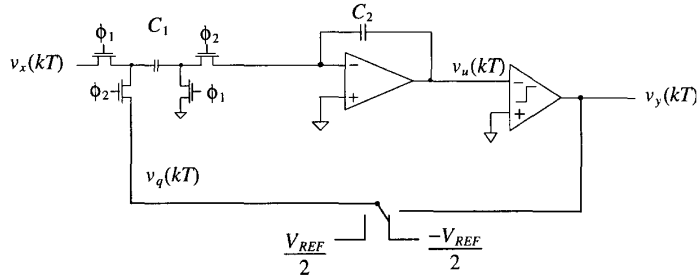
**Figure 29.45** A frequency domain model for the first-order sigma-delta modulator.

Note that the transfer function from  $v_{IN}$  to  $v_{OUT}$  follows that of a low-pass filter and that the transfer function of the quantization noise follows that of a high-pass filter. Plotted together in Fig. 29.46, it is seen that in the region where the signal is of interest, the noise has a small value while the signal has a high gain, and that at higher frequencies, beyond the bandwidth of the signal, the noise increases. The modulator has essentially pushed the power of the noise out of the bandwidth of the signal. This high-pass characteristic is known as *noise shaping* and is a powerful concept used within oversampling ADCs. Low-pass filtering will then be performed by the digital filter in order to remove all of the out-of-band quantization noise, which then permits the signal to be downsampled to yield the final high-resolution output.



**Figure 29.46** Frequency response of the first-order sigma-delta modulator.

As the  $\Sigma\Delta$  modulator is generating the pulse-density modulated output, it is interesting to examine the mechanics occurring in the loop, which result in an average of the input. An actual  $\Sigma\Delta$  modulator might resemble Fig. 29.47. A switched-capacitor integrator (see Sec. 27.2) provides the summing as well as the delay needed. The 1-bit ADC is a simple comparator, and 1-bit DAC is simply two voltage-controlled switches that select either  $\frac{1}{2}V_{REF}$  or  $-\frac{1}{2}V_{REF}$  to be summed with the input. A latched comparator is used to provide the necessary loop delay. Notice that the variables used are voltage representations of the variables used in Fig. 29.44. Remember that the function of the integrator is to accumulate differences between the input signal and the output of the DAC. If it is assumed that the input,  $v_i(kT)$ , is a positive DC voltage, then the output of the integrator should increase. However, the feedback mechanism is such that the 1-bit ADC (the comparator) has a low output if the integrator output,  $v_u(kT)$ , is positive. Thus,  $\frac{1}{2}V_{REF}$  appears at the output of the DAC and is subtracted from the input, and the integrator output is driven back toward zero. The opposite occurs when  $v_i(kT)$  is negative such that the integrator output is always driven toward zero by the feedback mechanism. An example will illustrate the operation of the modulator in more detail.



**Figure 29.47** Implementation of a first-order sigma-delta modulator using a switched capacitor integrator.

### Example 29.16

Using a general first-order  $\Sigma\Delta$  modulator, assume that the input to the modulator,  $v_x(kT)$  is a positive DC voltage of 0.4 V. Show the values of each variable around the  $\Sigma\Delta$  modulator loop and prove that the overall average output of the DAC approaches 0.4 V after 10 cycles. Assume that the DAC output is  $\pm 1$  V, and that the integrator output has a unity gain with an initial output voltage of 0.1 V, and that the comparator output is either  $\pm 1$  V.

The present integrator output will be equal to the sum of the previous integrator output and the previous integrator input. Therefore, Eq. (29.95) becomes

$$v_u(kT) = v_u(kT - T) + v_q(kT - T) \quad (29.102)$$

where

$$v_a(kT) = v_x(kT) - v_q(kT) \quad (29.103)$$

and the quantizing error,  $Q_e(kT)$ , is defined by Eqs. (29.96) and (29.98) as

$$Q_e(kT) = v_q(kT) - v_u(kT) \quad (29.104)$$

The initial conditions define the values of the variable for  $k = 0$ . The output of the integrator is given to be 0.1 V. Thus, the ADC output is low, the DAC output is  $V_{REF}$ , and the output of the summer,  $v_u(0)$ , is  $0.4 - V_{REF} = -0.6$  V.

The output for  $k = 1$  begins again with the integrator output. Using Eq. (29.102),  $v_u(kT)$  becomes

$$v_u(T) = 0.1 + (-0.6) = -0.5 \text{ V}$$

Since the output of the integrator is negative, the output of the comparator is positive and  $-V_{REF}$  is subtracted from 0.4 to arrive at the value for  $v_a(T)$ .

Continuing in the same manner and using the above equations, we note the voltages for each cycle in Fig. 29.48. After 10 cycles through the modulator, the average value of  $v_q(kT)$  becomes,

$$\overline{v_q(kT)} = \frac{7-3}{10} = 0.4 \text{ V}$$

Notice that the behavior of  $\overline{v_q(kT)}$  swings around the desired value 0.4 V. If we were to continue computing values, as  $k$  increases, the amount that  $v_q(kT)$  differs from 0.4 V would decrease. Ideally, we could make the deviation of  $\overline{v_q(kT)}$  as small as desired by allowing the modulator to take as many samples as necessary to meet that accuracy. ■

| $k$ | $v_a(kT)$ | $v_u(kT)$ | $v_q(kT) = v_y(kT)$ | $Q_e(kT)$ | $\overline{v_q(kT)}$ |
|-----|-----------|-----------|---------------------|-----------|----------------------|
| 0   | -0.6      | 0.1       | 1.0                 | 0.9       | 1.0                  |
| 1   | 1.4       | -0.5      | -1.0                | -0.5      | 0                    |
| 2   | -0.6      | 0.9       | 1.0                 | 0.1       | 0.333                |
| 3   | -0.6      | 0.3       | 1.0                 | 0.7       | 0.50                 |
| 4   | 1.4       | -0.3      | -1.0                | -0.7      | 0.20                 |
| 5   | -0.6      | 1.1       | 1.0                 | -0.1      | 0.333                |
| 6   | -0.6      | 0.5       | 1.0                 | 0.5       | 0.429                |
| 7   | 1.4       | -0.1      | -1.0                | -0.9      | 0.25                 |
| 8   | -0.6      | 1.3       | 1.0                 | -0.3      | 0.333                |
| 9   | -0.6      | 0.7       | 1.0                 | 0.3       | 0.40                 |

**Figure 29.48** Data from the first-order  $\Sigma\Delta$  modulator.

One interesting note is to examine the effects of using a nonideal comparator. Suppose the integrator's output was smaller than the offset voltage of the comparator. A wrong decision would be made, causing  $v_y(kT)$  to be the opposite of the desired value. However, as  $k$  increases, this error is averaged out, and the modulator still converges on the correct answer. Therefore, the comparator does not have to be very accurate in its ability to distinguish between two voltages, in contrast to Nyquist rate comparators.

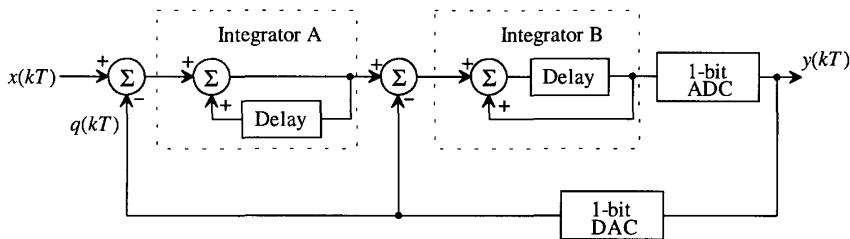
An interesting application of the first-order  $\Sigma\Delta$  modulator was reported in [20]. First-order  $\Sigma\Delta$  modulators were used to perform pixel-level A/D conversion within a CMOS image sensor. In most digital imaging systems, ADCs are used to convert the analog signal representing the light intensity gathered by a charge-coupled device (CCD). Either a single high-speed ADC is used to convert the entire signal off of the CCD, or a parallel array of lower performance ADCs is used with one ADC assigned to





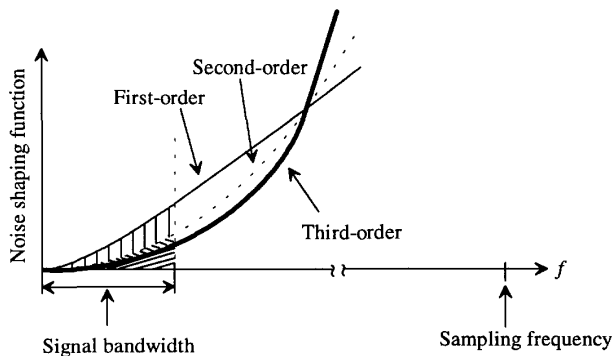
### The Higher Order $\Sigma\Delta$ Modulators

Higher order  $\Sigma\Delta$  modulators exist which provide a greater amount of noise shaping. A second-order  $\Sigma\Delta$  modulator can be seen in Fig. 29.50. A derivation of the second-order transfer function would reveal that the output contained a delayed version of the input plus a second-order differencing of the quantization noise,  $Q_e$  (see Problem 29.38). A third-order modulator would contain third-order differencing of the quantization and can be constructed by adding another integrator similar to integrator A into the system.



**Figure 29.50** A second-order sigma-delta modulator.

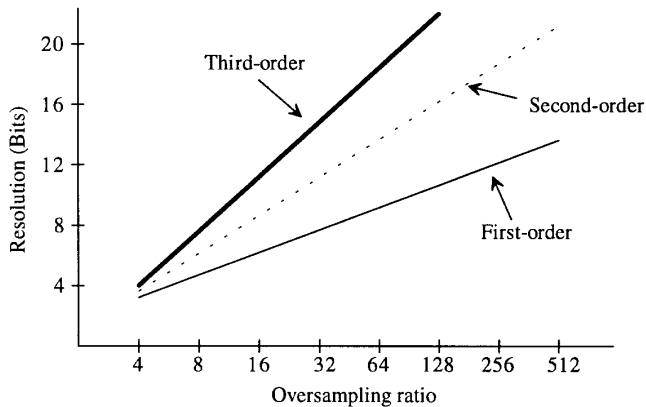
Figure 29.51 shows the noise-shaping functions of a first-, second-, and third-order modulator. The cross-hatched area under each of the curves represents the noise that remains in the signal bandwidth and is a magnified version of the blackened area of Fig. 29.46. As the order increases, notice that more of the noise is pushed out into the higher frequencies, thus decreasing the noise in the signal bandwidth. It should be reiterated that  $\Sigma\Delta$  modulators do not attenuate noise at all. In fact, they add quantization noise that is very large at high frequencies. But because almost all of the



**Figure 29.51** Noise shaping comparison of a first-, second-, and third-order modulator.

noise is out of the signal bandwidth, it can be easily filtered, leaving only a small portion within the signal bandwidth. This point is important because the  $\Sigma\Delta$  modulator should not be construed as a filtering circuit.

The resolution also increases as the order of the  $\Sigma\Delta$  modulator and the oversampling ratio increases, as seen in Fig. 29.52 [21]. Using a first-order modulator, one can expect an increase in dynamic range of 9 dB with every doubling of the oversampling ratio. This correlates to an approximate increase in resolution of 1.5 bits according to Eq. (28.28). The higher-order modulators have even greater gains in resolution as a 2.5-bit increase is attained with each doubling of the oversampling ratio using a second-order modulator, while the third-order modulator increases 3.5 bits.



**Figure 29.52** Comparison of first-, second-, and third-order modulators versus oversampling ratio and resolution.

One could essentially construct a high-order  $\Sigma\Delta$  modulator with many integrators. However, as with any system employing feedback, stability becomes a critical issue. The same holds true for the high-order  $\Sigma\Delta$  modulators. Several other topologies have been developed which can implement modulators in a cascaded fashion and are guaranteed to be stable [22, 23]. However, considerable matching requirements need to be overcome.

**REFERENCES**

- [1] R. L. Geiger, P. E. Allen, and N. R. Strader, *VLSI - Design Techniques for Analog and Digital Circuits*, McGraw-Hill Publishing Co., 1990.
- [2] R. E. Suarez, P. R. Gray, and D. A. Hodges, "All-MOS Charge Redistribution Analog-to-Digital Conversion Techniques - Part II," *IEEE Journal of Solid State Circuits*, Vol. 10, No. 6, pp. 379-385, December 1975.
- [3] J. Shyu, G. C. Temes, and F. Krummenacher, "Random Errors in MOS Capacitors and Current Sources," *IEEE Journal of Solid State Circuits*, Vol. 16, No. 6, pp. 948-955, December 1984.
- [4] M. J. M. Pelgrom et. al, "25-Ms/s 8-bit CMOS A/D Converter for Embedded Application," *IEEE Journal of Solid-State Circuits*, Vol. 29, No. 8, pp. 879-886, August 1994.
- [5] D. Choi et. al, "Analog Front-End Signal Processor for a 64 Mb/s PRML Hard-Disk Drive Channel," *IEEE Journal of Solid-State Circuits*, Vol. 29, No. 12, pp. 1596-1605, December 1994.
- [6] N. Shiwaku, "A Rail-to-Rail Video-band Full Nyquist 8-bit A/D Converter," *Proceedings of the 1991 Custom Integrated Circuits Conference*.
- [7] B. Razavi and B. A. Wooley, "A 12-b, 5-MSample/s Two-Step CMOS A/D Converter," *IEEE Journal of Solid State Circuits*, Vol. 27, No. 12, pp. 1667-1678, December 1992.
- [8] J. Dornberg, P. R. Gray, and D. A. Hodges, "A 10-bit, 5-Msample/s CMOS Two-Step Flash ADC," *IEEE Journal of Solid State Circuits*, Vol. 24, No. 2, pp. 241-249, April 1989.
- [9] T. Shimizu, et al., "A 10-bit, 20 MHz Two-Step Parallel A/D Converter with Internal S/H," *IEEE Journal of Solid State Circuits*, Vol. 24, No. 1, pp. 13-20, February 1989.
- [10] B. S. Song, S. H. Lee, and M. F. Tompsett, "A 10-bit 15 MHz CMOS Recycling Two-Step A/D Converter," *IEEE Journal of Solid State Circuits*, Vol. 25, No. 12, pp. 1328-1338, December 1990.
- [11] B. S. Song, M. F. Tompsett, and K. R. Lakshmikumar, "A 12-bit, 1-MSample/s Capacitor Error-Averaging Pipelined A/D Converter," *IEEE Journal of Solid State Circuits*, Vol. 23, No. 6, pp. 1324-1333, December 1988.
- [12] S. H. Lewis and P. R. Gray, "A Pipelined 5-Msample/s 9 bit Analog-to-Digital Converter," *IEEE Journal of Solid State Circuits*, Vol. 22, No. 6, pp. 954-961, December 1987.

- [13] S. Sutarja and P. R. Gray, "A Pipelined 13-bit, 250-ks/s, 5-V Analog-to-Digital Converter," *IEEE Journal of Solid State Circuits*, Vol. 23, No. 6, pp. 1316-1323, December 1988.
- [14] P. Vorenkamp and J. P. M. Verdaasdonk, "A 10 b 50 Ms/s Pipelined ADC," *IEEE ISSCC Digest of Technical Papers*, pp. 34-35, February 1992.
- [15] M. Yotsuyanagi, T. Etoh, and K. Hirata, "A 10 Bit 50 MHz Pipelined CMOS A/D Converter with S/H," *IEEE Journal of Solid State Circuits*, Vol. 28, No. 3, pp. 292-300, March 1993.
- [16] J. L. McCreary and P. R. Gray, "All-MOS Charge Redistribution Analog-to-Digital Conversion Techniques - Part I," *IEEE Journal of Solid State Circuits*, Vol. 10, No. 6, pp. 371-379, December 1975.
- [17] K. Bacrania, "A 12 Bit Successive-Approximation ADC with Digital Error Correction," *IEEE Journal of Solid State Circuits*, Vol. 21, No. 6, pp. 1016-1025, December 1986.
- [18] M. Ismail and T. Fiez, *Analog VLSI Signal and Information Processing*, McGraw-Hill, 1994.
- [19] B. E. Boser, "Design and Implementation of Oversampled Analog-to-Digital Converters," Ph.D. Dissertation, Stanford University, 1988.
- [20] B. Fowler, "CMOS Area Image Sensors with Pixel Level A/D Conversion," Ph.D. Dissertation, Stanford University, October 1995.
- [21] B. P. Brandt, *Oversampled Analog-to-Digital Conversion*, Integrated Circuits Laboratory, Technical Report No. ICL91-009, Stanford University, 1991.
- [22] Y. Matsuya, K. Uchimura, et al, "A 16-bit Oversampling A/D Conversion Technology Using Triple Integration Noise Shaping," *IEEE Journal of Solid State Circuits*, Vol. 22, No. 6, pp. 921-929, December 1987.
- [23] K. Uchimura et al, "Oversampling A-to-D and D-to-A Converters with Multistage Noise Shaping Modulators," *IEEE Transactions on Acoustics, Speech and Signal Processing*, pp. 1899-1905, December 1988.

## PROBLEMS

- 29.1** A 3-bit resistor string DAC similar to the one shown in Fig. 29.2a was designed with a desired resistor of 500  $\Omega$ . After fabrication, mismatch caused the actual value of the resistors to be
- $$R_1 = 500, R_2 = 480, R_3 = 470, R_4 = 520, R_5 = 510, R_6 = 490, R_7 = 530, R_8 = 500$$
- Determine the maximum INL and DNL for the DAC assuming  $V_{REF} = 5\text{ V}$ .
- 29.2** An 8-bit resistor string DAC similar to the one shown in Fig. 29.2b was fabricated with a nominal resistor value of 1 k $\Omega$ . If the process was able to

provide matching of resistors to within 1 percent, find the effective resolution of the converter. What is the maximum INL and DNL of the converter? Assume that  $V_{REF} = 5$  V.

- 29.3** Compare the digital input codes necessary to generate all eight output values for a 3-bit resistor string DAC similar to those shown in Fig. 29.2a and b. Design a digital circuit that will allow a 3-bit binary digital input code to be used for the DAC in Fig. 29.2a. Discuss the advantages and disadvantages of both architectures.
- 29.4** Plot the transfer curve of a 3-bit  $R$ - $2R$  DAC if all  $R_s = 1.1$  k $\Omega$  and  $2R_s = 2$  k $\Omega$ . What is the maximum INL and DNL for the converter? Assume all of the switches to be ideal and  $V_{REF} = 5$  V.
- 29.5** Suppose that a 3-bit  $R$ - $2R$  DAC contained resistors that were perfectly matched and that  $R = 1$  k $\Omega$  and  $V_{REF} = 5$  V. Determine the maximum switch resistance that can be tolerated for which the converter will still have 3-bit resolution. What are the values of INL and DNL?
- 29.6** The circuit illustrated in Fig. 29.5 is known as a current-mode  $R$ - $2R$  DAC, since the output voltage is defined by the current through  $R_f$ . Shown in Fig. P29.6 is an  $N$ -bit voltage-mode  $R$ - $2R$  DAC. Design a 3-bit voltage mode DAC and determine the output voltage for each of the eight input codes. Label each node voltage for each input. Assume that  $R = 1$  k $\Omega$  and that  $R_2 = R_1 = 10$  k $\Omega$  and  $V_{REF} = 5$  V.

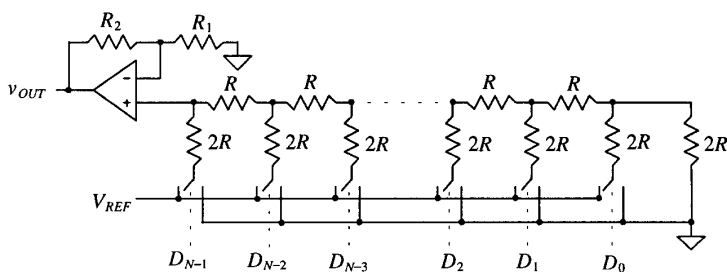


Figure P29.6

- 29.7** Design a 3-bit current steering DAC using the generic current steering DAC shown in Fig. 29.9. Assume that each current source,  $I$ , is 5 mA, and find the total output current for each input code.
- 29.8** A certain process is able to fabricate matched current sources to within 0.05 percent. Determine the maximum resolution that a current steering (nonbinary weighted) DAC can attain using this process.

- 29.9** Design an 8-bit current steering DAC using binary-weighted current sources. Assume that the smallest current source will have a value of  $1\ \mu\text{A}$ . What is the range of values that the current source corresponding to the MSB can have while maintaining an INL of  $\frac{1}{2}$  LSB? Repeat for a DNL less than or equal to  $\frac{1}{2}$  LSB.
- 29.10** Prove that the 3-bit charge scaling DAC used in Ex. 29.6 has the same output voltage increments as the  $R$ - $2R$  DAC in Ex. 29.3 for  $V_{REF} = 5\ \text{V}$  and  $C = 0.5\ \text{pF}$ .
- 29.11** Determine the output of the 6-bit charge scaling DAC used in Ex. 29.7 for each of the following inputs:  $D = 000010, 000100, 001000$ , and  $010000$ .
- 29.12** Design a 4-bit charge scaling DAC using a split array. Assume that  $V_{REF} = 5\ \text{V}$  and that  $C = 0.5\ \text{pF}$ . Draw the equivalent circuit for each of the following input words and determine the value of the output voltage:  $D = 0001, 0010, 0100, 1000$ . Assuming the capacitor associated with the MSB had a mismatch of 4 percent, calculate the INL and DNL.
- 29.13** For the cyclic converter shown in Fig. 29.17, determine the gain error for a 3-bit conversion if the feedback amplifier had a gain of  $0.45\ \text{V/V}$ . Assume that  $V_{REF} = 5\ \text{V}$ .
- 29.14** Repeat Problem 29.13 assuming that the output of the summer was always  $0.2\ \text{V}$  greater than the ideal and that the amplifier in the feedback path had a perfect gain of  $0.5\ \text{V/V}$ .
- 29.15** Repeat Problem 29.13 assuming that the output of the summer was always  $0.2\ \text{V}$  greater than ideal and that the amplifier in the feedback path had a gain of  $0.45\ \text{V/V}$ .
- 29.16** Design a 3-bit pipeline DAC using  $V_{REF} = 5\ \text{V}$ . (a) Determine the maximum and minimum gain values for the first-stage amplifier for the DAC to have less than  $\pm\frac{1}{2}$  LSBs of DNL assuming the rest of the circuit is ideal. (b) Repeat for the second-stage amplifier. (c) Repeat for the last-stage amplifier.
- 29.17** Using the same DAC designed in Problem 29.16, (a) determine the overall error (offset, DNL, and INL) for the DAC designed in Problem 29.16 if the S/H amplifier in the first stage produces an offset at its output of  $0.25\ \text{V}$ . Assume that all the remaining components are ideal. (b) Repeat for the second-stage S/H. (c) Repeat for the last-stage S/H.
- 29.18** Design a 3-bit Flash ADC with its quantization error centered about zero LSBs. Determine the worst-case DNL and INL if resistor matching is known to be 5 percent. Assume that  $V_{REF} = 5\ \text{V}$ .
- 29.19** Using the ADC designed in Problem 29.18, determine maximum offset which can be tolerated if all the comparators had the same magnitude of offset, but with different polarities, to attain a DNL of less than or equal to  $\pm\frac{1}{2}$  LSB.

- 29.20** A 4-bit Flash ADC converter has a resistor string with mismatch as shown in Table P29.20. Determine the DNL and INL of the converter. How many bits of resolution does this converter possess?  $V_{REF} = 5$  V.

| Resistor | Mismatch (%) |
|----------|--------------|
| 1        | 2            |
| 2        | 1.5          |
| 3        | 0            |
| 4        | -1           |
| 5        | -0.5         |
| 6        | 1            |
| 7        | 1.5          |
| 8        | 2            |
| 9        | 2.5          |
| 10       | 1            |
| 11       | -0.5         |
| 12       | -1.5         |
| 13       | -2           |
| 14       | 0            |
| 15       | 1            |
| 16       | 1            |

**Table P29.20**

- 29.21** Determine the open-loop gain required for the residue amplifier of a two step ADC necessary to keep the converter to within  $\frac{1}{2}$  LSB of accuracy with resolutions of (a) 4 bits, (b) 8 bits, and (c) 10 bits.
- 29.22** Assume that a 4-bit, two-step Flash ADC uses two separate Flash converters for the MSB and LSB ADCs. Assuming that all other components are ideal, show that the first Flash converter needs to be more accurate than the second converter. Assume that  $V_{REF} = 5$  V.
- 29.23** Repeat Ex. 29.12 for  $V_{IN} = 3, 5, 7.5, 14.75$  V.
- 29.24** Repeat Ex. 29.13 for  $V_{IN} = 1, 4, 6, 7$  V and  $V_{REF} = 8$  V.
- 29.25** Assume that an 8-bit pipeline ADC was fabricated and that all the amplifiers had a gain of 2.1 V/V instead of 2 V/V. If  $V_{IN} = 3$  V and  $V_{REF} = 5$  V, what would be the resulting digital output if the remaining components were considered to be ideal? What are the DNL and INL for this converter?
- 29.26** Show that the first-stage accuracy is the most critical for a 3-bit, 1-bit per stage pipeline ADC by generating a transfer curve and determining DNL and INL for the ADC for three cases: (1) The gain of the first-stage residue amplifier set

- equal to 2.2 V/V, (2) the second-stage residue amplifier set equal to 2.2 V/V, and (3) the third-stage residue amplifier set equal to 2.2 V/V. For each case, assume that the remaining components are ideal. Assume that  $V_{REF} = 5$  V.
- 29.27** An 8-bit single-slope ADC with a 5 V reference is used to convert a slow-moving analog signal. What is the maximum conversion time assuming that the clock frequency is 1 MHz? What is the maximum frequency of the analog signal? What is the maximum value of the analog signal which can be converted?
- 29.28** An 8-bit single slope ADC with a 5 V reference uses a clock frequency of 1 MHz. Assuming all other components to be ideal, what is the limitation on the value of RC? What is the tolerance of the clock frequency which will ensure less than 0.5 LSB of INL?
- 29.29** An 8-bit dual slope ADC with a 5 V reference is used to convert the same analog signal in Problem 29.27. What is the maximum conversion time assuming that the clock frequency is 1 MHz? What is the minimum conversion time that can be attained? If the analog signal is 2.5 V, what will be the total conversion time?
- 29.30** Discuss the advantages and disadvantages of using a dual-slope versus a single slope ADC architecture.
- 29.31** Repeat Ex. 29.15 for a 4-bit successive approximation ADC using  $V_{REF} = 5$  V for  $v_{IN} = 1, 3$ , and full-scale.
- 29.32** Assume that  $v_{IN} = 2.49$  V for the ADC used in Problem 29.31 and that the comparator, because of its offset, makes the wrong decision for the MSB conversion. What will be the final digital output? Repeat for  $v_{IN} = 0.3025$ , assuming that the comparator makes the wrong decision on the LSB.
- 29.33** Design a 3-bit, charge redistribution ADC similar to that shown in Fig. 29.39 and determine the voltage on the top plate of the capacitor array throughout the conversion process for  $v_{IN} = 2, 3$ , and 4 V, assuming that  $V_{REF} = 5$  V. Assume that all components are ideal. Draw the equivalent circuit for each bit decision.
- 29.34** Determine the maximum INL and maximum DNL of the ADC designed in Problem 29.33 assuming that the capacitor array matching is 1 percent. Assume that the remaining components are ideal and that the unit capacitance,  $C$ , is 1 pF.
- 29.35** Show that the charge redistribution ADC used in Problems 29.32 and 29.33 is immune to comparator offset by assuming an initial offset voltage of 0.3 and determining the conversion for  $v_{IN} = 2$  V.
- 29.36** Discuss the differences between Nyquist rate ADCs and oversampling ADCs.



**29.37** Write a simple computer program or use a math program to perform the analysis shown in Ex. 29.15. Run the program for  $k = 200$  clock cycles and show that the average value of  $v_q(kT)$  converges to the correct answer. How many clock cycles will it take to obtain an average value if  $v_q(kT)$  stays within 8-bit accuracy of the ideal value of 0.4 V? 12-bit accuracy? 16-bit accuracy?

**29.38** Prove that the output of the second-order  $\Sigma\Delta$  modulator shown in Fig. 29.50 is,

$$y(kT) = x(kT - T) + Q_e(kT) - 2Q_e(kT - T) + Q_e(kT - 2T)$$

**29.39** Assume that a first order  $\Sigma\Delta$  ADC used on a satellite in low earth orbit experiences radiation in which an energetic particle causes a noise spike resulting in the comparator making the wrong decision on the 10th clock period. Using the program written in Problem 29.37, determine the number of clock cycles required before the average value of  $v_q(kT)$  is within 12-bit accuracy of the ideal value of 0.4 V. How many extra clock cycles were required for this case versus the ideal conversion used in Prob. 36?

# Index

---

## A

- Abutting, 309
- Accumulation, 84
- Active-PI loop filter, 396
- Active load, 489
- Adaptive biasing, 699
- ADC. See Analog-to-digital converter
- Adder, 246, 285, 287
- Aliasing. See Analog-to-digital converter
- Amplifier, 489
  - adaptive biasing, 699
  - biasing, 502
  - cascode, 506
  - class A, 505, 517
  - class AB, 505, 517
  - common gate, 500, 505
  - current source load, 500
  - distortion, 512
  - efficiency, 506
  - frequency response, 492
  - noise, 512, 515
  - noise figure (NF), 515
  - overshoot, 760
  - push-pull, 510
  - sense amp, 335, 338
  - single-stage, 490
  - slew-rate, 517
  - SNR, 515
  - source-follower, 499, 505
  - transimpedance, 496
- Analog-to-digital converter, 755, 772
  - 0.5 LSB, 769
  - aliasing, 778, 838
  - aperture error, 782
  - architectures, 813
  - binary search, 832
  - charge redistribution, 835
  - coarse conversion, 819
  - DFT, 782
  - differences of oversampling and Nyquist rate converters, 838
  - differential nonlinearity (DNL), 774, 818, 827, 837
  - Discrete Fourier Transform, 782
  - fine conversion, 819
  - flash, 813
  - flash accuracy issues, 816
  - folding frequency, 780
  - gain error, 778
  - higher-order sigma-delta modulation, 846
  - integral nonlinearity (INL), 776, 817, 826, 837
  - least significant bit (LSB), 773
  - missing codes, 776
  - noise shaping, 842
  - Nyquist rate, 837
  - offset, 778
  - operational amplifier issues, 821
  - oversampling, 837
  - parallel, feed-forward, 818
  - pipeline, 823
  - pipeline accuracy issues, 825
  - pulse-density modulation, 838
  - quantization error, 774
  - quantization levels, 773
  - residue, 823
  - resolution versus oversampling, 847
  - sampling function, 779
  - SAR, 832
  - scale factor error, 778
  - sigma-delta modulation, 839
  - signal-to-noise ratio (SNR), 780
  - specifications, 772
  - successive approximation, 832
  - successive approximation accuracy issues, 837
  - successive approximation register, 832
  - transfer curve, 814
  - two-step, 818
  - two-step accuracy issues, 820
- Analog models, 165
  - body transconductance, 168
  - high-frequency model, 171
  - low-frequency MOSFET model, 165
  - transconductance, 166
- Analog signal:

characteristics of, 756  
 AOI logic, 243  
 Aperture error. See Sample-and-hold  
 ASIC, 3  
 Averaging circuit, 406

## B

Bandgap voltage references, 477, 487, 488  
 Batteries, 518, 712  
 Battery, 369  
 BCD, 791  
 Bi-Phase, 407  
 BiCMOS, 314  
   CMOS to ECL translator, 329  
   comparison to other processes, 313  
   ECL to CMOS translator, 327  
   full-swing, 323  
   inverter, 320  
   NAND gate, 323  
   NOR gate, 323  
 Biquad, 641  
 BJT (bipolar junction transistor):  
   capacitances, 319  
   design rules, 316  
 Bloats, 27  
 Body effect, 113, 168, 523  
 Body transconductance, 865  
 Bonding pad, 45, 46  
 Bootstrapped inverter, 226  
 Bottom plate of cap., 726  
 Breakdown  
   oxide, 336  
   voltage, 44, 859  
 BSIM, 107, 170, 829  
   BSIM3, 124  
   comparison with level 2, 107  
   DC equations, 111  
   hand calculations, 862  
   lateral diffusion, 75  
   oxide encroachment, 75  
   subthreshold current, 117  
 Buffers, 216, 511, 660, 702  
   n-channel only, 225  
 Bulk CMOS process, 23  
 Bus, 291

## C

C'ox, 85  
 Calma Stream Format, 5, 16  
 Capacitance, 171  
   bottom, 34  
   bottom and sidewall, 171

depletion, 33  
   n+ to p-sub, 79  
   sidewall, 34  
 Capacitances of the CN20 process, 856  
 Capacitive feedthrough, 721  
 Capacitor:  
   common centroid, 807  
   depletion, 33  
   formation with poly1/poly2, 131  
   layout, 736  
   other types, 134  
   parasitics, 132  
   undercutting, 806  
 Cascode amplifier, 506  
 Cascode differential amplifier, 602  
 Channel length modulation, 97  
 Charge injection, 720  
 Charge pump, 381  
 Chemical/mechanical polishing (CMP), 897  
 CIF, 15  
 Class AB differential amplifier, 596  
 Clock feedthrough, 719, 721  
 Clock generation, 399  
 Clock recovery circuit, 373  
   self-correcting, 412  
 Clock synchronization, 421  
 Clocked logic, 275, 281  
 CMOS, 3  
   passive elements, 131  
   scaling, 120  
   trends, 120  
   twin-tub, 23  
 CMRR, 591  
 CN20:  
   design rules, 867  
   electrical specifications, 857  
   oxide capacitance, 864  
   processes, 856  
 CN20 process, 855  
 Common-centroid, 137  
 Common-centroid layout, 595  
 Common gate amplifier, 500  
 Common-mode feedback (CMFB), 666, 674, 744  
 Communication system, 403  
 Comparators, 634, 685  
   block diagram, 686  
   characterizing, 691  
   clamped input stage, 694  
   clocked, 696  
   decision circuit, 687  
   dynamic, 748  
   gain and offset, 691  
   hysteresis, 688  
   improvements, 696

- input slew rate, 696
  - inverter based, 746
  - operation, 685
  - output buffer, 691
  - preamp, 686
  - propagation delay, 695
  - self-biased, 699
  - sensitivity, 718
  - transient response, 693
  - Compensation, 621
  - Composite, 150
  - Constant-gm diff-amp, 608
  - Contact resistance, 55, 858
  - Crosstalk, 56
  - Current differential amplifier, 607
  - Current source/sink, 427
    - biasing, 429, 431
    - cascode, 437
    - cascode connection, 433
    - design, 429
    - diode referenced, 470
    - dynamic, 748
    - excess gate voltage, 429, 430
    - floating, 661
    - layout, 444, 447, 449
    - matching, 446
    - mirror, 427
    - output resistance, 430, 434, 451, 452
    - regulated cascode, 451
    - self-biasing, 469
    - sensitivity, 438
    - SPICE statement, 446
    - subthreshold operation, 483
    - temperature coefficient, 439
    - thermal voltage referenced, 474
    - threshold voltage referenced, 469
    - transconductance multiplier, 480
    - transient response, 442
    - voltage output swing, 434
    - wide swing, 434, 455
    - Wilson mirror, 450
  - Current source load, 500, 582
  - Current starved VCO, 384
- 
- D**
- DAC. See Digital-to-analog converter
  - Data conversion:
    - signal characteristics, 755
  - DC sweep convergence, 128
  - Decoder, 341
    - tree, 341
  - Deep-Submicron MOSFET, 892
  - Delay elements, 419
  - Delay-locked loop, 417
  - Delay time, 189
  - Delta-sigma modulation. See Sigma-delta modulation
  - Depletion, 85
  - Depletion capacitance, 33
  - Design rules, 867
  - Design verification, 149
  - DIBL, 123
  - Dielectric constants, 856
  - Differential amplifier, 579
    - cascode, 602
    - class AB, 597
    - common-mode range (CMR), 584, 599, 604, 605
    - common-mode rejection ratio (CMRR), 591
    - current, 701
    - current source load, 582, 599
    - gain, 587
    - low-voltage, 605
    - matching, 594
    - noise, 592
    - output resistance, 587
    - p-channel, 583
    - self-biased, 691, 699
    - slew-rate, 588, 602
    - source cross-coupled pair, 596
    - transconductance, 582
    - wide-swing, 605
  - Differential output op-amp, 664, 672, 705, 742
    - benefits, 742
    - simulating, 707
  - Differential to single-ended conversion, 711
  - Digital-to-analog converter, 756, 762
    - .5 LSB, 769
    - accuracy, 765
    - binary switch array, 792
    - binary-weighted capacitor array, 805
    - charge scaling, 805, 835
    - current source mismatch, 802
    - current steering, 800
    - cyclic, 810
    - decoder, 792
    - differential nonlinearity (DNL), 766, 795, 797, 803, 805
    - dynamic range, 772
    - full-scale voltage, 764
    - gain error, 770
    - input combinations, 763
    - integral nonlinearity (INL), 768, 795, 803, 805
    - latency, 770
    - least significant bit (LSB), 765
    - monotonicity, 767
    - most significant bit (MSB), 765

- nonmonotonic, 767
  - offset, 770
  - pipeline, 811
  - R-2R* ladder, 797
  - reference voltage, 763
  - resistor string, 792
  - resistor string mismatch, 795
  - resolution, 765
  - settling time, 770
  - signal-to-noise ratio (SNR), 770
  - specifications, 762
  - split-array, 807
  - transfer curve, 764
  - Digital models, 185
  - Digital phase-locked loop, 373, 375
    - active-PI loop filter, 396
    - block diagram, 374
    - charge-pump, 382
    - lock range, 391
    - loop-filter, 391
    - PFD, 374, 382
    - PFD DPLL, 398
    - pull-in range, 391
    - tri-state output, 381
    - XOR DPLL, 391
    - XOR phase detector, 374
  - Digital signal:
    - characteristics of, 756
  - Diode, 32, 67, 471
    - breakdown voltage, 44
    - drain-substrate, 276
    - SPICE modeling, 37
    - storage capacitance, 35
    - storage time, 37
  - Distortion, 506, 512, 529
  - Distortionless transmission, 403
  - DLL. *See* Delay locked loop
  - Domino logic, 283
  - Doublet, 678
  - Drain-induced barrier lowering, 123
  - DRAM charge sharing, 335
  - Drawn length, 71
  - Drawn width, 71, 75
  - DRC (design rule check), 30
  - Drive current, 123, 891
  - Drivers, 216
    - distributed, 221
  - DSL logic, 247
  - Dynamic circuits, 719
    - amplifiers, 750
    - charge injection, 748
    - clock feedthrough, 719
    - comparator, 746
    - current mirrors, 748
    - IOS, 748
    - KT/C noise, 748
    - logic, 275
    - OOS, 748
    - reducing op-amp offset-voltage, 745
    - simulating, 277
    - storage capacitors, 748
    - switch, 719
    - switched-capacitor integrator, 731
    - switched-capacitor resistor, 728
  - Dynamic CMOS latch, 748
  - Dynamic logic, 275
    - domino logic, 283
    - PE logic, 282
- 
- ## E
- 
- Edge detection, 411
  - Effective channel length, 84
  - Effective switching resistance, 186
  - Efficiency, 506
  - Electrostatic discharge (ESD), 67
    - protection, 69
  - Epi layer, 23
  - Equalizer, 404
  - Excess gate voltage, 429
- 
- ## F
- 
- Feedback, 525
    - amplifier, 532
    - basic amplifier, 532
    - beta network, 534
    - BW extension, 528
    - closed-loop impedances, 540
    - closed-loop parameters, 539
    - determining signal path, 534
    - equivalent transconductance, 549, 551
    - forward path, 534
    - gain desensitivity, 527
    - gain margin, 567
    - input/output impedance control, 530
    - input mixing, 533
    - inversions around the loop, 534, 535
    - open-loop impedances, 538
    - open-loop parameters, 537
    - output sampling, 533
    - phase margin, 567
    - properties, 531
    - recognizing topologies, 532
    - reduction in nonlinear distortion, 529
    - rules for output sampling, 534
    - series-series, 537, 558, 562
    - series-shunt, 536, 540

series mixing, 532  
 series sampling, 532  
 shunt-series, 536  
 shunt-shunt, 537, 547  
 shunt mixing, 532  
 shunt sampling, 532  
 stability, 564  
 using a gate-drain resistor, 555  
 Field inversion, 859  
 Filter, 641  
     switched capacitor, 739  
 Flatband voltage, 92  
 Flicker noise, 178  
 Flip-flops, 265  
     clocked, 266  
     data (D), 269, 271  
     edge triggered, 268  
     JK, 266  
     set-reset (SR), 265  
     timing, 269  
 Floating battery, 712  
 Floating current source, 518, 661  
 Floorplan, 290, 783  
 Folded-cascode, 654  
 Foresight design rules, 867  
 FOX (field oxide), 32, 71  
 Frequency synthesis, 399  
 $f_T$ , unity current gain frequency, 174, 183

---

## G

Gain-enhancement, 674  
 Gain margin, 567  
 Gamma, 92  
 GDS, 5, 16  
 GMIN, 277  
 GOX (gate oxide), 71  
 Gradual channel approximation, 119  
 Gray code, 791  
 Ground bounce, 56  
 Guard rings, 137, 215

---

## H

Hand calculations, 862  
 Harmonic distortion, 513, 742  
 Hi-Z, 226, 252  
 Hierarchy, 20, 59  
 High-speed op-amp, 677  
 High-impedance node, 503  
 Hot-carrier effect, 120  
 HSPICE, 111  
 Hysteresis, 356, 688

---

## I

$I_{drive}$ , 123, 890  
 IGFET, 107  
 Inductance, 58  
 Input offset storage (IOS), 748  
 Instanted (or placed), 309  
 Interconnect burden, 307  
 Inverter, 201  
     BiCMOS, 320  
     bootstrapped, 226  
     buffer design, 216  
     dynamic power dissipation, 210  
     layout, 212  
     noise margins, 203  
     other configurations, 224  
     SPICE simulation, 203  
     switching characteristics, 205, 321  
     transfer characteristics, 202  
     tri-state output, 226

---

## J

Jitter, 379, 415, 829  
 JS (BSIM current density), 276, 471

---

## K

KP, 95  
 KT/C noise, 147, 723, 748

---

## L

Lambda, 170, 447  
     channel length modulation, 115  
     mobility modulation, 115  
 LASI:  
     common problems, 14  
     cursor, 15  
     distance measurements, 10  
     frustration, 12  
     help, 6  
     invisible cells, 15  
     ranking cells, 9  
     schematics, 150  
 LASICKT, 149  
     header, 158  
     inputs, 157  
     starting, 157  
 LASIDRC, 30, 867  
 Latch, 281  
 Latch-up, 102, 212, 225  
 Lateral diffusion, 27, 75

Layout:  
 adder, 298  
 binary-weighted capacitor array, 806  
 capacitor, 132, 141, 297, 736  
 capacitor bottom plate, 726  
 circle, 22  
 common-centroid, 137  
 D-FF, 298  
 hierarchy, 59  
 inverter, 212, 291  
 large MOSFETs, 102, 220  
 lateral npn, 316  
 MOSFET, 75, 77, 102  
 MUX/DEMUX, 299  
 n-well resistor, 27  
 NAND and NOR gates, 235, 291  
 pad, 46  
 parasitic pnp, 470  
 power and ground, 295  
 resistor, 136  
 SR latch, 291  
 standard-cell, 291  
 standard-cell frame, 71, 215  
 stick diagram, 301  
 TG, 291  
 VLSI, 289  
 LDD, 120  
 Leakage, 471  
 Leakage, charge, 276, 277, 278, 280  
 Leakage current, 42  
 Level shifting, 710  
 Linear region, 98  
 Lock range, 391  
 LOCOS, 71  
 Low-impedance node, 503  
 Low voltage:  
   current mirror, 455  
   differential amplifiers, 605  
   op-amp, 661

## M

Manchester NRZ, 407  
 Matching, 113, 137, 446  
   differential amplifier, 594  
   layout, 447  
 Memory, 331  
   cells, 332  
   DRAM, 332, 345  
   DRAM array, 333  
   DRAM mbit size, 347  
   DRAM timing, 343  
   EEPROM, 350, 352  
   EPROM, 350, 351

flash, 350, 352  
 folded bit line, 333, 347  
 organization, 331  
 PROM, 350  
 ROM, 350  
 SRAM, 332  
 Metal layers, 45  
   capacitive feedthrough, 53  
   parasitics, 50  
 Metal migration, 54  
 Miller's Theorem, 492, 493, 521  
 Mixed signal, 665, 742  
 MOSFET, 83  
   adjusting threshold voltage, 93  
   capacitances, 84, 87  
   channel length modulation, 97  
   depletion, 93  
   diode connected, 168  
   effective width, 75  
    $f_T$ , 183  
   IV characteristics, 93  
   output resistance, 129  
   overlap capacitances, 84  
   pinch-off, 96  
   punchthrough, 96  
   temperature effects, 175  
   threshold voltage, 88  
 MOSIS, 15, 45, 107, 149, 196, 867, 873, 885  
 Multipliers, 704  
   multiplying quad, 705  
   simulating, 707  
   squaring circuits, 715  
 Multivibrator circuits, 362  
   source coupled, 387  
 MUX/DEMUX, 261

## N

N-select, 149, 878  
 NAND gate, 231  
   layout, 235  
   switching characteristics, 236  
   switching point voltage, 233  
 NMOS, 23  
 NMOS inverter, 224  
 Noise, 141, 178  
   1/f (flicker), 178  
   amplifier, 512  
   analysis, 141  
   differential amplifier, 592  
   KT/C, 147, 748  
   MOSFET, 178  
   phase, 417  
   procedure for noise analysis, 144

substrate, 736  
 temperature, 516  
 thermal, 142, 178  
 Noise figure (NF), 515  
 Noise margins, 203  
 Noise temperature, 516  
 Nonoverlapping clocks, 279  
 NOR gate, 231  
   layout, 235  
   switching characteristics, 236  
 NPN, junction-isolated, 314  
   model, 316  
 NRD and NRS, 102  
 Nyquist Criterion, 758

---

## O

Offset voltage, 595, 630, 632, 745  
 Ohmic region, 94  
 One-shot, 254, 362  
 Operational amplifiers, 617  
   block diagram, 617  
   cascode, 636  
   characterizing, 630  
   circuit for simulating open-loop gain, 623  
   CMFB circuit, 667  
   CMRR, 631  
   common-mode output voltage, 665  
   common-mode feedback circuit, 666  
   common-mode voltage temperature coefficient, 712, 713  
   compensating with buffer, 649  
   compensation, 621  
   compensation capacitor, 625  
   differential output, 664, 707  
   folded-cascode, 654  
   gain, 618  
   gain accuracy in ADCs, 821  
   gain-enhancement, 674  
   high-gain, 675  
   high-speed, 675  
   layout of folded-cascode, 670  
   linearity, 822  
   low-voltage, 661  
   offset-voltage, 630, 745  
   OTA with buffer, 647  
   output stage, 620, 660  
   output voltage swing, 631  
   phase margin, 625  
   PSRR, 632, 654, 665  
   reducing offset-voltage, 745  
   RHP zero, 627  
   short-circuit protection, 620  
   slew-rate, 633  
   voltage regulator, 650  
   wide-swing, 642, 654  
   without slew-rate limitations, 634  
 Operational transconductance amplifier, 607, 637  
 Orbit Semiconductor, 855  
 Oscillator, 383  
   astable, 364  
   ring, 209  
   ring oscillator with enable, 369  
   Schmitt Trigger, 359  
   voltage-controlled, 361  
 OTA, 607  
   compensating with buffer, 649  
   compensation, 644  
   DC stability, 645  
   folded-cascode, 654  
   fully differential, 666  
   wide-swing, 642  
 Output buffer, 518, 660  
 Output offset storage (OOS), 748  
 Overglass layer, 46  
 Overlap capacitance, 84  
 Oxide-encroachment, 75  
 Oxide-nitride-oxide (ONO), 345  
 Oxide breakdown, 123, 336  
 Oxide capacitance, 85, 864  
 Oxide growth, 26

---

## P

P-base, 314  
 P-select, 149, 878  
 Pad layer, 46  
 Parasitics, 3  
 Pass transistor, 255  
 Path selector, 260  
 Patterning, 24  
 PE logic, 282  
 Peak detector, 406  
 Phase-locked loop (PLL), 373  
 Phase detector, 375, 418  
 Phase frequency detector (PFD), 379  
   characteristics, 381  
   gain, 382  
 Phase margin, 567  
 Physical constants, 855  
 Piecewise-linear, 159  
 Pinch-off, 96  
 Pipelining, 285  
 Pitch, 347  
 Planarization, 897  
 PMOS, 23  
 Poly, 65  
 Poly2, 131



Polycide, 75  
 Power and ground, 295  
 Power conversion efficiency, 506  
 Power delay product (PDP), 211  
 Process characteristic time constant, 188  
 PSPICE, 40, 439, 441, 443, 466, 479  
   noise analysis, 144  
 PSRR, 632  
 PTAT, 475  
 Pull-in range, 391  
 Pulse-density modulation, 838  
 Pulse statement, 159  
 Pumps, 364  
 Punchthrough, 96  
 Push-pull amplifier, 510

## R

Random offsets, 630  
 Ranking, 9  
 Ratioless shift register, 288  
 Receiver (digital), 424  
 References, 463  
   PTAT, 475  
   voltage and current, 463  
 Regularity, 290  
 Regulated cascode, 451  
 Reliability, 468  
 Resistance:  
   looking into the drain, 497  
   looking into the source, 498  
   sheet, 858  
 Resistor, 27  
   delay through, 39  
   n+, 78  
   n-well, 24, 32, 792  
   sheet resistance, 31  
   switched-capacitor, 728  
   TC, See Temperature Coefficient  
   Temperature Coefficient (n-well), 134, 441  
 Reticle, 26, 736  
 RHP zero, 627  
 Ring oscillator, 209  
 Ringing, 175  
 Routing, 309  
 ROX (recessed oxide), 32, 71

## S

S/H. See Sample-and-hold  
 Sample-and-hold, 724, 729  
   acquisition time, 760  
   aperture error, 761  
   aperture jitter, 761

aperture uncertainty, 761  
 droop, 761  
 gain error, 760  
 hold mode, 761  
 linearity, 761  
 offset, 760  
 overshoot, 760  
 sample mode, 760  
 sampling error, 761  
 settling time, 760  
 Scale current, 276  
 Schmitt trigger, 355, 370  
   applications, 359  
   oscillator, 359  
 Schottky diode, 66  
 Scope probe, 196  
 Self-biased diff-amp, 691  
 Self-biasing, 469  
   startup circuit, 470  
 Self-correcting clock, 412  
 Sense amplifier, 335, 338  
 Sensitivity analysis, 438  
   voltage divider, 465  
 Series connection of MOSFETs, 192  
   delay, 194  
 Settling time, 678, 726, 741  
 Sheet resistance, 31, 858  
 Shift register, 279  
 Short channel, 119  
 Sigma-delta modulation, 839  
   first-order, 840  
   higher order, 846  
   noise shaping, 842, 846  
   resolution versus oversampling, 847  
 Silicide, 75  
 Single-ended to differential conversion, 711, 753  
 Single-ended to differential converter, 670  
 Slew rate, 517, 588, 633, 702, 742  
 Small-signal models, 167  
 Small-signal parameters, 865  
 SNR (signal to noise ratio), 515  
 Source coupled multivibrators, 387  
 Source coupled VCO, 387  
 Source follower, 499, 505  
   body effect, 522  
 SPICE, 98, 149, 158, 163  
   ABSTOL, 277, 279, 663  
   BSIM, 859  
   common mistakes, 127  
   diode breakdown voltage, 44  
   diode model, 473  
   diode modeling, 37  
   distortion, 513  
   GMIN, 277, 278

LASICKT, 149  
 level 2, 98, 859  
 noise analysis, 144  
 operating point analysis, 130  
 pulse statement, 159  
 RELTOL, 277  
 sensitivity analysis, 438  
 simulating dynamic circuits, 277  
 VNTOL, 277  
 SPICE problems, 127, 239, 644  
   .NODESET, 476  
   common mistakes, 127  
   DC sweep, 476  
   transient, 590  
 Squaring circuits, 715  
 Stability, 564, 621  
 Standard-Cell, 291, 311  
 Startup circuit, 470  
 Static logic, 231  
 Stepper, 26  
 Stick diagram, 301  
 Storage node, 275  
 Strong inversion, 86  
 Student projects, 196  
 Substrate, 23  
 Substrate pump, 364  
 Subthreshold:  
   current, 117, 166  
   current source/sink, 483  
   output resistance, 117  
   transconductance, 167, 865  
 Switch logic:  
   CSVL, 247  
   DVSL, 247  
 Switched capacitor:  
   capacitor layout, 736  
   filter, 733  
   integrator, 731  
   ladder filter, 739  
   resistor, 728  
   slew-rate requirements, 754  
 Systematic offsets, 630

---

## T

Temperature coefficient:  
   bandgap, 479  
   diode, 473  
   diode self-biasing, 473  
   MOSFET voltage divider, 465  
   resistor-MOSFET divider, 464  
   thermal voltage self-biasing, 476  
   threshold voltage, 465  
   transconductance multiplier, 480  
 Temperature effects, 175  
   bandgap, 175  
   capacitor, 134, 140  
   KP, 176, 440  
   MOSFETs, 175  
   threshold voltage, 175  
 TG (transmission gate), 255  
 Thermal runaway, 176  
 Thermal voltage, 33, 471, 856  
 Thermometer code, 791, 800, 813  
 Threshold voltage, 88  
   matching, 446  
 Timing errors, 404  
 TLC files, 5  
 Total harmonic distortion, 513  
 Transconductance, 167, 865  
   variation with frequency, 173  
 Transconductance parameter, 95, 864  
 Transconductor-C Filters, 641  
 Transimpedance amplifier, 496  
 Transmission gate, 255, 281  
   applications, 260  
   delay through, 258  
   layout, 258  
   static gates, 264  
 Triode region, 94, 98, 171, 864  
   channel resistance, 171  
 Tri-state buffer, 252  
 Tri-state phase detector output, 381  
 Two's complement, 791

---

## V

Velocity saturation, 120  
 VLSI, 289  
 Voltage coefficient, 134  
   resistor, 134  
 Voltage controlled oscillator (VCO), 374, 383, 420  
   characteristics, 383  
   current starved, 384  
   for use with the PFD, 383  
   logic based, 391  
   source coupled, 387  
   use with XOR phase detector, 379  
 Voltage dividers, 463  
 Voltage follower, 702  
 Voltage generators, 364  
   example, 368  
   higher voltages, 367  
 Voltage reference, 478  
   bandgap, 477  
   transconductance multiplier, 481  
 Voltage regulator, 650  
 $V_T$  (thermal voltage), 856

---

**W**

Wafer, 5

Weak inversion, 119

Wide-swing diff-amp, 606

---

**X**

XOR gate, 245, 264, 311

XOR phase detector, 375

    characteristics, 377

    gain, 378

---

**Y**

Yield, 289

---

**Z**

Zero-nulling resistor, 627

Zipper logic, 285

# About the Authors

**R. Jacob (Jake) Baker** is an Associate Professor of Electrical Engineering and Assistant Director of the Microelectronics Research Center at the University of Idaho. He teaches several courses on CMOS analog and digital circuit design and has consulted internationally in these areas. With eight years industry experience at E.G. & G. Energy Measurements, Lawrence Livermore National Laboratory and Micron Semiconductor, Dr. Baker brings an industrial perspective into the classroom and was selected the College of Engineering's Outstanding Young Faculty in 1997. He received the BS and MS degrees in Electrical Engineering from the University of Nevada, Las Vegas and the Ph.D. degree in Electrical Engineering from the University of Nevada, Reno.

**Harry W. Li** is an Associate Professor in the Department of Electrical Engineering at the University of Idaho. He received a BS in Electrical Engineering from the University of Tennessee and his MS and Ph.D. in Electrical Engineering from the Georgia Institute of Technology. Actively involved with the university's Microelectronics Research Center, Dr. Li's research interests include data-converters, analog CAD, and delay-locked loops. He has received numerous teaching awards and is the department's outreach coordinator, presenting "hands-on" demonstrations of electrical engineering to area schools.

**David E. Boyce** is an independent consultant on semiconductor technologies. He received his B.S. and Ph.D. in Electrical Engineering from Syracuse University. He has over 20 years experience working for General Electric, RCA, and Harris. He has been involved in numerous projects mainly in the fields of medical ultrasonic imaging, power and high voltage integrated circuits, and optoelectronics. Because of his broad scope of experience in integrated circuit technologies, he started writing his own software for personal computers a few years ago, almost as a hobby. This eventually evolved into the LASI software used in this book. For questions regarding LASI, he may be contacted at the address or phone number that can be found in the LASI software package.

# LASI Software

If a floppy disk is not included with this text, the LASI software can be downloaded from the Internet at:

[http://www.mrc.uidaho.edu/vlsi/cad\\_free.html](http://www.mrc.uidaho.edu/vlsi/cad_free.html)

## Hardware Requirements

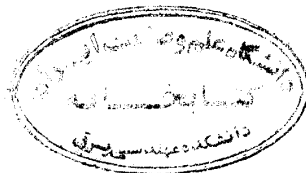
LASI software will run in some configuration on almost any IBM compatible computer with a 386 or higher. You need 4 Mbytes of memory, a hard disk drive, VGA video, and a mouse. For small to medium layouts, a 386 or 486 CPU will be adequate. For larger layouts, a Pentium with PCI bus is recommended.

## Software Contents and Installation

Whether downloaded from the Internet, or on floppy disk, you should have the following files:

- 2INSTALL.EXE: Installation program.
- LASI52?.ZIP: Main LASI system files. ("?" indicates revision number)
- CN20.ZIP: Setups for Orbit's CN20 process using the Foresight design rules. Information is given in Appendix A.
- MOSIS.ZIP: Setups using the MOSIS Scalable Design Rules in terms of  $\lambda$ . Information is given in Appendix B.
- 2UCHIP.ZIP: An example of chip design (layouts and schematics) using the MOSIS Scalable Design Rules with  $\lambda = 1 \mu\text{m}$ .

If not on a floppy disk already, copy the above files to a 1.44MB floppy disk. At the DOS command prompt C:\>, type: "a:2install". Assuming LASI is to be installed on the C: drive, specify the hard drive as "C" and answer "Y" to all questions. The LASI tutorial will run when the installation is complete. If the tutorial runs, LASI is probably installed correctly. Reboot after you quit the tutorial. To run LASI on any of the sample layouts, go to Chapter 1 of the text and follow the instructions in Section 1.2 "Using the LASI Program." This will give you the fundamentals of how to work with LASI.





## Electrical Engineering

### CMOS: CIRCUIT DESIGN, LAYOUT, AND SIMULATION

Complete with layout software for the PC, this exceptionally comprehensive presentation of CMOS integrated circuit design will guide you through the process of implementing a chip from the physical definition through the design and simulation of the finished chip. *CMOS: Circuit Design, Layout, and Simulation* provides an important contemporary view of a wide range of analog/digital circuit blocks, the BSIM model, data converter architectures, and much more. Outstanding features of this text include:

- Phase- and delay-locked loops, mixed-signal circuits, and data converters
- More than 1,000 figures, 200 examples, and over 500 end-of-chapter problems
- In-depth coverage of both analog and digital circuit-level design techniques
- Real-world process parameters and design rules
- Information on MOSIS fabrication procedures and other key topics of interest

*CMOS: Circuit Design, Layout, and Simulation* can also be used with standard software packages used in academia and industry (Cadence, L-Edit, Magic, Mentor, etc.). It is useful as an advanced-level textbook or reference for engineers, engineering managers, layout designers, layout draftsmen, computer engineers, professors, and computer scientists.

### About the Authors

**R. Jacob Baker** is an Associate Professor of Electrical Engineering and Assistant Director of the Microelectronics Research Center at the University of Idaho. He teaches several courses on CMOS analog and digital circuit design and has consulted internationally in these areas. With eight years of industry experience at E. G. & G. Energy Measurements, Lawrence Livermore National Laboratory, and Micron Semiconductor, Dr. Baker brings an industrial perspective into the classroom and was selected as the College of Engineering's Outstanding Young Faculty in 1997.

**Harry W. Li** is an Associate Professor in the Department of Electrical Engineering at the University of Idaho. Dr. Li is actively involved with the University's Microelectronics Research Center. He has received numerous teaching awards and is the department's outreach coordinator, presenting "hands-on" demonstrations of electrical engineering to area schools.

**David E. Boyce** is an independent consultant on semiconductor technologies. He has more than 20 years experience working for General Electric, RCA, and Harris. He has been involved in numerous projects, mainly in the fields of medical ultrasonic imaging, power and high-voltage integrated circuits, and optoelectronics.

IEEE Press  
445 Hoes Lane  
P.O. Box 1331  
Piscataway, NJ 08855-1331 USA

IEEE Order No. PC5689

ISBN 0-7803-3416-7



9 780780 334168

# CMOS

CIRCUIT DESIGN,  
LAYOUT, AND  
SIMULATION

**Baker**

TK  
7871.99  
.M44B335  
1998  
C.1