BÀI GIẢNG MÔN HỌC
CÔNG NGHỆ VI ĐIÊN TỬ

Credits: 2

Prerequisites:- Semiconductor Devices
- Microelectronic Circuit Design

REFERENCES

1. HONG H. LEE, *Fundamentals of Microelectronics Processing.* 3rd Ed., McGraw-Hill; USA; 1990.
2. STEPHEN BROWN and ZVONKO VRANESIC, *Fundamentals of Digital Logic with VHDL Design*, 3rd Ed., Mc.Graw-Hill, 2000.
3. SUNG-MO KANG and YUSUF LEBLEBICI, *CMOS Digital Integrated Circuits Analysis and Design***.** Mc.Graw-Hill, 2005.
4. DAN CLEIN, *CMOS IC Layout*, Newnes, 2000.
5. DAVID A. HODGES, HORACE G. JACKSON, RESVE A. SALEH, *Analysis and Design of Digital Integrated Circuits in Deep Submicron Technology*, Mc.Graw-Hill, 2003.

CHƯƠNG 1. CƠ SỞ CÔNG NGHỆ MẠCH TÍCH HỢP

§1.1 Các mạch tích hợp

Các mạch tích hợp (IC) là các mạch điện tử được chế tạo bởi việc tạo ra một cách đồng thời các phần tử riêng lẻ như transistor, diodes ... trên cùng một chip bán dẫn nhỏ (điển hình là Si), các phần tử được nối với nhau nhờ các vật liệu kim loại được phủ trên bề mặt của chip. Các vật liệu kim loại đóng vai trò như các " wireless wires". Ý tưởng này lần đầu tiên được đưa ra bởi Dummer năm 1952. Các mạch tích hợp đầu tiên được phát minh bởi Kilby, 1958.

Các mạch tích hợp về cơ bản được chia thành 2 loại chính: analog (hay linear) và digital (hay logic). Các mạch tích hợp tương tự hoặc khuếch đại hoặc đáp ứng các điện áp biến đổi. Tiêu biểu là các mạch khuếch đại, timers, dao động và các mạch điều khiển điện áp (voltage regulators). Các mạch số tạo ra hoặc đáp ứng các tín hiệu chỉ có hai mức điện áp. Tiêu biểu là các bộ vi xử lý, các bộ nhớ, và các microcomputer. Các mạch tích hợp cũng có thể được phân loại theo công nghệ chế tạo: monolithic hoặc hybrid. Trong khôn khổ giáo trình này chúng ta chỉ ngiên cứu loại thứ nhất.

Quy mô của sự tích hợp của các mạch tích hợp trên sơ sổ Silicon đã tăng lên rất nhanh chóng từ thế hệ đầu tiên được chế tạo bởi Texas Instruments năm 1960 với tên gọi SSI (Small Scale Integration) đến thế hệ mới ULSI. Hiện nay công nghệ CMOS với minimum device dimension (khoảng cách gate to gate) đạt tới cỡ vài chục nm (0.65, 0.45).

Khuynh hướng chủ đạo trong việc giảm nhỏ kích thước linh kiện trong công nghệ mạch tích hợp là giảm chi phí cho cùng một chức năng, giảm tiêu thụ công suất và nâng cao tốc độ của linh kiện. Một khuynh hướng khác là vẫn tiếp tục sử dụng các đĩa bán dẫn lớn để giảm chi phí trên chip. Với cả hai khuynh hướng trên, công nghệ xử lý vi điện tử luôn phải được cải tiến.

Các công nghệ IC chủ yếu hiện nay là công nghệ MOS và công nghệ BJT cho silicon và MES cho gallium arsenide.

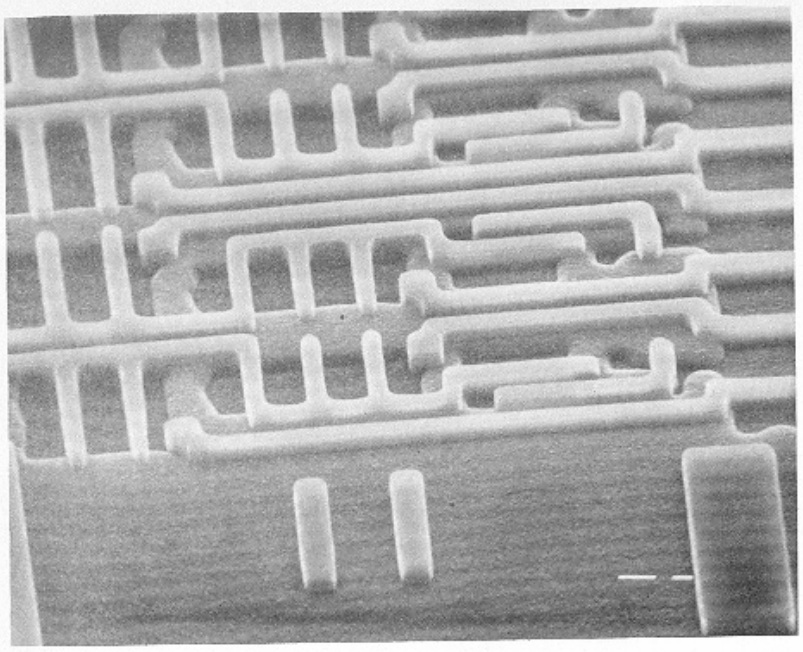


FIGURE 1-1
A silicon wafer containing 89 chips and 7 test structures (Sze, 1983). (Courtesy A. Kornblit and T. Giniecki, AT&T Bell Laboratories)

Hinh 1-1 (256 K DRAM, 1983, AT&T Bell Laboratories)

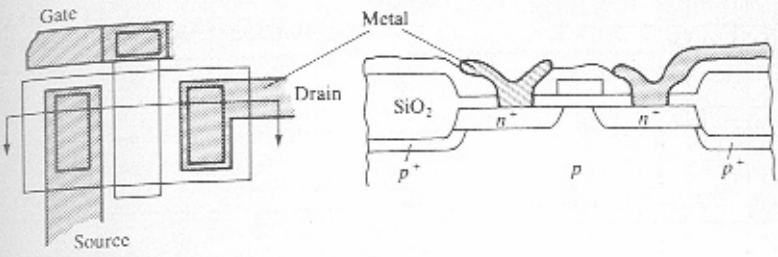


FIGURE 1-2
Planar and sectional view of a MOS structure (Hodges and Jackson, 1983).

§1.2 Bán dẫn và các hạt tải

Si đơn tinh thể là vật liệu cơ sở cho công nghệ IC. Hình 1-2a mô tả một planar view của tinh thể Si với các điện tử của lớp ngoài cùng (lớp vỏ) trong các liên kết cọng hóa tri (covalent bond) giữa các nguyên tử lân cận. Một chất bán dẫn có thể được định nghĩa như là một vật liệu có độ dẫn điện có thể điều khiển được, trong khoảng trung gian giữa điện môi và kim loại. Khả năng thay đổi độ dẫn của Si trong khoảng nhiều bậc có thể được thực hiện bởi việc đưa vào mạng tinh thể Si các nguyên tử tạp chất hóa trị 3 như Boron hoặc hóa trị 5 như Phosphorus, chúng được gọi là các dopant hoặc là các tạp chất mong muốn. Quá trình này gọi là quá trình pha tạp hay doping. Các bán dẫn sạch được gọi là bán dẫn thuần hay intrinsic, các bán dẫn pha tạp gọi là extrinsic. Nếu pha tạp nhóm 5 (chẳng hạn P) vào Si thì ngoài 4 điện tử liên kết cọng hóa trị với 4 điện tử lớp vỏ của các nguyên tử Si lân cận, điện tử thứ 5 của nguyên tử tạp có liên kết lỏng lẻo với hạt nhân và có thể chuyển động tương đối dễ dàng trong mạng tinh thể Si. Dạng bán dẫn này được gị là bán dẫn loại-n, và tạp nhóm 5 được gọi là tạp donor. Nếu pha tạp nhóm 3 (chẳng hạn B) vào Si thì 3 điện tử lớp vỏ của nguyên tử tạp liên kết cọng hóa trị với các điện tử lớp vỏ của các nguyên tử Si lân cận do đó có thể coi lớp vỏ của nguyên tử tạp có 7 điện tử, và bị trống một điện tử. Vị trí liên kết khuyết này được gọi là một lỗ trống (hole). Một điện tử từ nguyên tử Si gần đó có thể "rơi" vào chỗ trống này và lỗ trống được xem như chuyển dời đến vị trí mới. Bán dẫn loại này được gọi là bán dẫn loại -p, và tạp nhóm 3 được gọi là tạp acceptor. Các điện tử và lỗ trống khi dịch chuyển sẽ mang theo chúng các điện tích âm và dương nên được gọi là các hạt tải. Các chất bán dẫn có thể ở dạng nguyên tố (như Si, Ge) hoặc hợp phần. Số điện tử trung bình trên một nguyên tử thường bằng 4, ngoại trừ trường hợp các bán dẫn A^{V}-B^{VI}.

Một bán dẫn thuần thường là điện môi trừ khi nó được kích thích nhiệt hoặc quang. Nếu kích thích đủ mạnh nó có thể trở thành dẫn điện. Các mức năng lượng khả dĩ của điện tử là rời rạc và sự kích thích sẽ làm cho các điện tử có thể nhảy lên mức năng lượng cao hơn. Vì chất bán dẫn có thể là điện môi hay dẫn điện tùy thuộc vào mức độ kích thích, nên có thể coi nó biểu hiện như một chất dẫn điện nếu năng lượng kích thích vượt quá một mức ngưỡng nhất định, gọi là energy barrier, ký hiệu E_g (còn được gọi là khe năng lượng - energy gap). Khe năng lượng thay đổi từ 0.18 eV cho InSb tới 3.6 eV cho ZnS. Các vật dẫn như kim loại không có khe năng lượng nên có thể dẫn điện khi có hoặc không có kích thích. Các chất cách điện có khe năng lượng lớn đến mức không dẫn điện ngay cả khi kích thích mạnh. Khi

không có kích thích tất cả các điện tử của bán dẫn chiếm các mức năng lượng thấp trong các trạng thái hóa trị. Mặc dù các mức năng lượng là gián đoạn nhưng vì có rất nhiều mức nên có thể xem tập hợp các trạng thái cọng hóa trị như một dải hay vùng hóa trị (valence band). Mức năng lượng cao nhất của vùng hóa trị ký hiệu là E_v. Phía trên khe năng lượng (còn gọi là vùng cấm) là dải năng lượng của các trạng thái dẫn, gọi là vùng dẫn. Mức năng lượng thấp nhất của vùng dẫn ký hiệu là E_c. Hình 1-2a mô tả cấu hình các mức năng lượng của một bán dẫn thuần ở 0^oK. Khi bán dẫn thuần được pha tạp donor, các điện tử donor sẽ chiếm các mức năng lượng gần dưới vùng dẫn, với mức năng lượng thấp nhất trong các mức này được gọi là mức donor, ký hiệu là E_d (hình 1-2b). Khi bán dẫn thuần được pha tạp acceptor, các lỗ trống sẽ chiếm các mức năng lượng gần trên đỉnh vùng hóa trị, với mức năng lượng cao nhất trong các mức này được gọi là mức acceptor, ký hiệu là E_a (hình 1-2c). Khi bán dẫn thuần chịu kích thích nhiệt, một số điện tử trong vùng hóa trị bị kích thích có thể vượt qua vùng cấm để lên vùng dẫn đồng thời tạo ra một số lỗ trống tương ứng ở vùng hóa trị, và các cặp điện tử lỗ trống (EHP - electron hole pair) được tạo ra. Vì các mức donor trong bán dẫn loại -n rất gần với vùng dẫn nên các kích thích nhẹ cũng đủ để làm cho các điện tử donor nhảy lên vùng dẫn, do đó nồng độ điện tử trong vòng dẫn là rất lớn ngay cả ở nhiệt độ thấp đối với việc hình thành các EHP. Với bán dẫn loại -p, vì các mức acceptor rất gần trên đỉnh vùng hóa trị nên một kích thích nhẹ có thể làm cho các điện tử trong vùng hóa trị nhảy lên chiếm các mức acceptor và để lại các lỗ trống trong vùng hóa trị. Do đó các bán dẫn loại -p có thể có nồng độ lỗ trống lớn ngay cả ở nhiệt độ thấp.

Khi một bán dẫn được pha tạp loại-n hoặc loại -p, một trong hai loại hạt tải sẽ chiếm ưu thế về nồng độ và được gọi là hạt tải cơ bản (hay majority carrier), loại hạt tải còn lại được gọi là hạt tải không cơ bản (hay minority carrier).

§1.3 Các quan hệ cơ bản và độ dẫn điện

Vì chuyển động của các hạt tải tạo ra sự dẫn điện, nên nồng độ hạt tải là đại lượng được quan tâm hàng đầu trong công nghệ IC. Với bán dẫn thuần, nồng độ điện tử trong vùng dẫn n bằng nồng độ lỗ trống trong vùng hóa trị p:

$$n = p = n_i \qquad (1.1)$$

trong đó n_i gọi là nồng độ hạt tải nội của bán dẫn thuần ở trạng thái cân bằng (hay trạng thái tĩnh).

Giả thiết các tạp chất phân bố đồng nhất. Để thỏa mãn điều kiện trung hòa điện tích (trung hòa tĩnh điện) trong bán dẫn thuần, các điện tích dương phải bằng các điện tích âm. Với silicon, các tạp chất hoặc thiếu hụt hoặc dư thừa một điện tử so với Si. Vì vậy:

$$P + N_D = n + N_A \qquad (1.2)$$

trong đó, N_D là nồng độ các nguyên tử donor và N_A là nồng độ các nguyên tử acceptor. Phương trình (1.2) còn gọi là điều kiện trung hòa điện tích không gian, trong đó đã giả thiết rằng tất cả các điện tử donor và các lỗ trống acceptor đều được kích thích hoàn toàn sao cho các mức donor và acceptor đều hoàn toàn bị chiếm bởi các điện tử. Ở nhiệt độ phòng, giả thiết này nói chung có thể chấp nhận được trừ khi pha tạp quá mạnh (nồng độ nguyên tử tạp chất > 10^{18} cm^{-3}). Nói cách khác, N_D có thể được thay thế bởi N_D^+ và N_A bởi N_A^-.

Ở trạng thái cân bằng nhiệt:

$$pn = n_i^2 \qquad (1.3)$$

Quan hệ này đúng cho các loại bán dẫn bất kỳ ở cân bằng nhiệt.

Với một bán dẫn loại ⁻n, nồng độ điện tử n_n có thể nhận được khi thay (1.3) vào (1.2):

$$n_n = \frac{1}{2}\left\{ N_D - N_A + \left[\left(N_D - N_A\right)^2 + 4{n_i}^2 \right]^{\frac{1}{2}} \right\} \qquad (1.4)$$

Tương tự cho bán dẫn loại ⁻p:

$$p_p = \frac{1}{2}\left\{ N_A - N_D + \left[\left(N_A - N_D\right)^2 + 4{n_i}^2 \right]^{\frac{1}{2}} \right\} \qquad (1.5)$$

Nồng độ hạt tải nội của Si là 4.5 x 10^{10} cm^{-3} ở 27° C, của GaAs là 9 x 10^6. Độ lớn của nồng độ tạp chất tổng cọng | N_D - N_A| nói chung lớn hơn rất nhiều so với n_i. Vì vậy nồng độ hạt tải cơ bản có thể được tính xấp xỉ từ (1.4) và (1.5):

$$n_n \approx N_D - N_A \qquad (1.6)$$

$$p_p \approx N_A - N_D \qquad (1.7)$$

Nồng độ hạt tải không cơ bản (thiểu số) có thể được tính xấp xỉ từ (1.6) , (1.7) và (1.3):

$$p_n \approx \frac{n_i^2}{N_D - N_A} \qquad (1.8)$$

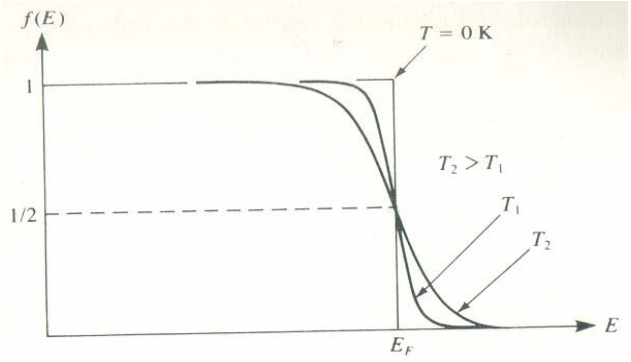
$$n_p \approx \frac{n_i^2}{N_A - N_D} \qquad (1.9)$$

trong đó p_n và là nồng độ lỗ trống trong bán dẫn n và n_p là nồng độ điện tử trong bán dẫn p.

Xác suất f(E) để một trạng thái điện tử với mức năng lượng E bị chiếm bởi một điện tử được cho bởi hàm xác suất Fermi-Dirac:

$$f(E) = \frac{1}{1 + e^{(E - E_F)/kT}} \qquad (1.10)$$

với T là nhiệt độ tuyệt đối, k là hằng số Boltzmann (8.62 x 10^{-5} eV/K = 1.38 x 10^{-23} J/K) và E_F được gọi là mức Fermi. Mức Fermi chính là thế hóa học của điện tử trong chất rắn, và có thể xem như mức năng lượng mà tại đó xác suất chiếm trạng thái của điện tử đúng bằng 1/2. Đồ thị hàm phân bố xác suất Fermi-Dirac cho các nhiệt độ khác nhau được minh họa ở hình (1-3):



Hình 1-3 phân bố xác suất Fermi-Dirac

Từ hàm phân bố xác suất Fermi-Dirac, số khả dĩ các điện tử trong bán dẫn có mức năng lượng xác định có thể được tính từ hàm mật độ xác suất N(E). Nếu số trạng thái năng lượng trên một đơn vị thể tích (hay mật độ trạng thái) ở trong khoảng năng lượng dE là N(E)dE, thì số điện tử trên một đơn vị thể tích (hay mật độ điện tử) trong vùng dẫn, n, được cho bởi:

$$n = \int_{E_c}^{\infty} f(E) N(E)\, dE \qquad (1.11)$$

Về nguyên tắc *N(E)* có thể được tính từ cơ học lượng tử và nguyên lý loại trừ Pauli. Tuy nhiên để tiện lợi có thể biểu diễn các điện tử phân bố trong vùng dẫn bởi *mật độ hiệu dụng* các *trạng thái* N_c định xứ tại bờ vùng dẫn E_c.Khi đó nồng độ điện tử trong vùng dẫn có dạng đơn gián:

$$N = N_c f(E_c) \qquad (1.12)$$

Trong đó N_c được cho bởi:

$$N_c = 2\left(\frac{2\pi m_n^* kT}{h^2}\right)^{3/2} \qquad (1.13)$$

$$= \begin{cases} 2.8\times10^{19}(T/300)^{3/2}\, cm^{-3} \text{ for } Si \\ 4.7\times10^{17}(T/300)^{3/2}\, cm^{-3} \text{ for GaAs} \end{cases}$$

trong đó m_n^* là khối lượng hiệu dụng của điện tử khi tính đến ảnh hưởng của mạng tinh thể lên đặc trưng của điện tử và h là hằng số Plank. Nếu $(E_c - E_f)$ lớn hơn một vài lần kT (thường ở nhiệt độ phòng kT = 0.026eV nên điều kiện này thỏa mãn), thi phân bố xác suất $f(E_c)$ có thể được tính gần đúng như sau:

$$f(E_c) = \frac{1}{1+e^{(E_c - E_F)/kT}} \approx e^{-(E_c - E_f)/kT} \qquad (1.14)$$

Khi đó (1.12) trở thành:

$$n = N_c e^{-(E_c - E_f)/kT} \qquad (1.15)$$

Tương tự:

$$p = N_v e^{-(E_f - E_v)/kT} \qquad (1.16)$$

$$N_v = 2\left(\frac{2\pi m_p^* kT}{h^2}\right)^{3/2}$$

Với

$$= \begin{cases} 1.04 \times 10^{19} (T/300)^{3/2} cm^{-3} \text{ for } Si \\ 7 \times 10^{18} (T/300)^{3/2} cm^{-3} \text{ for GaAs} \end{cases} \qquad (1.17)$$

m_p^* là khối lượng hiệu dụng của lỗ trống. Các phương trình (1.15) và (1.16) có hiệu lực cho cả bán dẫn thuần và pha tạp, chỉ thay E_F bằng E_I cho trường hợp bán dẫn thuần.

Nếu các hạt tải phân bố đều, mật độ dòng điện do sự dịch chuyển của các điện tử với vận tốc $\bar{v}_n$ trung bình theo một hướng nào đó (chẳng hạn hướng *x*) là:

$$J_n = qn\bar{v}_n \qquad (1.18)$$

Nếu các hạt tải phân bố không đều thì còn có thêm thành phần dòng khuếch tán:

$$J_n = qn\bar{v}_n - D_n(q)\frac{dn}{dx} \qquad (1.19)$$

Trong đó D là hệ số khuếch tán của hạt tải. Số hạng thứ nhất được gọi là dòng trôi (drift), tỷ lệ với cường độ điện trường E do vận tốc trung bình của các hạt tải tỷ lệ với cường độ điện trường E với hệ số tỷ lệ μ, đươc gọi là độ linh động:

$$\bar{v} = \mu E; \qquad \mu\left[\text{cm}^2/V.s\right] \qquad (1.20)$$

Với điện tử: $\bar{v}_n = -\mu_n E$,với lỗ trống: $\bar{v}_p = \mu_p E$

Độ linh động của hạt tải phụ thuộc vào nồng độ hạt tải và vào nhiệt độ. Nói chung độ linh động của điện tử lớn hơn độ linh động của lỗ trống. Với Si, ở nhiệt độ 20°C, μ_n = 1900 $cm^2/(V.s)$ và μ_p = 425 $cm^2/(V.s)$. Quan hệ (1.20) đúng với cường độ điện trường không quá lớn (thường nhỏ hơn 0.2V/cm). Với điện trường lớn hơn, độ linh động tăng chậm theo cường độ điện trường và tiến tới giá trị bão hòa. Dòng điện tổng cộng do cả hai loại hạt tải là:

$$J = J_n + J_p \qquad (1.21)$$

Từ (1.19) dễ thấy rằng độ dẫn điện:

$$\sigma = q(n\mu_n + p\mu_p) \qquad (1.22)$$

Hệ số khuếch tán trong (1.19) quan hệ với độ linh động theo hệ thức Einstein

$$D = \frac{kT}{q}\mu \qquad (1.23)$$

§1.4 Các đơn vị cơ sở của mạch tích hợp

Các đơn vị cơ sổ của Si-based Ics là MOSFET và BJT, và của GaAs-based ICs là MESFET. Một ứng dụng quan trọng của các tiếp xúc pn trong chế tạo IC là dùng để cách ly về điện cho nhiều loại phần tử tích cực. Với mục đích đó các tiếp xúc pn phải được áp đặt thế phân cực ngược hoặc bằng không. Ổ chế độ này chiều cao rào thế sẽ tăng khi tăng nồng độ pha tạp.

Các transistor có thể được dùng như các phần tử khuếch đại hoặc chuyển mạch. Trong cấu trúc ba lớp của BJT-transistor, lớp base (lớp giữa) rất mỏng và được pha tạp ít hơn so với emitter và collector. Vì vậy một dòng base rất nhỏ sẽ gây ra một dòng emitter-collector lớn hơn nhiều. Một BJT cách ly điển hình dùng cho các mạch tích hợp được mô tả ở hình (1.2).

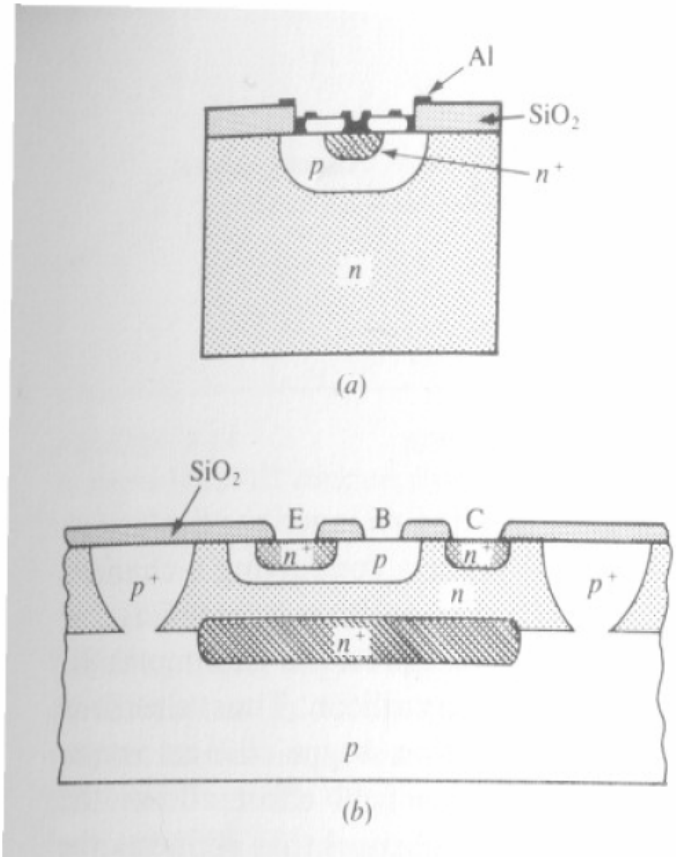


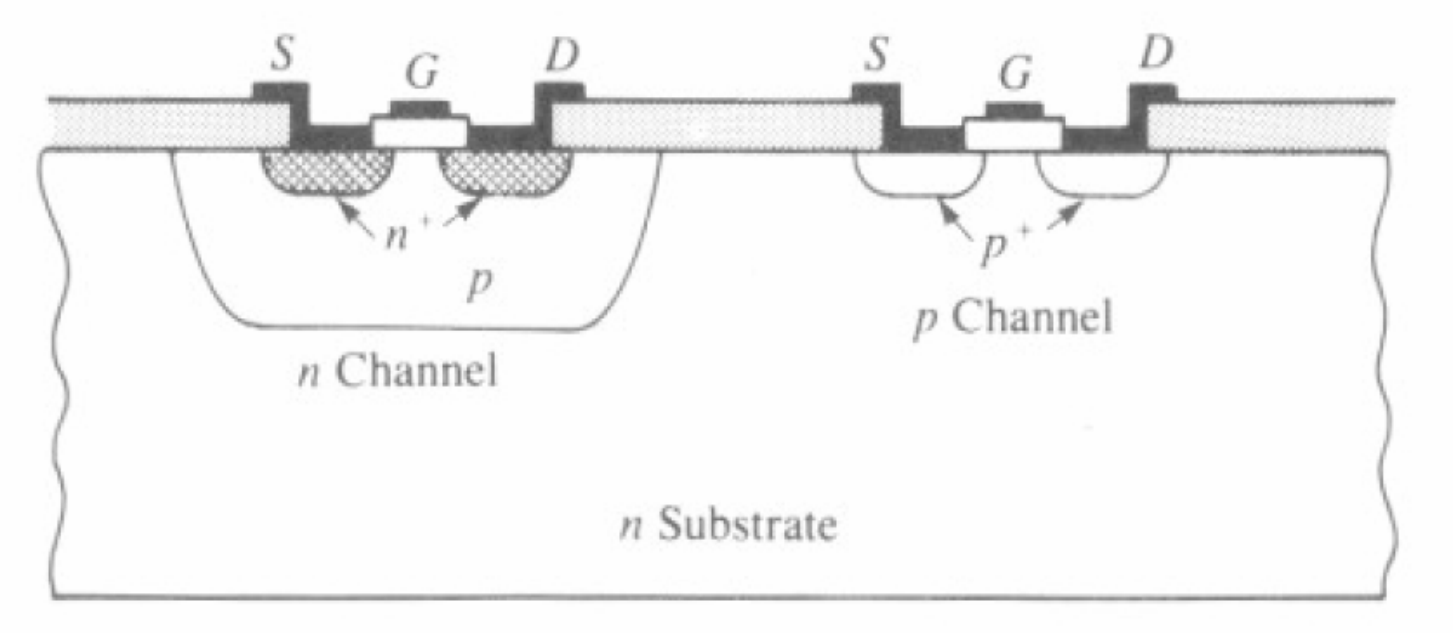
FIGURE 1-12
An *npn* transistor arrangement and a typical basic unit of *npn* bipolar junction IC transistors (Streetman, 1980; Hodges and Jackson, 1983).

Hình 1.2 Một đơn vị npn-BJT cơ bản dùng cho IC.

Vì cả ba cực đều phải ở trên bề mặt của chip, nên dòng collector phải chảy qua một đường dẫn có điện trở lớn trong vật liệu pha tạp nhẹ *n*. Một phương pháp chung để giảm điện trở collector là dùng một lớp pha tạp mạnh (n+) ngay bên dưới collector. Lớp n+ này được gọi là lớp ngầm (buried layer). Để cách

ly đơn vị BJT này với các đơn vị khác người ta dùng lớp đế p để tạo ra các chuyển tiếp pn cách ly. Các BJT loại npn được dùng nhiều vì công nghệ chế tạo đơn giản hơn so với pnp-BJT.

Transistor trường (FET) dựa trên công nghệ MOS chiếm ưu thế trong công nghệ IC, đặc biệt cho các IC logic. MOSFET có thể là kênh n hoặc kênh p tùy thuộc vào hạt tải cho sự dẫn điện là n hay p. Vì độ linh động của điện tử cao hơn nhiều so với lỗ trông nên MOSFET kênh n được dùng nhiều hơn. Một liên hợp có tính luân chuyển của NMOS và PMOS được gọi là CMOS (complimentary MOS), hình ( ).



Hình 1.3 Cấu hình CMOS đơn giản

Do khó khăn trong công nghệ chế tạo cấu trúc MOS cho GaAs nên MESFET là cấu trúc cơ sở cho IC trên cơ sở GaAs. Tuy nhiên các MESFET-IC trên cơ sở GaAs có tốc độ cao, mật độ tích hợp cao và độ rộng vùng cấm lớn. Một cấu trúc đơn giản của MESFET trên cơ sở GaAs được mô tả ở hình ().

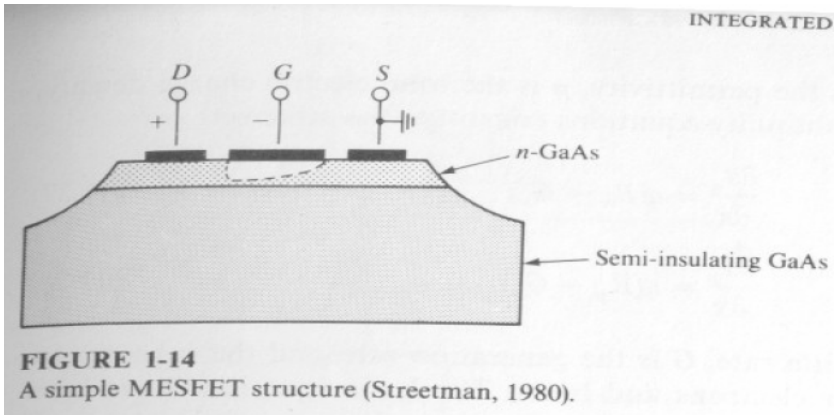
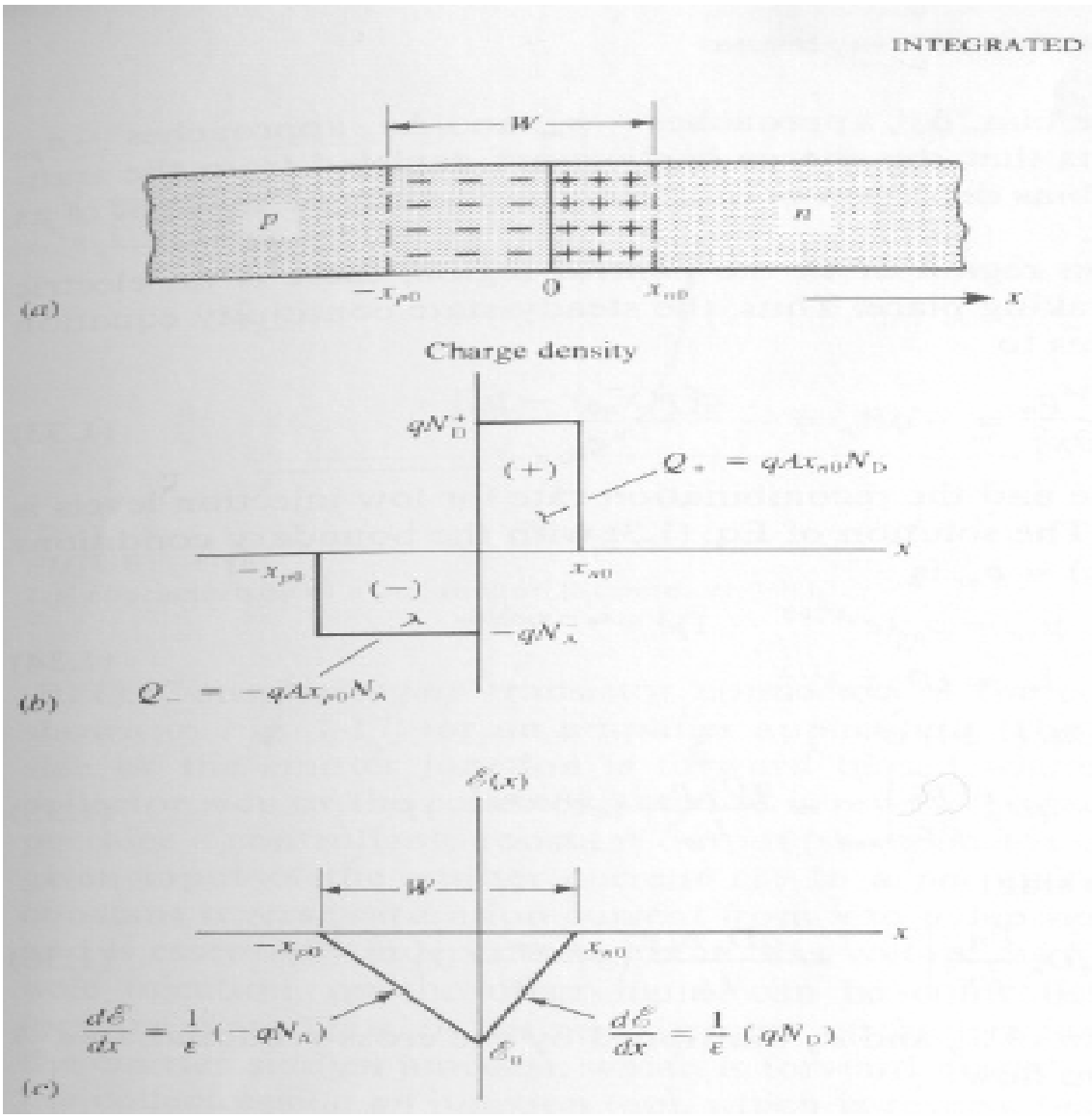


FIGURE 1-14
A simple MESFET structure (Streetman, 1980).

MESFET hoạt động với gate Schottky phân cực ngược và các tiếp xúc Ohmic cho drain và source. Đế là GaAs bán điện môi do pha tạp thích hợp, chẳng hạn Cl, sao cho mức Fermi được ghim ở gần giữa vùng cấm (do đó điện trở lớn).

§1.5 Một số cơ sở vật lý linh kiện bán dẫn



Nồng độ hạt tải vượt trội tại các bờ vùng điện tích không gian:

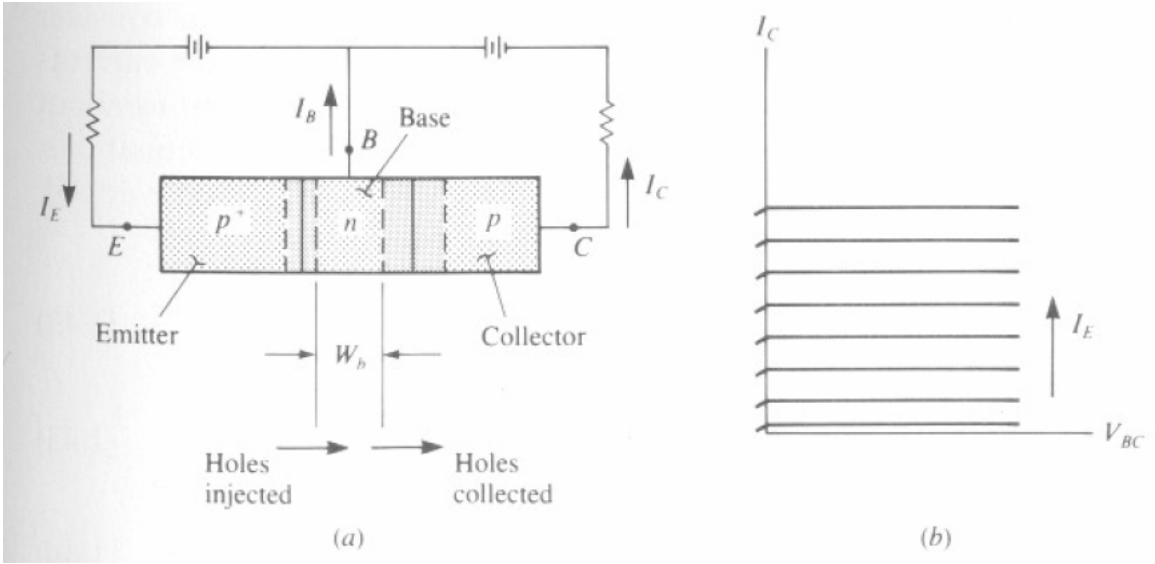
$$\Delta p_n = p(x_{n0}) - p_{ne} = p_{ne}\left(e^{qV/kT} - 1\right)$$

$$\Delta n_p = n(-x_{p0}) - n_{pe} = n_{pe}\left(e^{qV/kT} - 1\right)$$

Phương trình Shockley

$$\Delta I_0 = \left(\frac{qD_p p_{ne}}{L_p} + \frac{qD_n n_{pe}}{L_n} \right) A$$

$$L_p = \left(D_p \tau_p \right)^{1/2}, L_n = \left(D_n \tau_n \right)^{1/2}$$



Dưới thế phân cực ngược (C-B), dòng ngược từ n to p chỉ phụ thuộc vào tốc độ tiêm lỗ trống p được điều khiển bởi chuyển tiếp pn (Emitter-Base) phân cực thuận.

→ Good pnp Transistor cần gần như toàn bộ lỗ trống tiêm từ Emitter vào Base phải được góp vào Collector. → Base cần đủ mỏng sao cho neutral length của Base W_b nhỏ hơn nhiều so với quãng đường khuếch tán của lỗ trống (không xảy ra tái hợp trong vùng Base). Đồng thời dòng điện

tử từ Base đến Emitter phải nhỏ hơn nhiều so với dòng lỗ trống từ E đến B.
→Pha tạp miền B thấp hơn miền E (p^+n Emitter junction).

Các đại lượng quyết định tính năng của một BJT: hiệu suất tiêm Emitter, hệ số truyền đạt dòng, hệ số khuếch đại dòng base-collector.

$$\gamma \equiv \frac{I_{Ep}}{I_{En} + I_{Ep}}$$

$$\alpha \equiv \frac{I_C}{I_E} = \frac{I_{Ep}}{I_E}\frac{I_C}{I_{Ep}} = \gamma \frac{I_C}{I_{Ep}}$$

$$\beta \equiv \frac{I_C}{I_B} = \frac{\alpha}{1-\alpha}$$

- Chỉ cần giải phương trình trung hoà cho miền Base vì các dòng được xác định bởi đặc trưng của hạt tải trong 2 miền chuyển tiếp quanh Base.
- Khi các thế phân cực lớn và Emitter pha tạp mạnh thì:

$$I_B = a_1 \tanh\left(\frac{W_b}{2L_p}\right)$$

$$I_C = a_1 \operatorname{csch}\left(\frac{W_b}{L_p}\right)$$

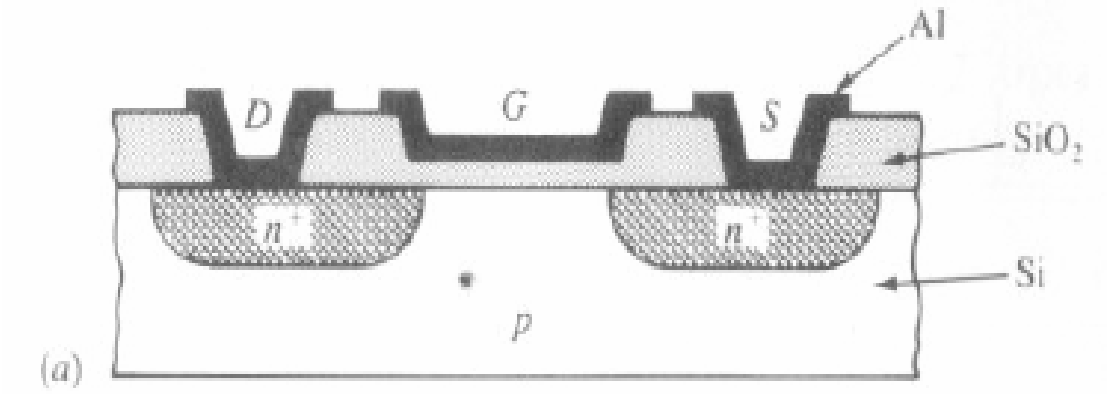
$$I_E = a_1 \coth\left(\frac{W_b}{L_p}\right)$$

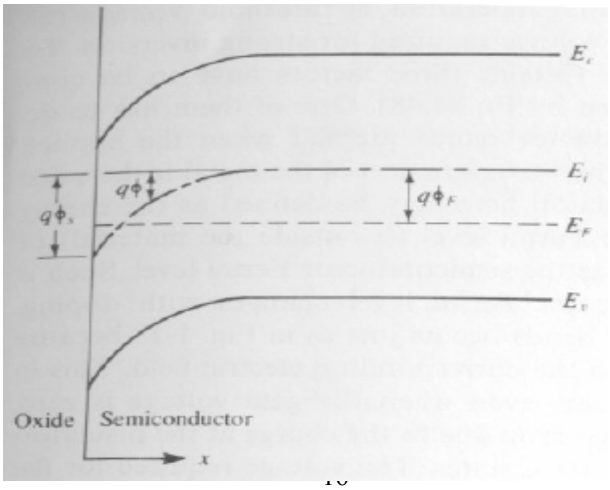
$$a_1 = \frac{qAD_p \Delta_{p_E}}{L_p}, \Delta_{p_E} = p_{Be}\left(e^{qV_{EB}/kT} - 1\right)$$

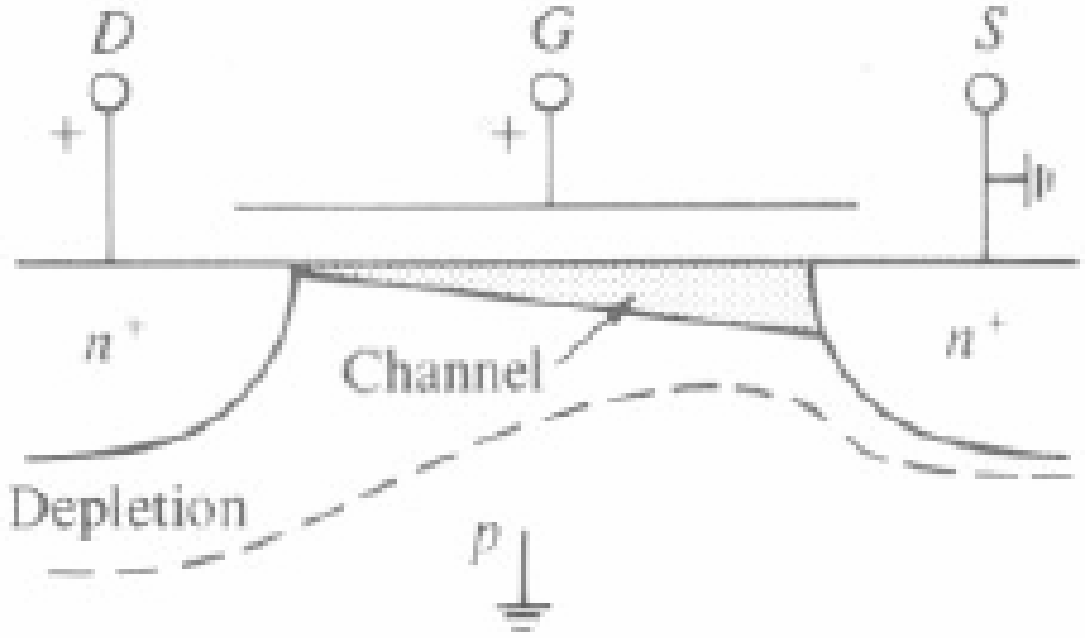
L_p là chiều dài khuếch tán trong miền Base và p_{Be} là nồng độ lỗ trống cân bằng trong miền Base.

- Ba yếu tố quan trọng:
  - Thế phân cực (số hạng exp(qV/kT)
  - Các dòng Emitter và Collector được xác định bởi gradient nồng độ hạt tải không cơ bản tại biên của chuyển tiếp.
  - Dòng Base bằng hiệu dòng Emitter và Collector

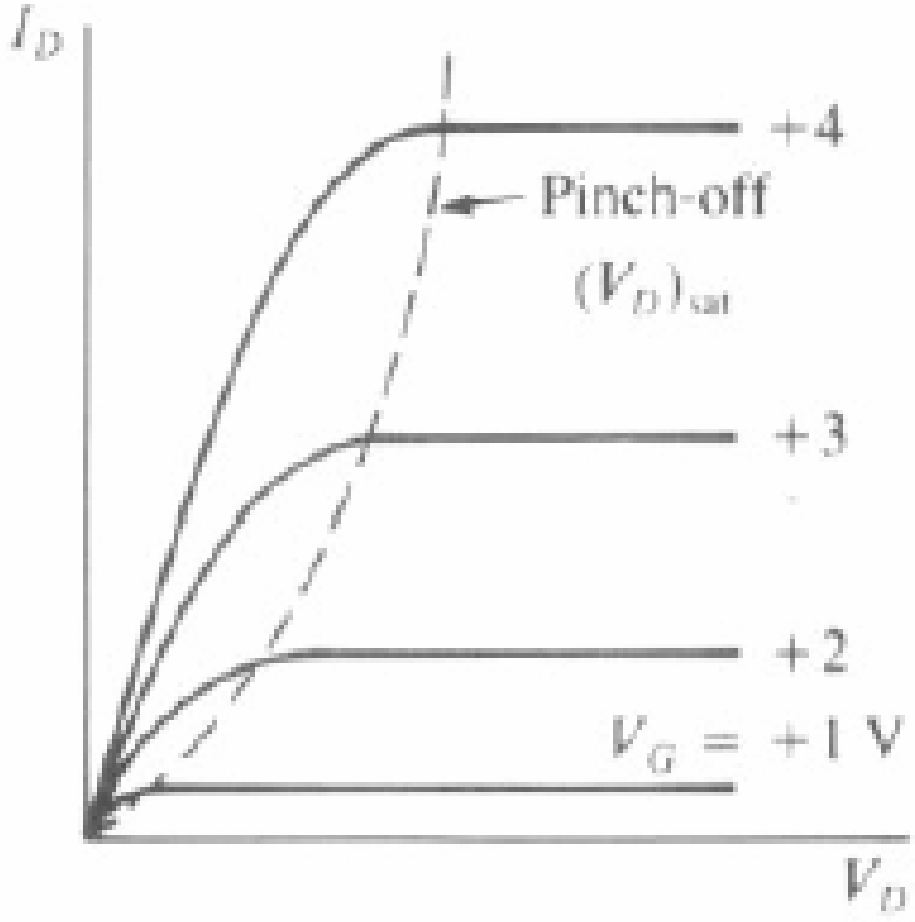
<u>Cấu trúc MIS:</u> đặc biệt quan trọng cho digital ICs.







(b)



(c)

Đặc trưng I-V:

$$I_D = \frac{\overline{\mu}_n Z C_i}{L}\left[(V_G - V_T)V_D - \frac{1}{2}V_D^2\right]$$

Z: chiều sâu của kênh, L: chiều dài kênh, C_i : điện dung lớp cách điện trên đơn vị diện tích, $\overline{\mu}_n$ độ linh động bề mặt của điện tử.

§1.6 Ví dụ thiết kế BJT

Phần này sẽ xem xét một thiết kế cho việc chế tạo một BJT với một lớp ngầm như đã nói tới ở phần trước. Tuần tự thiết kế và chế tạo chưa được đề cập ở đây. Hình (1.6.1) là sơ đồ của một n+pn+ BJT.

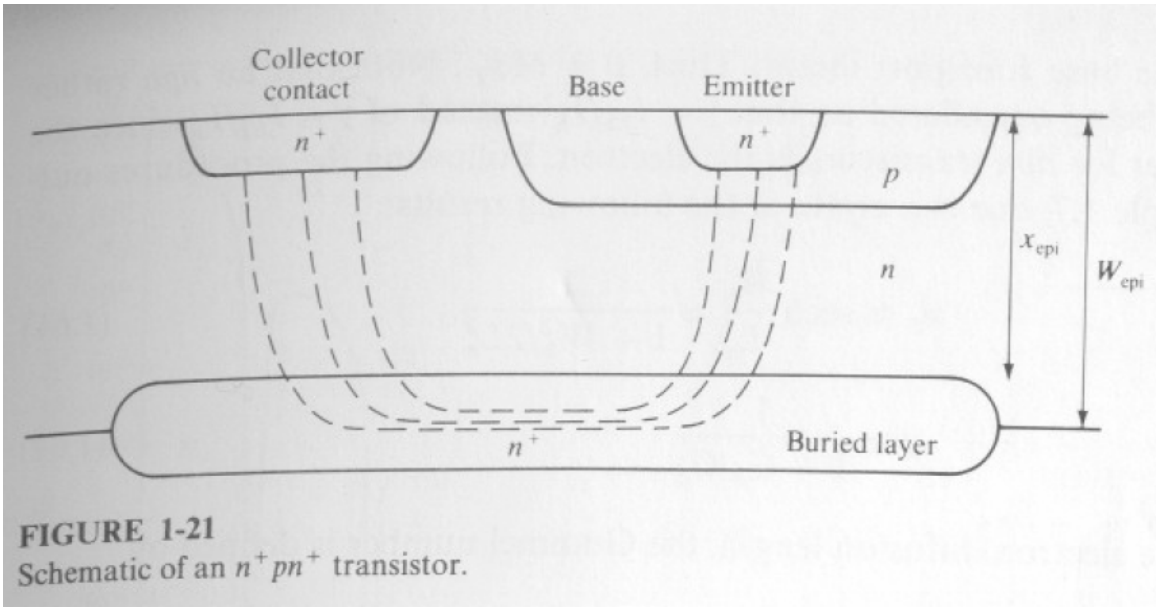


FIGURE 1-21
Schematic of an n^+pn^+ transistor.

Các thông số quan trọng là hệ số khuếch đại dòng base-collector, β, tần số cutoff, là tần số ứng với sự suy giảm của hệ số khuếch đại ac về đơn vị, tần số cắt alpha, f_α, liên quan với thời gian dịch chuyển của hạt tải thứ yếu qua miền base τ_B, tương ứng với sự suy giảm 3 dB của độ lợi so với giá trị của nó ở tần số thấp:

$$f_\alpha = 1/(2\pi\tau_B)$$

và:

$$\tau_B = \frac{W_b^2}{\eta D_n}$$

trong đó η là hệ số phụ thuộc vào mức pha tạp (=2 cho base pha tạp đồng nhất), và vào điện trường áp đặt.

Ngoài ra còn có hai tiêu chuẩn cho sự hoạt động bình thường của transistor. Một là thế đánh thủng. Dưới điều kiện phân cực ngược, có hai nguyên nhân gây ra hiện tượng đánh thủng. Một là tunnel do điện trường cảm ứng (thường giữa hai miền pha tạp mạnh, hiệu ứng Zener). Hai là đánh thủng thác lũ, do các cặp điện tử-lỗ trống được tạo ra do các hạt tải được gia tốc bởi điện trường. Thế đánh thủng (BV) thường liên quan với hệ số nhân collector như sau:

$$M = \frac{1}{1 - \left[V_{CB} / (BV)_{CB\circ}\right]^n}$$

Trong đó n là hằng số và $(BV)_{Cbo}$ là thế đánh thủng hở mạch.

Độ lợi dòng α được biểu diễn bởi:

$$\alpha = \frac{I_C}{I_E} = \frac{I_C}{I_{Cn}} \frac{I_{Cn}}{I_{En}} \frac{I_{En}}{I_E} = M \alpha_T \gamma$$

Trong đó α_T là hệ số vận chuyển base,

$$\alpha_T \approx \operatorname{sech}\frac{W_b}{L_n} \approx \frac{1}{1+W_b^2/2L_n^2}$$

$$\gamma \approx \frac{1}{1+G_B/G_E}$$

Trong đó L_n là quãng đường khuếch tán của điện tử, số Gummel G_B được xác định bởi:

$$G_B = \frac{1}{D_{nB}} \int_{base} N_B(x)dx = \frac{Q_{Bo}/q}{D_{nB}} \qquad (1.6.1)$$

và G_E được xác định tương tự cho emitter, Q_{Bo} là điện tích tiếp xúc của base. Dấu xấp xỉ ứng với điều kiện $W_b >> L_n$.

$$\beta \approx \frac{1}{G_B/G_E + W_b^2/2L - \left[V_{CB}/(BV)_{CBo}\right]^n}$$

Nếu giả thiết có một phân bố Gauss của nồng độ tạp chất (kết quả của ủ nhiệt), với nồng độ tại bề mặt là N_{Bo} thì:

$$N_B = N_{Bo}\,\mathrm{exp}(-\frac{x^2}{4Dt}) - N_C$$

Với N_C là nồng độ tạp ở collector (bên phía base của chuyển tiếp collector-base), khi đó:

$$\frac{Q_{Bo}}{q} = \left(\frac{Dt}{2}\right)^{1/2} N_{Bo}\left(\int_0^{t_2} e^{-y^2/2}dy - \int_0^{t_1} e^{-y^2/2}dy\right) - \int_{x_{EB}}^{x_{CB}} N_C dx$$

Với

$$t_1 = \frac{x_{EB}}{(2Dt)^{1/2}}$$

$$t_2 = \frac{x_{CB}}{(2Dt)^{1/2}}$$

Trong đó x_{EB} và x_{CB} là khoảng cách từ bề mặt tới mép của các miền emitter-base và collector-base tương ứng. Khi đó:

$$(BV)_{CBo} = \frac{q}{\varepsilon_S}\left(\frac{Q_{Bo}x_1}{q} - \frac{N_C x_1^2}{2} + \frac{Q_{Bo}W_b}{2q}\right) \qquad (1.6.2)$$

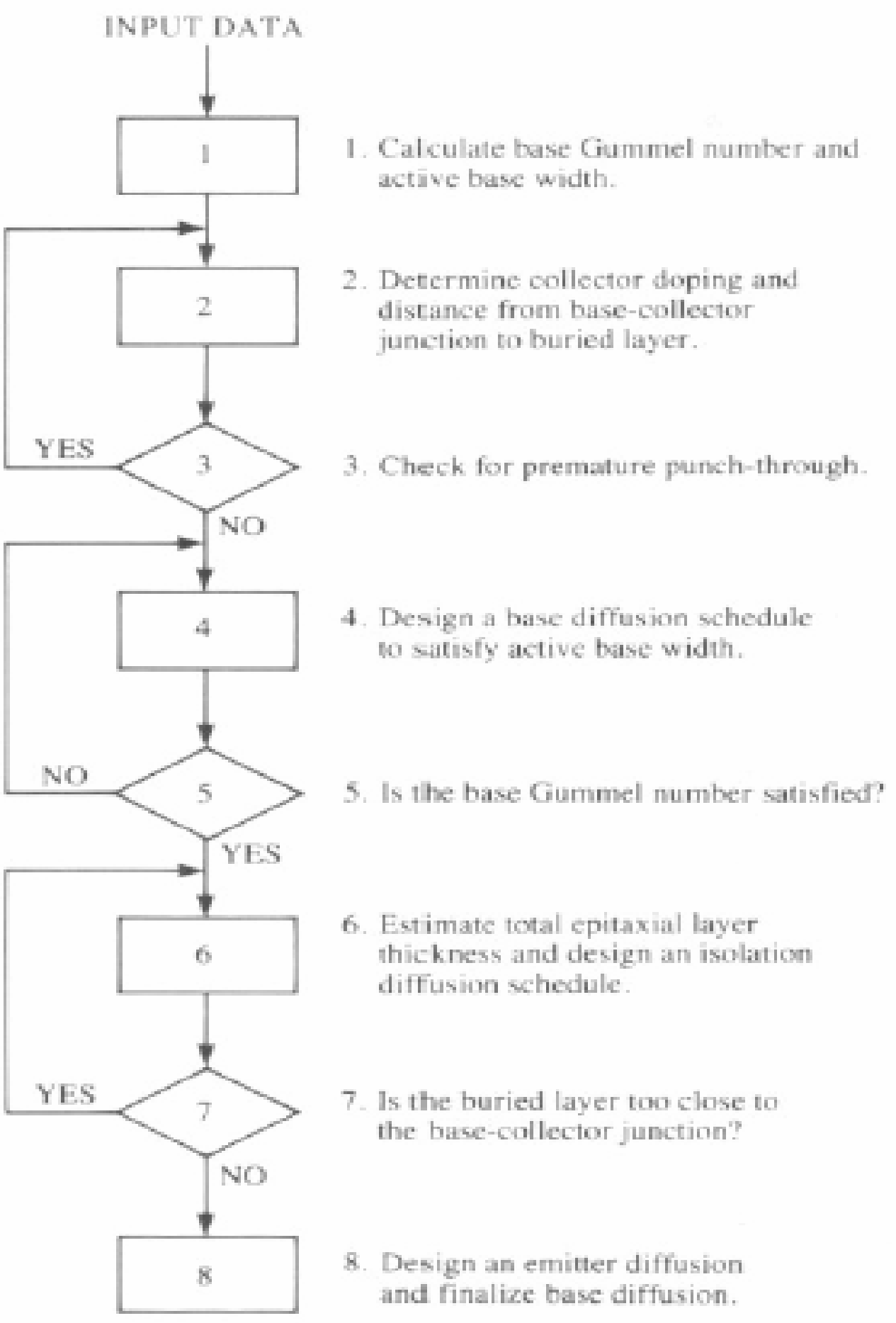
Với $x_1 = x_{epi} - x_{BC}$

Các phương trình (1.6.1 và 1.6.2) dùng để xác định công nghệ chế tạo. Vấn đề có thể được phát biểu dưới dạng: cho các giá trị mong muốn của β, f_T, $(BV)_{CBo}$, và R_{SB} (trở kháng sheet của base), cần xác định kích thước và cách thức pha tạp cho việc chế tạo BJT như hình (1.6.1). Bài toán và lời giải được tóm tắt trong bảng sau.

Design specifications for the example (Colclaser, 1980)

Design values:	$\beta_F = 45$	$\omega_a = 2\pi \times 5 \times 10^9$ rad/s	
	$(BV)_{CBo} = 25$ V	$R_{sB} = 200\ \Omega/\square$	
Dimensions:	$W_{epi} = 6.4\ \mu$m	$x_{epi} = 3.1\ \mu$m	
	$x_{jC} = 1.45\ \mu$m	$x_{jE} = 0.67\ \mu$m	
	$x_B = 0.49\ \mu$m	Buried layer $R_s = 25\ \Omega/\square$	

Diffusion processes	T, °C	t, min	R_s, $\Omega/\square$	N_o, cm^{-3}
Isolation				
Predeposit	950	39		1.55×10^{20}
Drive-in	1200	60	50	5.5×10^{18}
Base				
Predeposit	900	17.4		1.2×10^{20}
Drive-in	1100	23.4	200	6.5×10^{18}
Emitter	950	47.8	~20	8.4×10^{20}



Lưu đồ thuật toán cho bài toán

chế tạo:

1. Chọn một giá trị tiêu biểu của G_E ($5x10^{13}$ cm^{-4}). Giả thiết β thỏa mãn (có thể kiểm tra lại sau), tính G_B:

$G_B = G_E/\beta = 5x10^{13}/45 = 1.11x10^{12}$ $cm^{-4}s$

Nồng độ pha tạp trung bình ở base thường là 10^{17} cm^{-3}.

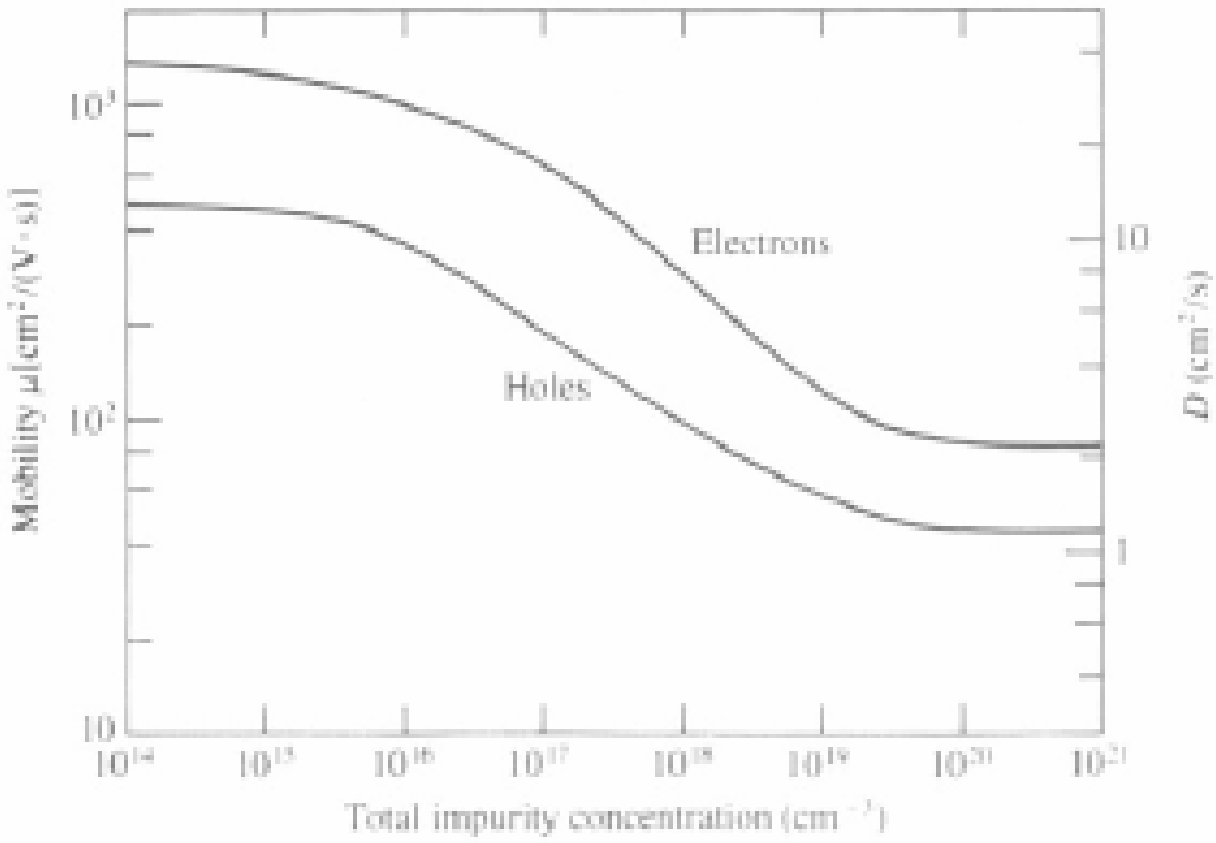


FIGURE 1-7
Mobility as a function of total dopant concentration (Grove, 1967).

Từ hình 1-7 đọc được D_{nB} là 15 cm^2/s. Từ (1.63) suy ra:

$Q_{Bo}/q = D_{nB}G_B = 1.67x10^{13}$ cm^{-2}

Từ 1.57 và 1.58 với η = 4 và giải cho W_b suy ra:

$W_b = (\eta D_{nB}/2\pi f\alpha)1/2 = 0.49$ μm

2. Giả thiết chuyển tiếp base-collector định xứ ở khoảng 2 μm tính từ bề mặt (x_{jC}), thế đánh thủng $(BV)_{Cbo}$ = 25 V, khi đó dùng hình 1-24 để tìm ra:

$X_{epi} - x_B = x1 \approx 1.2$ μm

Nồng độ pha tạp collector tối đa cho phép là $N_C = 8x10^{15}$ cm^{-3}

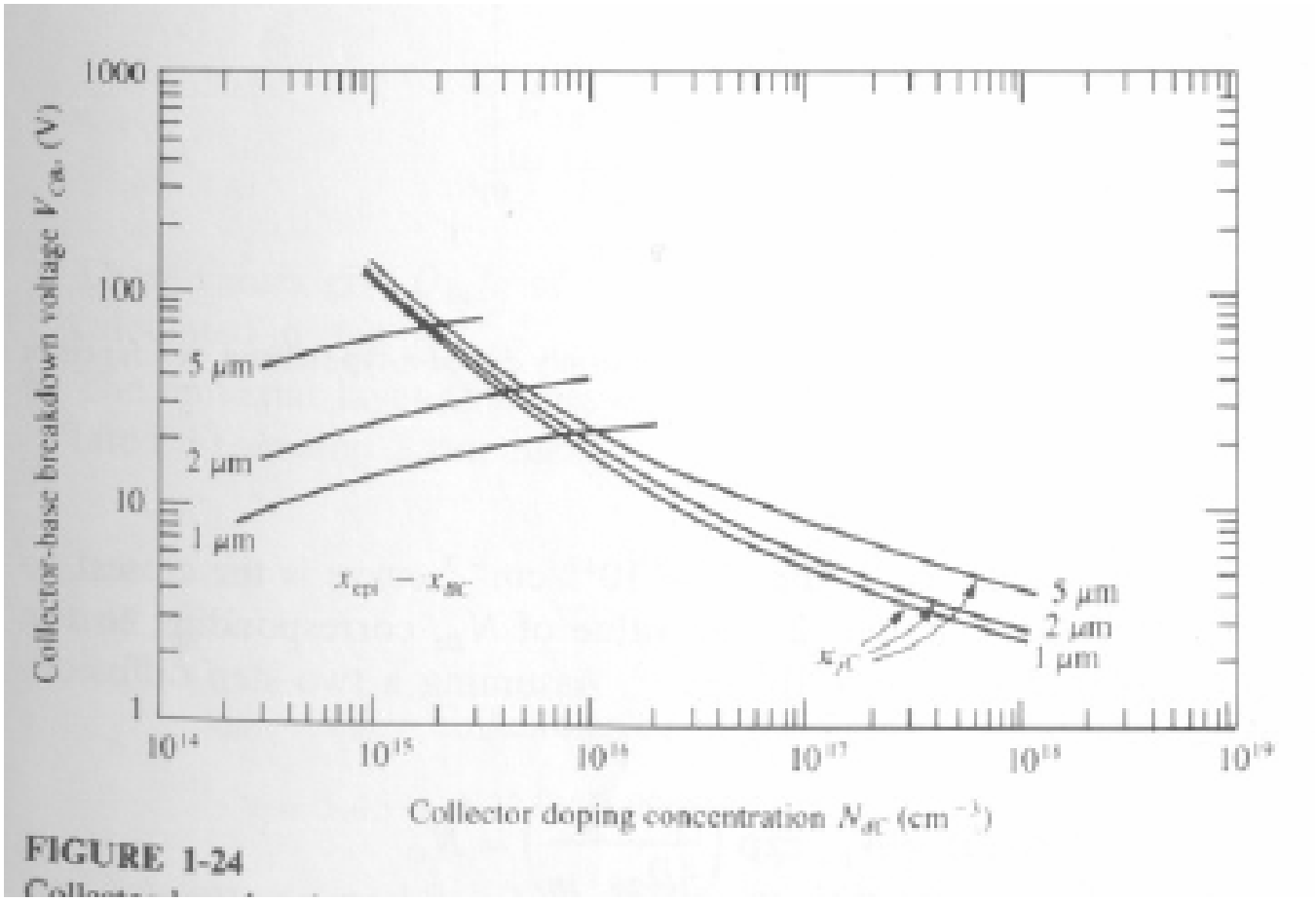


FIGURE 1-24

Hình 1-24

3. Kiểm tra thế punch - through sử dụng (1.6.2): $(BV)_{Cbo}$ = 372 V. Vì giá trị đã chọn là 25 V nên có thể chuyển sang bước 4

4. Với quá trình khuếch tán nhiệt theo phân bố Gauss, dùng hình 1-25 để nhận được giá trị N_{bo}.

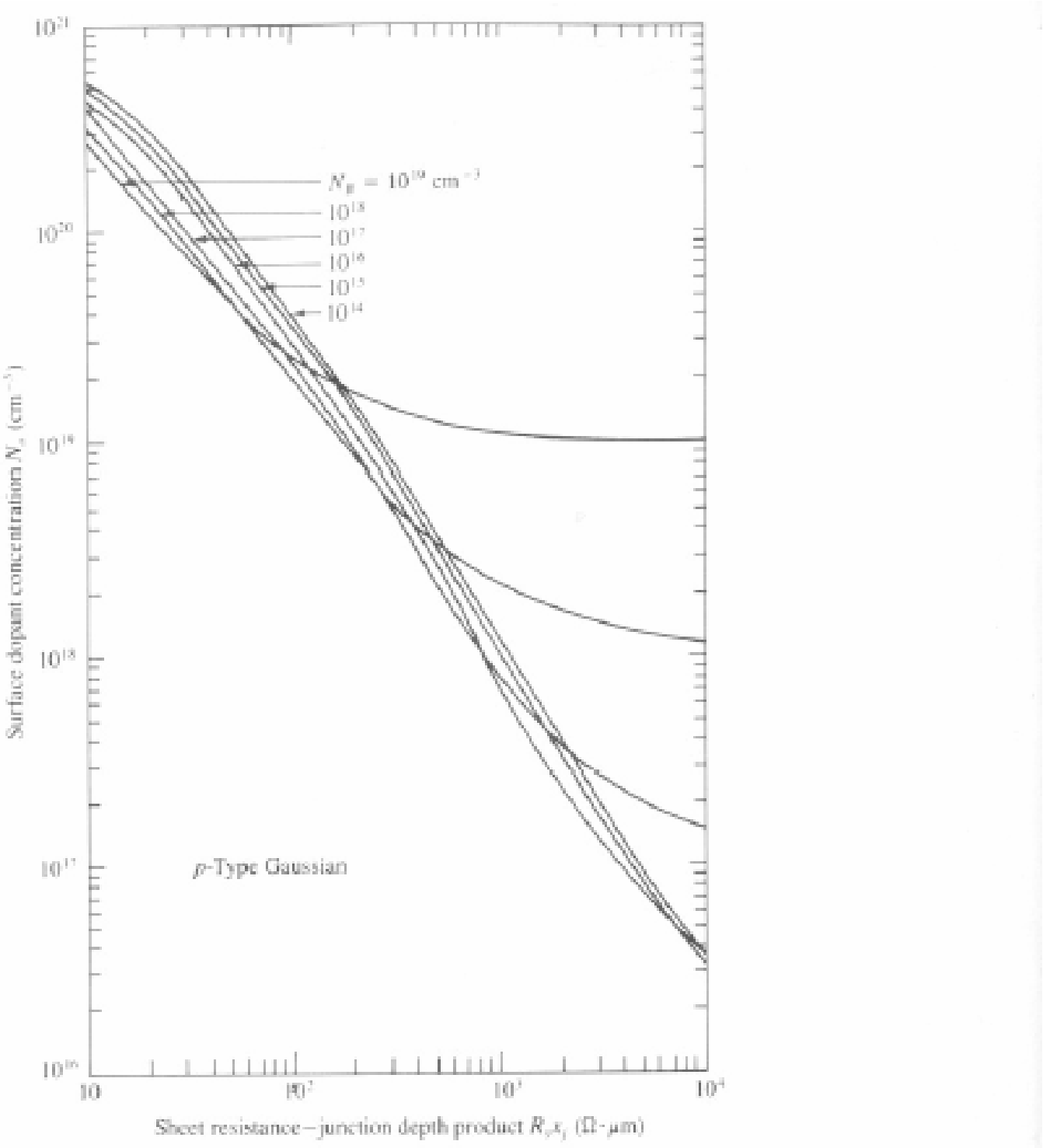


FIGURE 1-25
Surface dopant density of a *p*-type Gaussian diffusion in uniformly doped *n*-type silicon as a function of average resistivity at 300 K (Irvin, 1962).

Trong hình 1-25, đường $N_B = 10^{16}$ cm^{-3} gần với giá trị của N_C đã tính từ bước 2. Giá trị tìm được là $N_{bo} = 4x\ 10^8$ cm^{-3}. Giả thiết quá trình khuếch tán là 2 -step, chuyển tiếp xảy ra khi

$N(x_j,t) = N_{bo} \exp(-x_{jC}^2/4D_{2B}t_{2B}) = N_C$

Hoặc $D_{2B}t_{2B} = 1.6x10^{-9}$ cm^2.

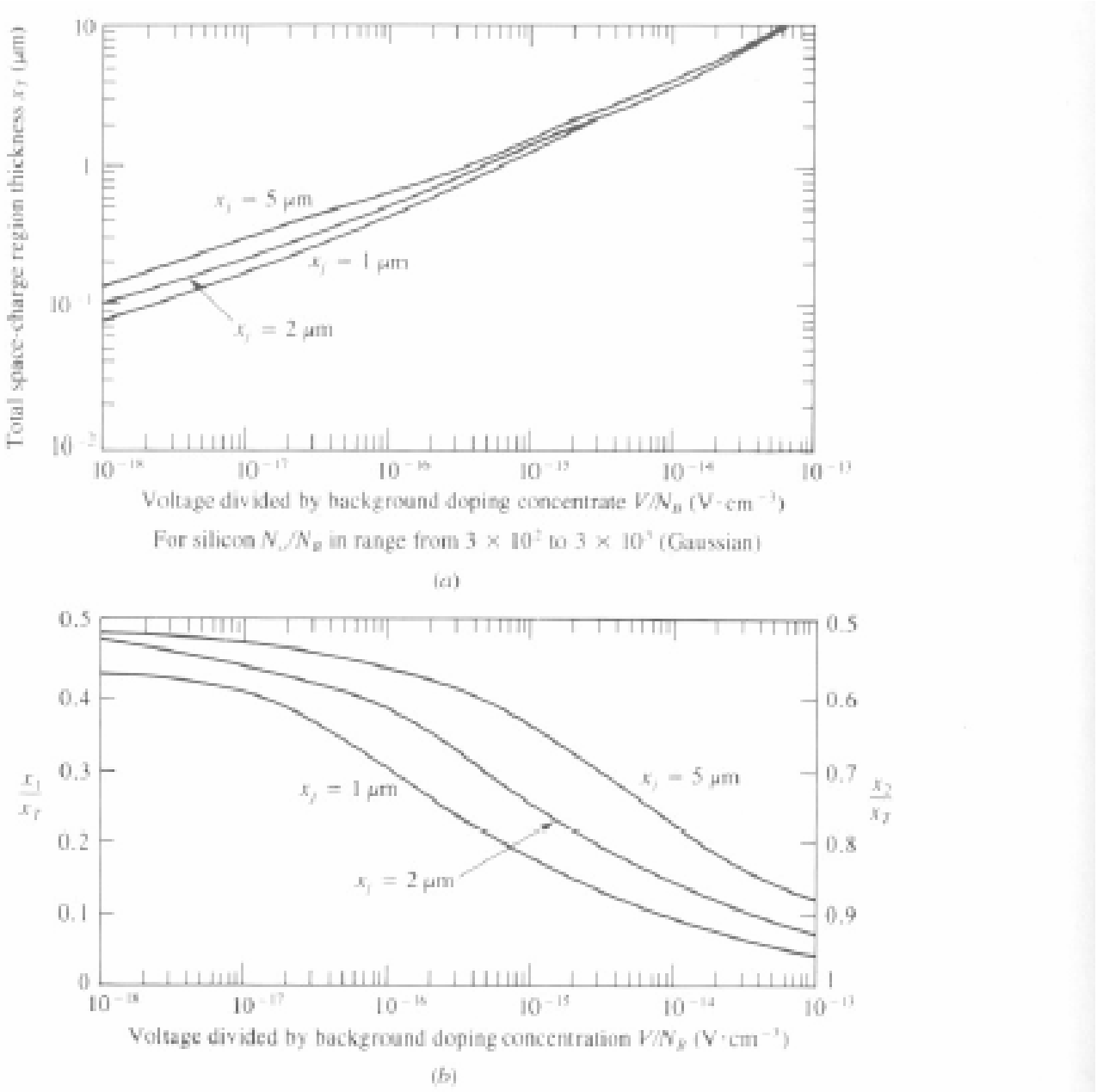


FIGURE 1-26
Space-charge region thickness as a function of voltage for a *pn* junction formed by a Gaussian diffusion into a constant background concentration: (*a*) total width x_T and (*b*) ratios x_1 and x_2 where x_1 is the portion in the heavier doped side and x_2 is that in the lighter doped side (Lawrence and Warner, 1960).

5. Bây giờ giá trị Q_{bo}/q đã tính từ trước có thể được kiểm tra nhờ hình 1-26 cho độ rộng của vùng điện tích không gian base - collector (x_T) và các độ rộng tương ứng của các miền pha tạp mạnh (x_1) và pha tạp nhẹ (x_2). Nếu giá trị tính được nhờ hình 1-26 lớn hơn giá trị đã tính trước, thì x_{jC} phải tăng và bước 4 được lặp lại cho đến khi điều kiện trên thỏa mãn. Với chuyển tiếp chưa bị áp đặt thế phân cực thì V =0.7 V. Khi đó :

Độ rộng toàn bộ miền điện tích không gian = 0.5 μm

Phía base = 0.19 μm

Phía collector = 0.31 μm

Các biên của base tích cực, x_{CB} và x_{EB} là:

$x_{CB} = 2 - 0.19 = 1.81$ μm

$x_{EB} = x_{CB} - W_b = 1.81 - 0.49 = 1.32$ μm

Dùng 1.66 cho kết quả: $Q_{bo}/q = 5.3 \times 10^{12}$ cm-2, Giá trị này nhỏ hơn giá trị 1.67×10^{13} đã nhận được từ bước 1. Do đó lặp lại các bước 4 và 5 cho các giá trị:

$x_{jC} = 1.45$ μm

$D_{2B}t_{2B} = 7.85 \times 10^{-10}$ cm^2

$N_{bo} = 6.5 \times 10^{18}$ cm^{-3}

Các giá trị này cho kết quả $Q_{bo}/q = 1.66 \times 10^{13}$ cm^{-2}, rất gần với giá trị đã tính ở bước 1.

6. Chiều dày lớp epitaxi W_{epi} và xepi có thể được tính. Từ bước 2 có:

$$X_{epi} - x_{BC} = 1.2\ \mu m$$

Và $x_{BC} = x_{jC}$ + độ rộng phía collector của chuyển tiếp B-C

$$= 1.45 + 0.31 = 1.76\ \mu m$$

Do đó $x_{epi} = 1.2 + 1.76 = 2.96$ μm. Nếu giả thiết sự phủ trước của lớp ngầm vào lớp epitaxi trong quá trình epitaxi, thì:

$$0.9W_{epi} = x_{epi} + x_2$$

với x_2 là chiều sâu thâm nhập của lớp ngầm vào lớp epitaxi trong quá trình khuếch tán cách ly. Để thử nghiệm, đặt $x_2 = 2$ μm, khi đó $W_{epi} = 5.5$ μm. Giả thiết điện trở sheet khuếch tán cách ly là 50 Ω/□, nồng độ bề mặt $N_{io}(N_{co})$ là 7×10^{18} cm^{-3} (từ hình 1-25). Dùng giá trị này cùng với phân bố Gauss có thể tính $X_2 = 2.44$ μm.

7. Giá trị tính được của x_2 lớn hơn giá trị thử (2μm). Điều này chứng tỏ lớp ngầm quá gần chuyển tiếp B-C và do đó bước 6 được lặp lại. Các giá trị mới là:

$$W_{epi} = 6.4\ \mu m$$

$$X_{epi} = 3.1\ \mu m$$

$$N_{io} = 5.5 \times 1018\ cm^{-3}$$

8. Vị trí tính từ bề mặt của chuyển tiếp emitter, x_{jE} cần phải được xác định. Phần base của miền B-E dưới thế phân cực thuận là xấp xỉ 0.1 μm. Khi đó:

$$X_{jE} = x_{jC} - x_{CB} - W_b - 0.1$$

$$= 1.45 - 0.19 - 0.49 - 0.1 = 0.67\ \mu m$$

Cuối cùng, nồng độ tạp tổng cộng tại x_{jE} phải bằng 0. Điều này ngụ ý rằng:

$$N_E(x_{jE}) = N_B(x_{jE}) - N_C(x_{jE})$$

Hay $N_E = N_{bo} \exp(-x_{jE}^2/4D_{2B}t_{2B}) - N_C = 1.55 \times 10^{18}\ cm^{-3}$

Phương trình này có thể dùng để quyết định thủ tục pha tạp.

§1.7 Các giai đoạn chính của qui trình chế tạo vi điện tử

Hình () mô tả các quá trình cơ bản của công nghệ chế tạo IC, bao gồm quá trình tinh chế vật liệu, quá trình mọc tinh thể và chuẩn bị các wafer trên đó IC được chế tạo, quá trình chế tạo linh kiện (IC), quá trình đóng kiện, lưu giữ và kiểm tra. Mặt sau của các chip (die) được gắn cơ học hoặc nôi với môi trường gá giữ thích hợp, thường là plastic hoặc ceramic.

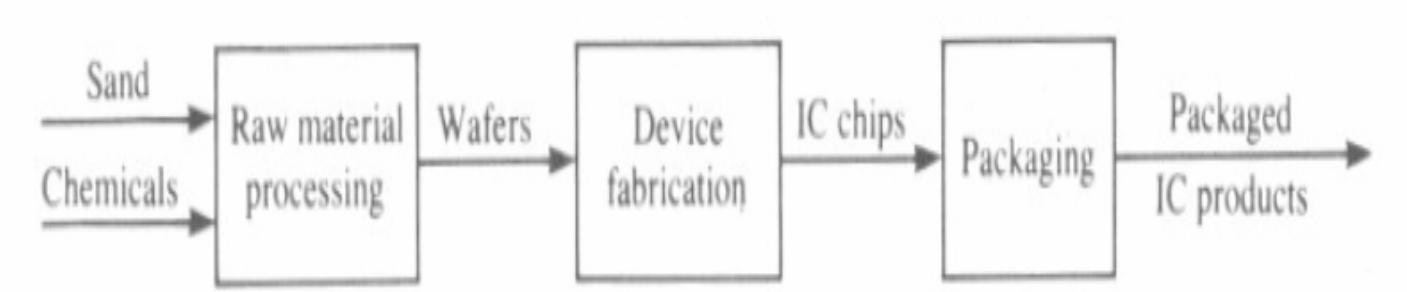


FIGURE 1-27
A broad view of microelectronics processing.

Các wafer thường là các đĩa mỏng (chẳng hạn 0.5 mm với Si) của vật liệu đơn tinh thể pha tạp donor hoặc acceptor. Yêu cầu độ sạch của tạp và độ hoàn hảo của cấu trúc tinh thể là rất nghiêm ngặt. Một qui trình xử lý têu biểu được mô tả ở hình 1-28.

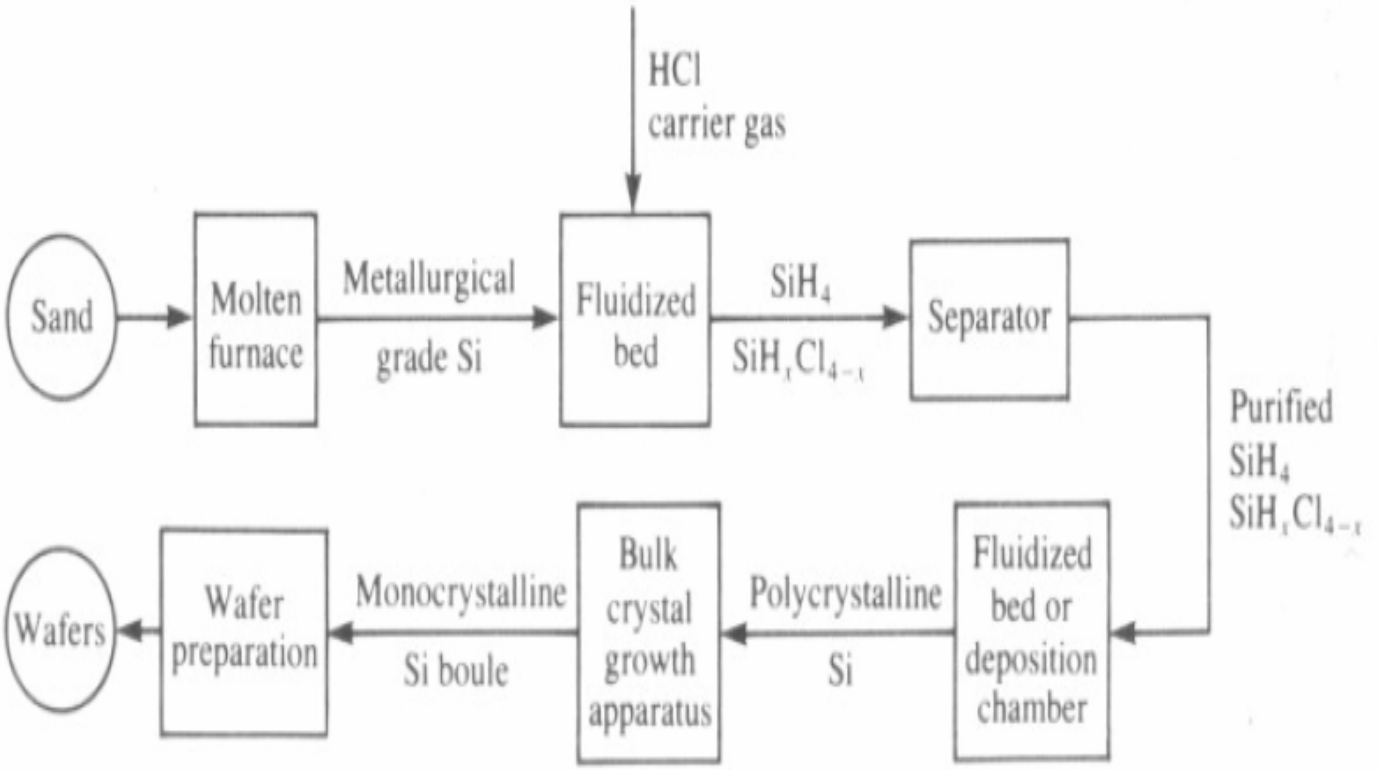


FIGURE 1-28
Processing steps for silicon wafers.

Vật liệu khởiđầu có thể là cát hoặc một khoáng chất của Si. Một lò hồ quang điện nóng chảy được dùng để tạo ra Silic có độ sạch cấp luyện kim (metallurgical grade silicon, MGS), độ sạch khoảng 98%. Các hạt MGS được đưa vào lò phản ứng lỏng với một khí tải chứa hydrochloric acid để chuyển MGS thành các khí chứa Si như silane và chlorosilane. Các khí được tách và làm sạch qua một dãy các bộ tách và chưng cất. Có hai phương pháp chính để tạo ra Si sạch cấp độ điện tử (electronic grade,

EGS). Một phương pháp bao gồm sự phủ Si từ khí chứa Si lên trên một ống Si nóng (ống Si nóng tạo ra các vị trí kích cỡ nguyên tử). Ống Si lớn rất nhanh đến đường kính 20 cm. Nếu dùng khí trichlorosilane, thì phản ứng xảy ra như sau:

$SiHCl_3(gas) + H_2(gas) \rightarrow Si(solid) + 3HCl(gas)$

Phương pháp thứ hai, có nhiều ưu điểm hơn, sử dụng phản ứng trong bể lỏng. Trong đó khí chứa Si và các hạt mầm Si tinh khiết được nuôi. Si EGS có cấu trúc đa tinh thể và chứa tạp chất có nồng độ trong khoảng ppm (nhỏ hơn 20 ppm). Tiếp theo Si EGS sẽ được cho nóng chảy để nuôi thành thỏi đơn tinh thể.

Có 3 phương pháp chính để nuôi thỏi đơn tinh thể từ Si EGS. Phương pháp được sử dụng rộng rãi nhất là kỹ thuật Czochralski. Trong phương pháp này, một hạt đơn tinh thể mầm nhỏ được nhúng trong EGS nóng chảy, và tinh thể mầm sẽ được kéo gradual sao cho thỏi đơn tinh thể được có đường kính 15 cm được hình thành từ quá trình làm nguội. Một phương pháp khác là phương pháp nóng chảy vùng. Trong phương pháp này, một thỏi Si đặt theo phương thẳng đứng được làm nóng chaỷ cục bộ từ dưới lên sử dụng lò cục bộ quét từ dưới lên (chẳng hạn lò vi sóng). Vùng nóng chảy được tái tinh thể hóa nhờ các tinh thể mầm. Phương pháp thứ ba là phương pháp Bridgeman, được dùng chủ yếu cho GaAs. Trong đó vật liệu đa tinh

thể được làm nóng chảy dọc theo một thuyền hẹp, dài nhờ lò quét dọc. Và được làm nguội từ một phía có gắn với tinh thể mầm.

Quá trình nuôi đơn tinh thể có hai mục tiêu. Một là chuyển cấu trúc đa tinh thể thành cấu trúc đơn tinh thể. Hai là loại bỏ các tạp chất không mong muốn. Quá trình này xảy ra ở mặt phân cách rắn - lỏng. Trong quá trình nuôi đơn tinh thể, các tạp chất có thể được đưa vào để tạo ra đơn tinh thể bán dẫn loại n hoặc p. Các thỏi bán dẫn sẽ được cắt thành các phiến mỏng, đường kính 0.5 mm (wafer). Mặc dù chỉ một phần rất mỏng của đĩa bán dẫn được dùng để tích hợp linh kiện, nhưng độ dày của đĩa bảo đảm cho sự ổn định cơ học của IC.

So với công nghệ chế tạo wafer thì công nghệ chế tạo linh kiện phát triển nhanh hơn nhiều. Một số công nghệ có thể trở thành lạc hậu trước khi được công bố. Tuy nhiền nền tảng và mục tiêu của chúng có thể chưa thay đổi.

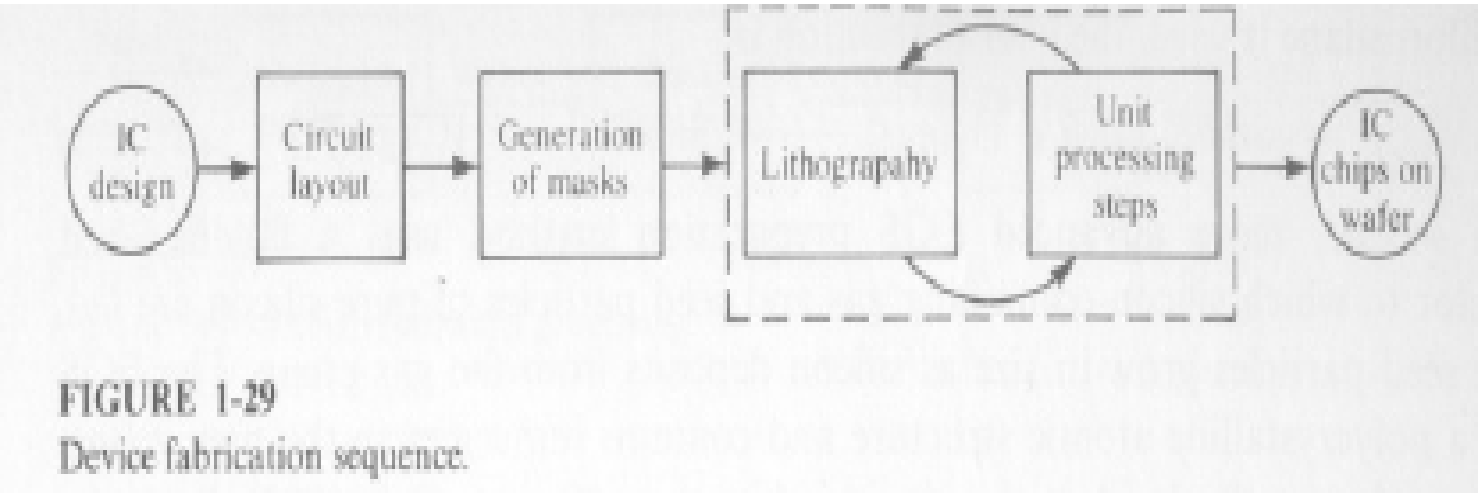


FIGURE 1-29
Device fabrication sequence.

Hình 1-29 mô tả các bước tuần tự của qui trìnhchế tạo IC. Khi cho trước các yêu cầu và các chi tiết kỹ thuật của mạch thì IC có thể được thiết kế dưới dạng circuit layout với các chi tiết về chiều rộng, chiều sâu của mỗi một đơn vị cơ bản. Chẳng hạn với một MOSFet đơn giản, sơ đồ bố trí tổng thể (layout) sẽ được chuyển thành các sơ đồ khổ lớn cho mỗi một mức mặt nạ. Sau đó các sơ đồ này sẽ được thu nhỏ lại để thu được mặt nạ cuối cùng. Với các mạch VLSI, người thiết kế có thể mô tả một cách điện tử toàn bộ layout của mạch. Sau đó thiết kế được chuyển thành dạng số và được lưu trữ trên băng từ. Mặt nạ có thể bao gồm nhiều mức khác nhau cho các chế tạo khác nhau. Các mặt nạ được làm từ các vật liệu như chromium, chromium oxide, hoặc silicon.

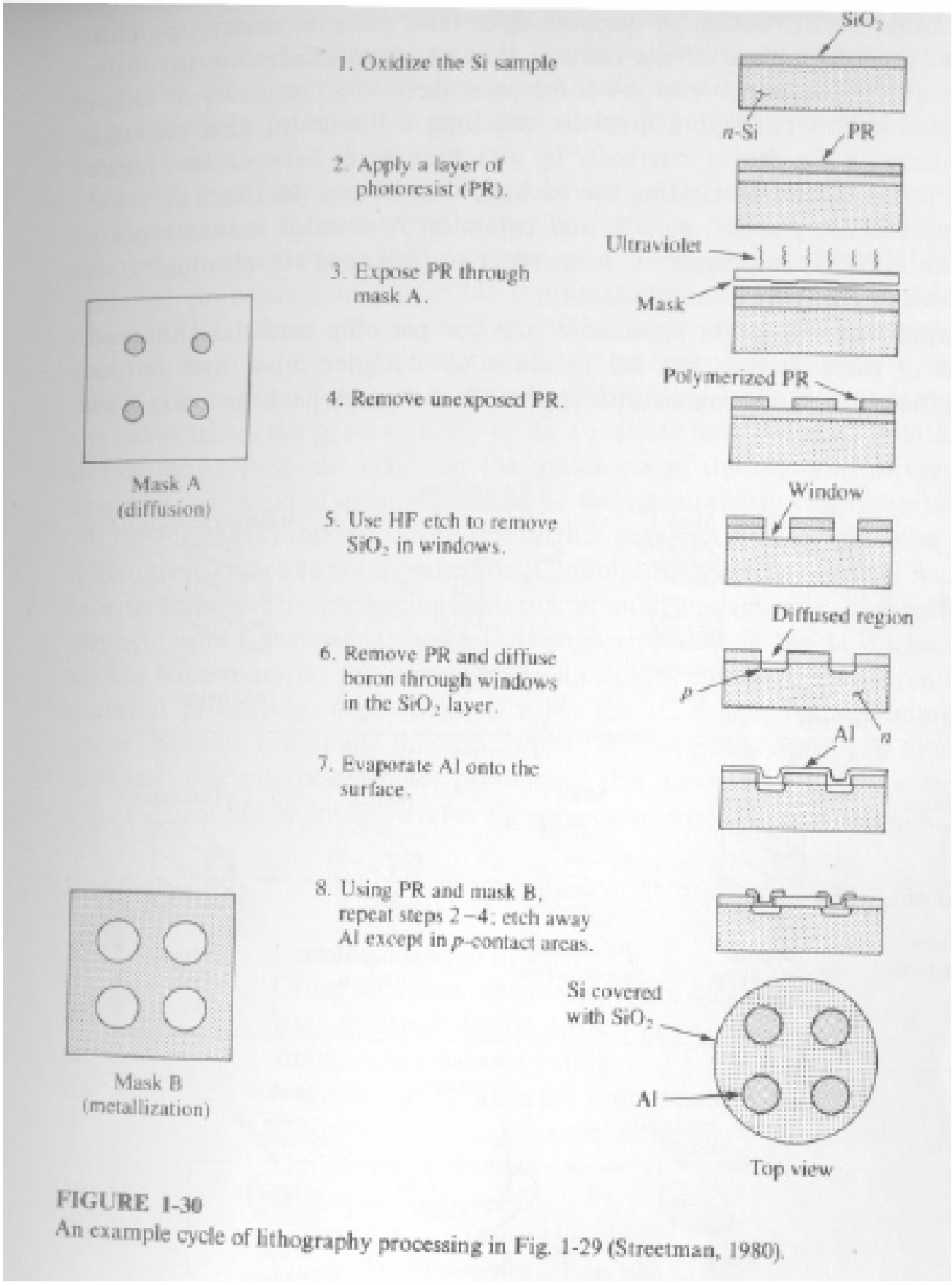


FIGURE 1-30
An example cycle of lithography processing in Fig. 1-29 (Streetman, 1980).

Bước tiếp theo là quá trình quang khắc (lithography), có thể được lặp lại nhiều lần cho các bước xử lý tương ứng với các mức mặt nạ khác nhau. Quang khắc là quá trình chuyển dạng hình học trên mặt nạ vào bề mặt của Si wafer. Mỗi chu trình quang khắc thường bao gồm sự ăn mòn qua các cửa sổ mở hoặc vạch các dạng nhất định cho bước tiếp theo, chẳng hạn như phủ các màng mong muốn hay đưa tạp chất vào các vùng đã được mở nhờ quá trình khuếch tán hoặc cấy ion. Các bước tiêu biểu của quá trình quang khắc được mô tả trong hình 1-30. Các mặt nạ được dùng để mở các cửa sổ qua lớp silicon dioxide sao cho tạp chất dưới dạng khí có thể khuếch tán qua đó. Một lớp cản quang (photoresist, PR) từ các vật liệu polymer nhạy sáng được phủ lên trên mặt lớp SiO2. Mặt nạ được đặt ên trên lớp PR và được chiếu tia cực tím. Các chỗ lộ sáng sẽ bị polymer hóa, còn các chỗ bị thì không. Các vùng không bị polymer hóa sẽ bị ăn mòn bởi acid BHF (buffered-HF), để lộ ra các cửa sổ cho quá trình khuếch tán. Các wafer sẽ được đặt vào lò để khuếch tán có chưa khí tải B2H6 hoặc PH3để tạo ra các miền pha tạp mong muốn.

Các bước xử lý đơn vị tiêu biểu bao gồm sự phủ màng epitaxi (đơn tinh thể) và màng không epitaxi, sự oxi hóa, cấy ion và kim loại hóa (phủ kim loại). Quá trình này nhằm mục tiêu tạo ra các miền tích cực và cách ly chúng với nhau. Yếu tố chính ở đay là sự phân bố tạp chất và sự mô tả rõ

ràng của phân bố tạp chất. Các wafer phải trải qua nhiều bước khác nhau của quá trình xử lý ở nhiệt độ cao nên có thể dẫn tới sự phân bố lại tạp chất. Do đó rất nhiều nỗ lực đã được thực hiện để hạ thấp nhiệt độ của các quá trình xử lý.

Quá trình đóng kiện mạch IC bao gồm việc gắn chip (die) vào một vật liệu gá nhất định, việc nối dây giữa các đường dẫnlinh kiện với gá và tạo hình cho nó. Các vật liệu gá thường là plastic, kim loại và gốm.

Hình 1-31 mô tả quá trình chi tiết chế tạo một NMOS Si - gate IC từ các wafer ban đầu đến đóng kiện cho IC. Khi mật độ linh liện tăng lên thì số chân của IC trở thành một nhược điểm do kích thước bộ gá lớn sẽ làm tổn hao nhiệt lớn. Đồng thời số chân của IC nhiều cũng làm tăng tín hiệu nhiễu. Bên cạnh đó thời gian sử dụng (lifetime) cũng là một yếu ttó quan trọng trong việc thiết kế gá giữ IC.

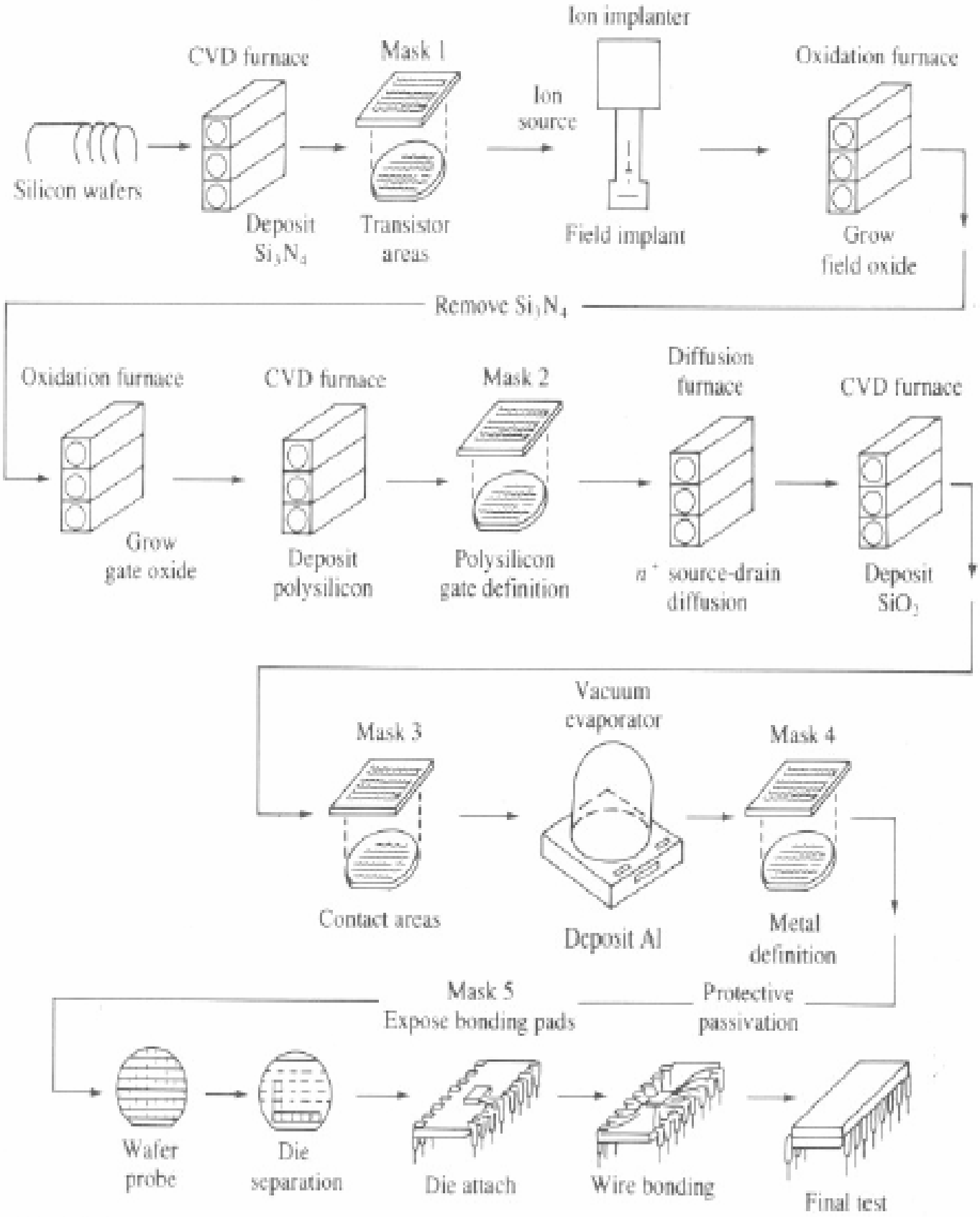


FIGURE 1-31
A manufacturing process for NMOS silicon-gate ICs (Integrated Circuit Engineering Company).

CHAPTER 2

TECHNOLOGY

2.0 INTRODUCTION

A good understanding of processing and fabrication technology on the part of the circuit designer is necessary to provide the flexibility needed to optimize integrated circuit designs. With this knowledge the actual layout can be considered during design and the appropriate parasitics can be included in the analysis. Innovative techniques that improve performance often involve circuits or geometries that are dependent on and applicable to a particular process. Knowledge of processing characteristics enables the designer to make yield calculations during design and consider tradeoffs between yield, performance and design simplicity.

In this chapter processing technology is discussed from a qualitative viewpoint. This is followed by a detailed discussion of typical NMOS, CMOS, bipolar, thick film, and thin film processes. Most processes in industry can be viewed as either a straightforward variant or extension of these processes. These processes are summarized in the appendices of this chapter. Included in the appendices are process scenarios, graphical process descriptions, design rules, process parameters, and some computer simulation model parameters. These appendices should provide a useful reference for material that is presented in later chapters of this book. This chapter is concluded with a discussion of practical layout considerations and comments about some CAD tools that have become an integral part of the IC design process.

2.1 IC PRODUCTION PROCESS

The major steps involved in producing integrated circuits are considered from a qualitative viewpoint in this section. These steps are used in the MOS and/or bipolar processes that will be discussed later in this chapter.

2.1.1 Processing Steps

CRYSTAL PREPARATION. The substrate of bipolar and MOS integrated circuits is generally a single crystal of silicon that is lightly doped with either n- or p-type impurities. The substrate serves both as the physical medium upon and within which the IC is built and as part of the electrical circuit itself. These crystals are sliced from large right-circular cylinders of crystalline silicon, which are carefully grown to lengths up to 2 m and which vary in diameter from 1 to several inches. The slices are typically 250 μ to 400 μ thick. From an electrical viewpoint much thinner slices would be acceptable; however, the thicker slices have been adopted because they are more practical to handle (less breakage) and are less likely to warp during processing. The size of the wafers has been increasing rapidly with time to allow for both large chips and a larger number of chips per wafer. As of 1989, many of the older processing lines were using 4 inch wafers, but the newer lines are typically using 5 and 6 inch wafers. The crystals are often cut so that the surface is oriented approximately in the ‹100› direction.

MASKING. IC masks are high-contrast (black on clear) photographic positives or negatives. They are used to selectively prevent light from striking a photosensitized wafer during the photolithographic process. The masks are typically made of glass covered with a thin film of opaque metal, although less costly and less durable emulsion masks are sometimes used. The masks are produced from a digitized description of the desired mask geometries. There are several different methods of generating the masks (called pattern generation) from the digitized circuit description. One method involves photographically reducing large copies of the desired patterns that have been generated with a computer-controlled drafting machine. This method was used widely in the past but has largely been replaced by the next two. A second uses a laser beam as a pattern generator in a raster-scan mode. Both of these methods generally also require a high-resolution step and repeat and/or reduction camera to make the final masks that will be used. The intermediate image that is created is called a reticle and is usually 5 or 10 times real size. A third method uses an electron beam (E-beam) to generate the actual patterns directly onto the final masks. This method produces the best quality masks and is used extensively for very small geometries, but it requires considerable time and expensive equipment.

PHOTOLITHOGRAPHIC PROCESS. Photoresist is a viscous liquid. It is applied in a thin, uniform layer (about 1 μ thick by spinning the wafer) to the entire surface of a wafer following cleaning. After application the photoresist is hardened by baking. The physical characteristics of the photoresist can be changed by exposure to light. The photoresist thus acts as a film emulsion and can be exposed by light through the transparent areas of a mask (either by contact printing or projection), by a projection of light through a reticle containing the same information (called direct step on wafer), or by an electron beam (E-beam) that scans the desired regions. Following exposure, the resist is developed to selectively remove the resist from unwanted areas. This step is often followed by another baking to further harden the remaining photoresist.

Both positive and negative photoresists are available. With negative photoresist the unexposed areas are removed during development, and with positive resist the exposed areas are removed. Negative resists are noted for being quite unaffected by etchants used in processing, but finer resolution can typically be obtained with positive resists. Photoresists serve as protective layers to many etchants and oxidizing agents, and as a barrier to ion implants.

Proper mask alignment is essential to maintain device operation, characteristics, and yield. Alignment markings are generally included with the circuit information when the masks are made so that these marks will appear on the wafer during and after processing. A machine called a mask aligner is used to align and expose the wafers. Figure 2.1-1 shows typical alignment characteristics. The physical size and geometry of the masks used for fabrication is governed by the particular technique used by the mask aligner to expose the wafer. Mask aligners that use contact printing have multiple copies of the individual circuits at actual size (i.e., 1×) accurately patterned on the mask. These aligners have a large throughput and are relatively inexpensive. The large masks, however, have a very short lifetime (typically 3 to 10 exposures) because of damage incurred when the mask contacts the photoresist for exposure. This increases effective mask costs. The direct step on wafer aligners typically use a 5× mask (often called a reticle) as a negative. It typically will contain only a single copy of the circuit, though several copies may be used for small ICs. The image is optically reduced to 1× upon exposure. The wafer must be repeatedly moved to the next location after each exposure until the entire wafer is exposed. The lifetime of the mask is very long since no physical contact is made, but the throughput has been decreased considerably to allow for the successive wafer movements. The

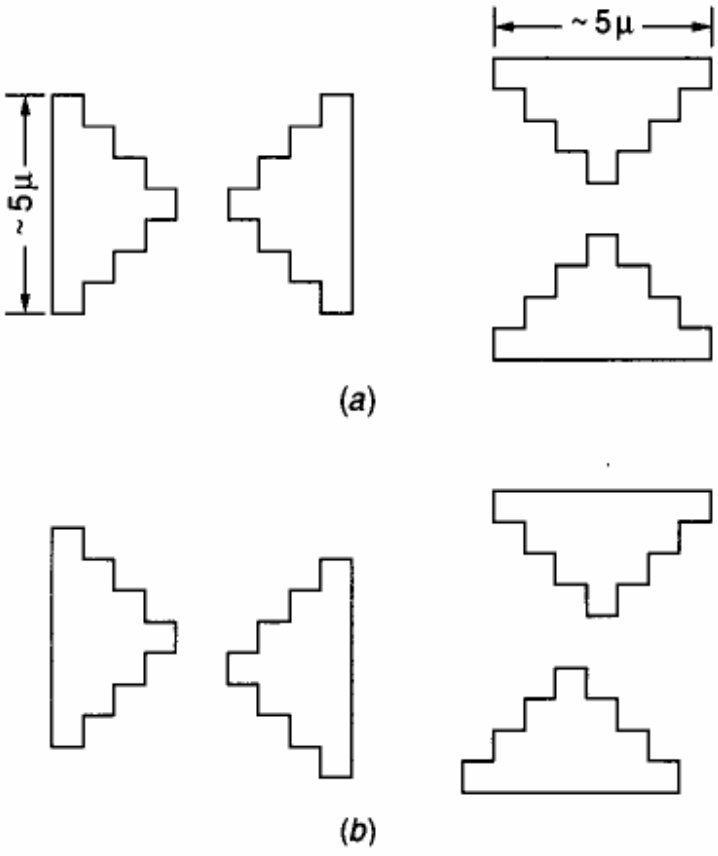


FIGURE 2.1-1
Alignment marks: (*a*) Mask marks, (*b*) Simulated positioning of alignment marks after fabrication.

equipment is also considerably more expensive because of the precision needed to maintain consistent and repeated uniform stepping of the wafer. Both types of aligners are widely used in industry.

One of the most practical and popular methods of exposure actually combines the mechanical economics of the 1× aligners and the mask life of the steppers. In this approach, a thin protective membrane, called a pelicle, is placed above the emulsion of 1× chrome masks for protection of the mask and long mask life. Although the membrane itself may get dirty or scratched, it is placed far enough away from the mask so as to remain out of focus and thus not project defects onto the wafer when columniated light is focused through the mask onto the wafer.

A fourth method of exposure actually uses no masks at all. Instead, a narrow electron beam (E-beam) is selectively focused on the wafer in a raster-scan manner in small regions, with wafer stepping to position successive portions of the wafer under the beam. The same digital database that is used to generate masks can be used to drive the E-beam system. This approach gives better resolution than any of the previously discussed methods but involves very expensive equipment and has a much smaller throughput. It is practical for only the most demanding applications.

DEPOSITION. Films of various materials must be applied to the wafer during processing for most existing semiconductor processes. Often these films are very thin (200 Å or less for some SiO_2 layers) but may be as thick as 20 μ for "thick film" circuits. Films that are deposited include insulators, resistive films, conductive films, dielectrics, n- and p-type semiconductor materials, and dopants that are subsequently forced deeper into the substrate. Deposition techniques include physical vapor deposition (evaporation and sputtering), chemical vapor deposition (CVD), and screen printing for the thick films. With the exception of the screen-printed films, the depositions are nonselective and are placed uniformly over the entire wafer.

Evaporation refers to evaporating the material that is to be deposited by controlling the temperature and pressure of the host material environment. A film is formed when the material condenses. A continuous evaporation–condensation process is established that allows for a controlled growth rate of the film.

Sputtering involves bombardment of the host material with high energy ions to dislodge molecules, which will reattach themselves to the surface of the wafer (as well as to other surfaces in the sputtering apparatus). Often two different host materials are simultaneously bombarded at different rates to establish the characteristics of the sputtered material. This dual host bombardment is termed cosputtering. With some materials, sputtering offers advantages over evaporation in host material integrity on the deposition surface.

Chemical vapor deposition (CVD) is achieved in two ways: (1) by causing a reaction of two gases near the substrate, a reaction occurs that creates solid molecules, which subsequently adhere to the substrate surface; or (2) by pyrolytic decomposition (a decomposition caused by heating) of a single gas, which also frees the desired molecules for reattachment.

ETCHING. Etching refers to selectively removing unwanted material from the surface of the substrate. Photoresist and masks are used to selectively pattern (expose) the surface of the substrate. Following this patterning, the physical characteristics of the surface are changed by etching. A single IC will generally undergo several different etches during processing. The chemicals used for etching are chosen to selectively react with unprotected areas on the wafer while not affecting the protected areas. A summary of the effects of some commonly used etchants on typical semiconductor materials is shown in Table 2.1-1.

There are two types of etches used in production: wet and dry. The *wet etches*, often called chemical etches, use liquid etching agents, which are applied to the substrate surface. Although they have received widespread application in the past, they etch horizontally as well as vertically into the surface of the substrate. This horizontal etching causes undercutting of the patterned areas. Unless the width of the nonetched regions is orders of magnitude greater than the thickness of the material being etched, the nonuniformity of the horizontal etching causes significant changes in desired device characteristics.

TABLE 2.1–1
Characteristics of commonly used fabrication materials

I. Materials used in IC fabrication		
Purpose	**Materials**	**Comments**
Silicon crystal substrates	$SiCl_4$	Silicon source for growth of single crystal silicon
	$SiHCl_4$	Silicon source for growth of single crystal silicon
	SiO_2 (Sand)	Silicon source for growth of single crystal silicon
Silicon layers (both single crystalline and polysilicon)	$SiCl_4$ and H_2	The hydrogen gas strips the Cl atoms to form solid silicon.
	SiH_4	Heat causes the release (pyrolysis) of H_2 gas.
	SiH_2Cl_2	Heat causes the release (pyrolysis) of HCl gas.
Oxides	O_2	Used to grow SiO_2 by thermal oxidation
	H_2O (Steam)	Used to grow SiO_2 by thermal oxidation
	SiH_4 and O_2	Used for CVD deposition of SiO_2 and to grow protective "glass" (SiO_2)
Nitride layers	Si_4 and NH_3	The ammonia causes the release of hydrogen gas and leaves Si_3N_4.
	$SiCl_4$ and NH_3	The ammonia causes the release of HCl and leaves Si_3N_4.
Etches, wet	HF	Hydrofluoric acid etches SiO_2 but not Si, Si_3N_4, or photoresist.
	HF and HNO_3	Etches Si

TABLE 2.1–1
(Continued)

Purpose	Materials	Comments
	H_3PO_4	Hot phosphoric acid etches Si_3N_4 but not SiO_2. Removes some types of photoresist.
Etches, dry	CHF_3	Etches SiO_2
	C_3F_8	Etches SiO_2
	SF_6	Etches silicon
	CF_4	Etches Si_3N_4
	CCl_4	Etches aluminum
Patterning	Photoresist	Used as barrier to ion implants. Also used to pattern SiO_2 since photoresist is not affected by HF, a common SiO_2 etchant.
	SiO_2	Acts as a barrier to some p- and n-type impurities
	Si_3N_4	Used as protective layer over silicon or SiO_2 to prevent thermal growth of SiO_2. Also serves as a barrier to low-energy ion implants although thin layers can be and are penetrated with higher-energy implants. Also serves as a diffusion barrier to impurities such as Ga, Al, Zn, and Na.

II. Sources of impurities

	Impurities	Source
n-type	Arsenic	As_2O_3, AsH_3
	Antimony	Sb_2O_3, Sb_2O_4
	Phosphorus	P_2O_5, $POCl_3$ (liquid), PH_3 (gas—implant or diffusion)
p-type	Gallium	
	Aluminum	
	Boron	BN (solid), BBr_3 (liquid), B_2O_3 (gas), BCl_3, (gas), B_2H_6 (gas), BF_3 (for implants)

III. Impurity migration in silicon

Impurity	Silicon	SiO_2
Arsenic†(n)	moderate	very slow
Antimony (n)	moderate	very slow
Phosphorus (n)	fast	slow
Gallium (p)	moderate	fast
Aluminum (p)	fast	fast
Boron (p)	fast	slow

†Arsenic is often preferred to the other n-type impurities because it gives more abrupt junction gradients, which yield better frequency response and improved current gain in bipolar transistors. Due to environmental concerns, however, the use of arsenic in the semiconductor industry is limited.

Dry etching, also termed ion etching, is directional and thus much less susceptible to the undesirable horizontal undercutting. Dry etching techniques include sputter etching, ion-beam etching, and plasma etching. Since no liquid chemicals are involved, a significant reduction of costs associated with disposal of spent chemicals is realized when dry etches are used. The etch rate for dry etches is generally lower than the wet etch rate. Dry etching is recognized as a practical alternative to wet etching and is widely used.

The characteristics of an ideal wet etch and an ideal directional etch are shown in Fig. 2.1-2. The nondirectional etch is termed an isotropic etch. The edge profile appears approximately circular, with radius r and center at point A. If the etch is stopped precisely when the underlying layer is exposed the radius r will be T, the thickness of the layer, and the undercut of the protective layer, X, will be also T. If the etch is not stopped precisely when the underlying layer is exposed, the radius will be T_1, which is greater than T, and both the effective opening and the undercut will thus be larger than desired.

An ideal directional etch is termed an anisotropic etch. Note that an anisotropic etch has a very abrupt edge, which causes problems for applying subsequent layers uniformly and reliably across this edge.

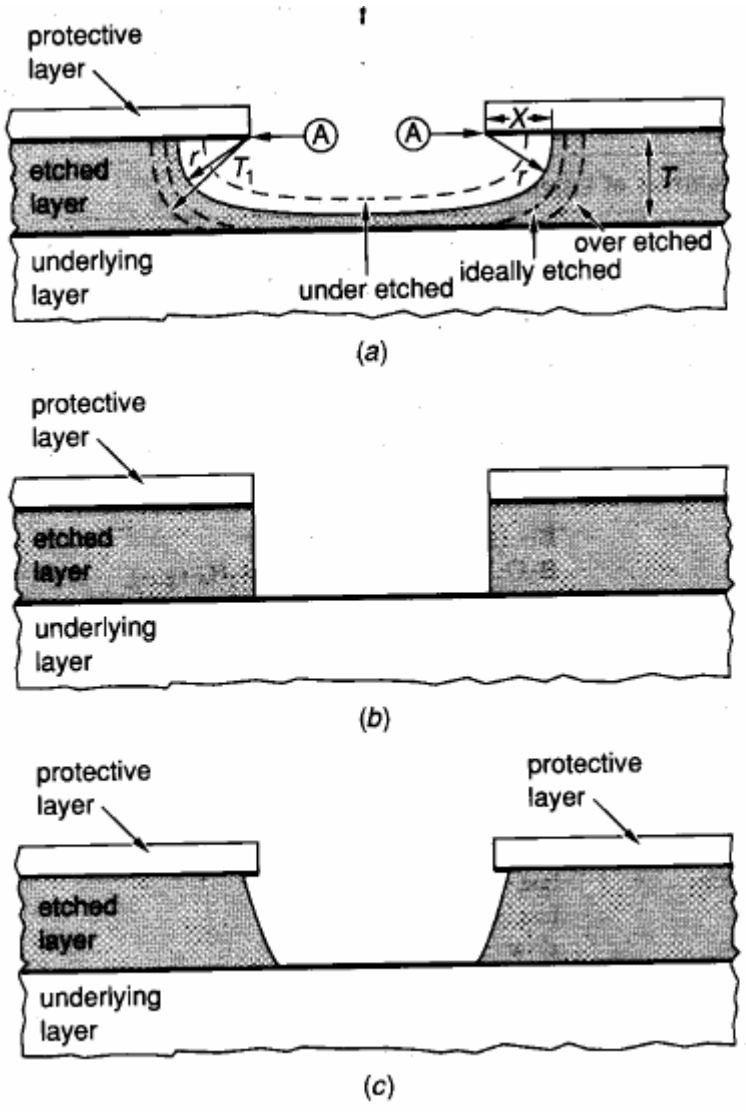


FIGURE 2.1-2
Characteristics of etches: (*a*) Isotropic etch, (*b*) Anisotropic etch, (*c*) Preferential etch.

DIFFUSION. Diffusion, in the sense of an IC processing step, refers to the controlled forced migration of impurities into the substrate or adjacent material. The resultant impurity profile, which plays a major role in the performance of the integrated circuit, is affected by temperature and time as well as the temperature–time relationship during processing. Subsequent diffusions generally cause some additional migration of earlier diffusions. Actually, the diffusion process continues indefinitely, but at normal operating temperatures of the integrated circuit it takes tens of years or longer for the additional movement to become significant.

The method by which the impurities are introduced varies. A solid deposition layer or a gaseous layer above the surface can be used as the source of impurities. Impurities can also be accelerated to selectively bombard the substrate so that they actually become lodged inside the substrate very near the surface. This technique, termed *ion implantation*, offers very accurate control of impurity concentrations but causes significant crystal damage near the surface.

The purpose of a diffusion following deposition is to cause a migration of carriers into the substrate from either solid or gaseous surface layers. A diffusion step following ion implantation is used to mend or anneal bombardment–induced fractures in the single crystalline structure at the surface of the substrate as well as to cause additional impurity migration.

As in the etching process, the direction of impurity diffusion is difficult to control with accuracy. Impurities typically diffuse both vertically and laterally from the surface at comparable rates in a manner similar to that observed for the isotropic etch of Fig. 2.1-2*a*.

CONDUCTORS AND RESISTORS. Aluminum or other metals are often used as conductors for interconnection of components on an integrated circuit. These metals are typically deposited, patterned, and etched to leave interconnects where desired. The thickness of the popular aluminum films is typically about 6000–8000 Å but may be as much as 20,000 Å for linear (analog) single-level metal processes. Metal films are particularly useful for interconnects that must carry large currents, but traces must be wide enough to avoid the *metal migration*, or *electromigration* problem. Electromigration is the movement of atoms with current flow and can be likened to wind erosion of dirt. If significant metal migration occurs, the conductors become open, resulting in failure of an integrated circuit. Metal migration is insignificant provided the peak current density in the conductor is below a certain threshold. For aluminum, this threshold is around 1 mA/μ^2. This threshold is material dependent and ranges from 0.05 mA/μ^2 to 2 mA/μ^2 for other similar materials.

Nonmetallic films are widely used for conductors and interconnects when current flow is small. These materials are typically worse conductors than metals

and thus cause a significant voltage drop when currents are large. These materials find limited applications as resistors.

Polysilicon is one of the most popular nonmetallic conductors. Polysilicon differs from single crystalline silicon, which is often used as a substrate material, only in that polysilicon is composed of a large number of nonaligned, randomly oriented, small silicon crystals. Although polysilicon is chemically identical to single crystal silicon, its electrical characteristics are much different. Polysilicon is a good conductor when heavily doped and a good resistor when lightly doped. Polysilicon is often used for gates of field effect transistors (MOSFETs) and as an electrode for capacitors. Polysilicon can be deposited over SiO_2. SiO_2 can also be readily grown on polysilicon and is often used to serve as a dielectric and isolate two polysilicon layers in processes where double polysilicon (*double poly*) layers are available. Polysilicon's characteristics are dependent on the size of the small crystals, often termed the grain size. It can be deposited on a variety of materials and the growth rate can be fairly fast. Polysilicon films are typically about 2000 Å thick and are often termed *poly*.

Silicides and/or refractory metals are often used on top of or in place of polysilicon for fabricating conductors. These materials are often much better conductors than polysilicon.

OXIDATION. Oxidation is the process whereby oxygen molecules from a gas above the substrate or surface material cause the growth of an oxide on the surface. Since the substrate or surface material is typically silicon, the oxidation process produces silicon dioxide. The speed at which the SiO_2 layer grows is a function of the doping concentration and the temperature of the substrate during oxidation. The SiO_2 layer serves as a very good insulator between the substrate or surface material and whatever is placed upon it. When the SiO_2 layer is grown on the substrate, a small amount of the Si in the substrate is consumed to provide for the Si molecules in the oxide. The growth of x microns of SiO_2 consumes approximately $0.47x$ microns of single crystal silicon.

As an alternative to oxidation, the SiO_2 layer can be applied by CVD. This technique is used extensively when the SiO_2 layer must cover something other than Si since no silicon molecules are available for oxidation. CVD can also be done at lower temperatures, which is advantageous if additional diffusion of previously deposited materials must be minimized. SiO_2 layers formed by oxidation are generally more uniform than those formed by a CVD process.

Other types of oxides are also used as insulating layers in fabricating ICs. Doped deposited oxides such as phosphosilicate glass (PSG) are often used as insulators on top of polysilicon. Some of these are doped to improve reflow characteristics during annealing. This doping helps reduce sharp boundaries (improve step coverage) introduced during etching of polysilicon.

Nitride (Si_3N_4) is also used as a dielectric between two levels of polysilicon in some processes. The dielectric constant of Si_3N_4 is about four times that of SiO_2. This offers potential for much higher capacitance densities for fixed dielectric thicknesses or much thinner dielectrics for a fixed capacitance density. A thin layer of SiO_2 is generally applied to the Si prior to the Si_3N_4 to minimize the mechanical stress associated with a direct Si interface to Si_3N_4. This stress is caused by a difference in the lattice characteristics of single-crystal silicon and the Si_3N_4 layer.

Although somewhat different chemically, polyimides are also used as insulating layers, most notably between two metal layers. Polyimides tend to smooth abrupt underlying irregularities, thus reducing the effects of sharp boundaries of underlying metal layers.[2]

EPITAXY. Epitaxial growth is generally a CVD. It warrants singling out, however, because of the extensive use of this process step in bipolar integrated circuitry and because epitaxial layers are ideally single crystalline extensions of the substrate. Epitaxial layers are grown slowly enough that the molecules added to the surface can align with the underlying crystalline structure of the substrate to form a crystalline epitaxial layer. A small amount of n- or p-type impurities is generally intentionally introduced into the epitaxial layer during the epitaxial growth to obtain a doped epitaxial layer.

2.1.2 Packaging and Testing

After processing, the integrated circuits are tested and packaged. The first step in the testing process generally involves a process verification to make certain that the process parameters are within the tolerances acceptable for the product. To facilitate this verification, *test plugs* containing special test structures specially designed for this purpose are included on the wafer at several locations in place of the regular circuits themselves. Alternatively, to avoid sacrificing the potential production die sites that are devoted to test plugs, there is a growing trend to integrate the test patterns into the *scribe lines*, which are existing grid lines void of circuitry where cuts will be made to separate the dies. A wafer prober is used to make mechanical contact with the test plugs so that electrical measurements can be made. Assuming the process parameters are within tolerance, the individual dies are automatically probed and electrically tested. Defective dies are marked with ink and later discarded. After probing, the wafer is scribed (typically with a wafer saw) both horizontally and vertically between adjacent dies, and the dies are separated. Following separation the individual dies are *die attached*, or *die bonded*, to a carrier or to the IC package itself. Wire bonds are subsequently made from the pins of the package to the appropriate locations on the die. The bonding wires are typically of either gold or aluminum. The diameter of this wire is in the range of 1 mil. After the wire bonds are complete, the packages are formed or closed and a final electrical test (and burn in for some parts) is completed.

Packaging technology saw minimal advancements through the late 1970s and early 1980s. It is well recognized that existing packaging techniques are a major bottleneck in the evolution of IC technology. Considerable effort on a worldwide basis is focused on the packaging problem. Practical alternatives to the conventional packaging approach, described above, will likely evolve in the next few years.

2.2 SEMICONDUCTOR PROCESSES

There are currently three basic processes used for the fabrication of monolithic integrated circuits containing active devices. These are the NMOS, CMOS, and bipolar processes. The first two are both termed MOS (Metal Oxide Semiconductor) processes even though, as will be discussed later, the standard acronym is no longer completely descriptive. A fourth approach essentially combines bipolar and MOS technologies into a single but more involved process. The mixed bipolar–MOS process is called Bi–MOS. A fifth method for constructing ICs is termed the hybrid process. These processes are depicted in Fig. 1.2-1.

Generic processes similar to those used in industry will be discussed in this section. The generic NMOS and CMOS processes discussed are very similar to those available through MOSIS† and the same terminology and conventions that have been established by MOSIS will be followed when practical. Several excellent references provide additional information about the NMOS and CMOS processes available through MOSIS and about MOS processing in general.[1, 2, 7–11] Additional information about these generic processes, such as design rules and process parameters, are discussed in Section 2.3. Details about a typical Bi–MOS process are not presented, but the basic approach should be apparent after studying the basic MOS and bipolar processes.

The NMOS (n-channel MOS) and CMOS (Complementary Metal Oxide Semiconductor) processes are quite similar in that both have the field effect transistor (FET or MOSFET) as the basic active device. In the NMOS process n-channel MOSFETs are available as the active devices whereas in the CMOS process both n-channel and p-channel devices are available. When compared to the NMOS process, the CMOS process offers advantages in design simplicity at the expense of more processing steps. It is often the case that CMOS also offers improvements in power dissipation and performance and in some cases even size over NMOS. These tradeoffs must be considered when selecting the most economical process for a given application. Another MOS process, PMOS, is available but will not be singled out because it is essentially a dual of the NMOS process. In the PMOS process the basic active device is the p-channel MOSFET. Although the PMOS process was commonly used for some of the earlier MOS circuits, the NMOS process offers some advantages due to characteristics of semiconductor materials available (specifically, electron mobility is higher than hole mobility) and is more popular today.

The bipolar process is so named because the basic active device is the Bipolar Junction Transistor (BJT). Higher speeds are currently available with the bipolar process than for the NMOS and CMOS processes although significant improvements in the speed of the latter processes have been and continue to be made. Bipolar integrated circuits are noted for their considerable internal power dissipation compared to that of the NMOS and CMOS processes. For logic circuits the NMOS and CMOS circuits have a significantly higher component density than their bipolar counterparts.

The hybrid process combines thin and/or thick film passive components that are on one or more separate substrates with active devices from a separate substrate onto a common carrier. This makes hybrid ICs quite expensive. For applications that require precise and temperature-stable passive components, the hybrid process often offers a practical solution.

MATERIAL CHARACTERIZATION. Some terminology that is common to most semiconductor processes is best introduced at this point. Throughout this text the notation n^+ will denote a heavily doped n-type semiconductor region, and n^- will denote a lightly doped region. The designation n^+ or n^- will be assumed relative to the context in which this designation is made. No superscript will be included if the region is doped somewhere between n^+ and n^- or if it is not necessary to make the distinction in the given context. The same convention will be followed for p, p^+, and p^- designations.

The *resistivity* of a homogeneous material is a volumetric measure of resistive characteristics of the material. The resistivity is typically specified in terms of ohms-cm (Ω-cm). If a right rectangular solid of material of length L and cross sectional area A (see Fig. 2.2-1*a*) has a measured resistance of R between the two ends, then the resistivity of the material is given by

$$\rho = \frac{AR}{L} \tag{2.2-1}$$

where it is assumed that the contacts on the two ends cover the entire surface and are perfectly conducting.

The *sheet resistance* is a measure of the characteristics of a large, uniform sheet or film of material that is arbitrarily thin. The sheet resistance is specified in terms of *ohms per square* of surface area. If a rectangular sheet of material of length L and width W (see Fig. 2.2-1*b*) has a measured resistance R between the two opposite ends, then the sheet resistance of the material is given by

$$R_{\square} = R\frac{W}{L} \tag{2.2-2}$$

where it is assumed that the contacts on the two edges cover the entire edge and are perfectly conducting. It should be emphasized that both the resistivity and sheet resistance are characteristics of materials independent of particular values for A, L, W, and R in the previous equations.

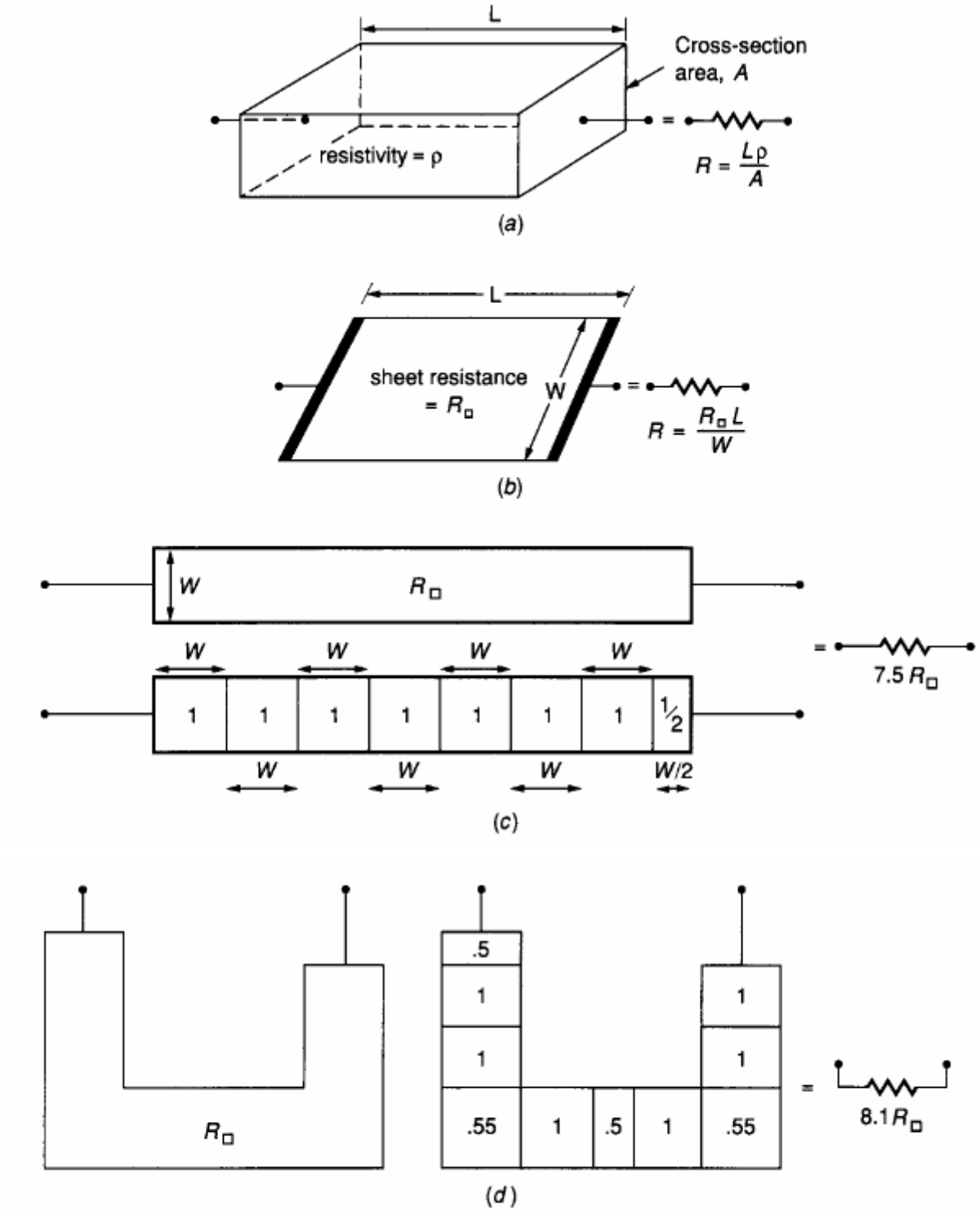


FIGURE 2.2-1
Resistive characteristics of bulk and sheet materials: (*a*) Resistivity, (*b*) Sheet resistance, (*c*)&(*d*) Graphical calculations from sheet resistance.

The sheet resistance of a thin layer of thickness z constructed from a material that has a resistivity ρ is given by

$$R_{\square} = \frac{\rho}{z} \tag{2.2-3}$$

The resistance of thin rectangular regions of length L and width W on an integrated circuit can be readily obtained from the sheet resistance by counting the number of square blocks of length W that can be placed in the rectangular region. If N blocks can be placed adjacently in the region, then the resistance in terms of the sheet resistance, $R_{\square}$, is given by

$$R = NR_{\square} \tag{2.2-4}$$

An example illustrating this technique is depicted in Fig. 2.2-1*c*.

Occasionally it is necessary to determine the resistance of nonrectangular regions such as that shown in Fig. 2.2-1*d* or the serpentined pattern of Problem 2.11. The problem of determining resistance of irregular regions is difficult, but for rectangular regions containing the right angles shown, the rule of thumb of adding 0.55 squares for each corner is often used.

It is often necessary to specify the temperature characteristics of resistors and capacitors. The Temperature Coefficient of Resistance (TCR) and Temperature Coefficient of Capacitance (TCC) are typically used for this purpose. These temperature coefficients, which are generally expressed in terms of ppm/°C, are defined by

$$\text{TC} = \left(\frac{1}{x}\right)\left(\frac{dx}{dT}\right) 10^6 \text{ppm/°C} \tag{2.2-5}$$

where x is the temperature-dependent value of either the resistor or the capacitor.

If the temperature coefficient is independent of temperature, then the value of the component at a temperature T_2 can be obtained from its value at temperature T_1 by the expression

$$x(T_2) = x(T_1)e^{[\text{TC}(T_2 - T_1)/10^6]} \tag{2.2-6}$$

which is often closely approximated by

$$x(T_2) \simeq x(T_1)[1 + (T_2 - T_1)(\text{TC}/10^6)] \tag{2.2-7}$$

If the TC is a function of temperature, then the previous expressions are good only in local neighborhoods of T_1. The value of TC, which can be either positive or negative, is determined by the material properties and is often quite small. The absolute values of TC are often less than 1000 ppm/°C. Unfortunately, since integrated circuits are often expected to operate over a relatively wide temperature range (0–70°C commercial or −55 to 125°C military) the effects of the TC can be significant.

Some resistors and capacitors, in addition to being temperature dependent, are also somewhat voltage dependent. This voltage dependence introduces nonlinearities in circuits using these devices along with the corresponding harmonic distortion (THD) in many applications. The voltage dependence of resistors and capacitors is characterized by the voltage coefficient, defined by

$$\mathrm{VC} = \left(\frac{1}{x}\right)\left(\frac{dx}{dV}\right)10^6 \mathrm{ppm/V} \tag{2.2-8}$$

where x is the voltage-dependent value of a resistor or capacitor. The voltage coefficient is analogous to the temperature coefficient, as can be seen by comparing Eqs. 2.2-5 and 2.2-8.

2.2.1 MOS Processes

A brief qualitative discussion of the principle of operation of the MOS transistor at dc and low frequency is now presented to provide insight into the MOS process itself. A detailed quantitative presentation about modeling these devices appears in Chapter 3.

OPERATION OF THE MOSFET. Consider the n-channel enhancement MOSFET shown in Fig. 2.2-2. In the cross-sectional views it can be seen that the gate (Metal or conductor) is over the insulator (Oxide), which is in turn over the substrate (Semiconductor). The source of the acronym MOSFET should be apparent. If the substrate is tied to the source as shown in Fig. 2.2-2*b*, then with a zero gate–source voltage the n-type drain and source regions are isolated from each other by the p-type substrate, preventing any current flow from drain to source. A depletion region also forms between the n^+ drain and source regions and the lightly doped substrate, as depicted in Fig. 2.2-2*b-f*. The corresponding pn junction is reverse biased under normal operation and has minimal effects on current flow at dc and low frequencies.

If a positive gate voltage is applied, electrons will start to deplete the substrate near the surface under the gate. This tends to deplete the p-type substrate in this region and form what is called a depletion region under the gate. A simplified pictorial presentation of this situation is shown in Fig. 2.2-2*c*.

If the gate voltage is increased sufficiently, a number of electrons will be attracted to the substrate surface under the gate sufficient to make this region n-type. This n-type region, which is created electrically (by the electric field established by the gate bias) in the p-type substrate, is called an *inversion layer*. The gate–source voltage necessary to create the inversion layer is called the *threshold voltage*, V_T. The inversion layer, shown in Fig. 2.2-2*d*, is often termed the *channel* of the MOSFET.

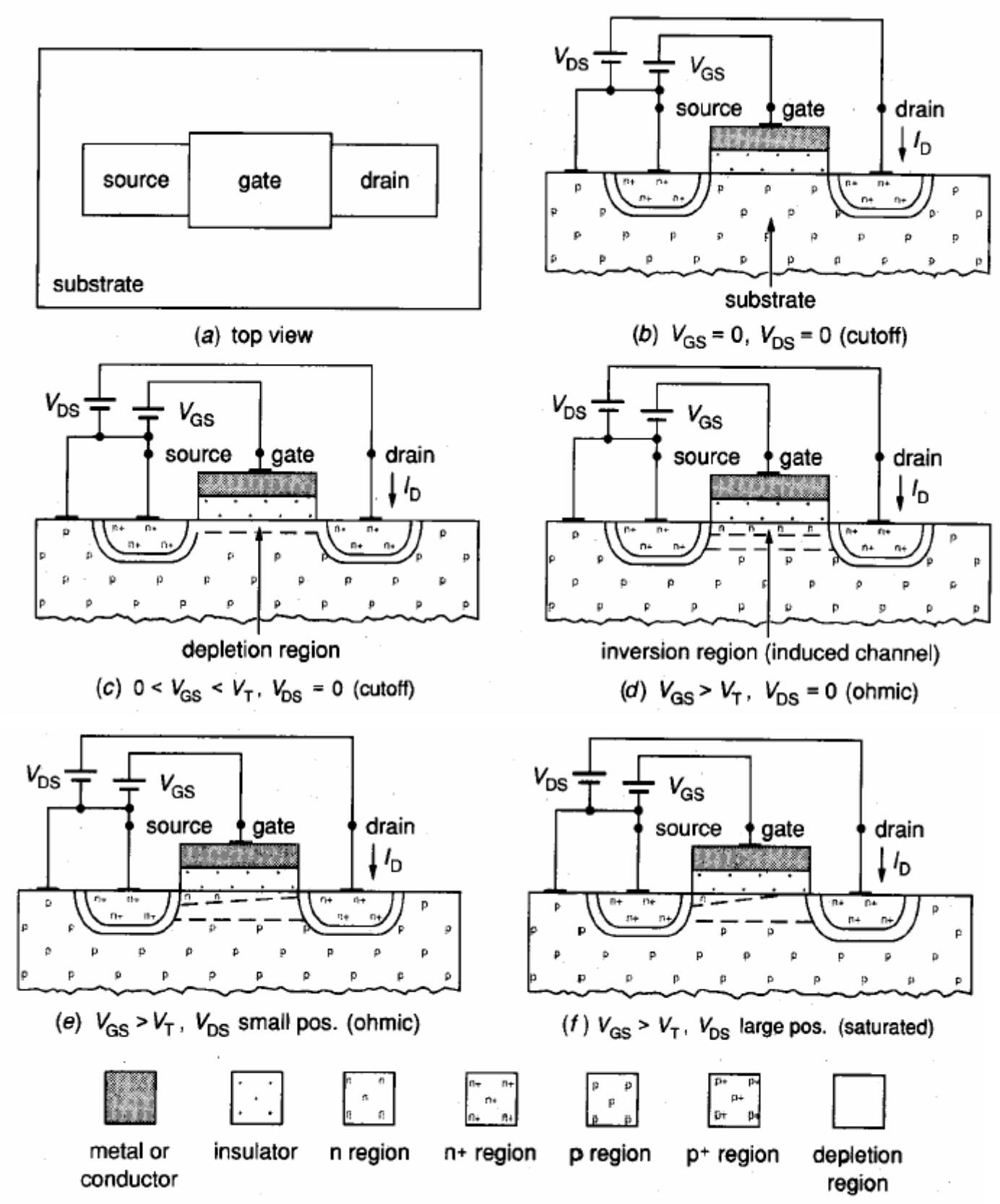


FIGURE 2.2-2
Operation of n-channel MOSFET (horizontal and vertical scale factors are different).

Once the inversion layer is created, current will flow from the drain to source or source to drain if a small voltage is applied between these regions. The insulator under the gate prevents any gate current from flowing, thus forcing the current entering the drain to be equal to that leaving the source. Increasing the gate–source voltage beyond the threshold voltage brings additional electrons under the gate, causing an increase in the thickness of the inversion layer and increased current flow from the drain to source (or source to drain) under a fixed drain-to-source bias. If a large drain–source (or source–drain) voltage is applied, this voltage itself will tend to deplete the inversion layer due to the potential drop across the region caused by the current flow. A cross section of the device is shown under a small drain-to-source bias in Fig. 2.2-2*e* and under a large drain-to-source bias in Fig. 2.2-2*f*. For a fixed gate–source voltage, there is a value of V_{DS} that effectively pinches off the channel near the drain. This does not cause a decrease in drain current (I_D), for if it did, the inversion layer would immediately reappear since the channel current itself causes the pinching of the inversion layer. If the value of V_{DS} is increased further, the drain current will remain nearly constant.

When the gate–source voltage is greater than the threshold voltage, the MOS transistor is said to be operating in the *ohmic region* prior to the pinching of the channel and in the *saturation region* when the channel is pinched off. If the gate–source voltage is less than the threshold voltage, almost no drain or source current will flow even when a bias is applied to the drain and source contacts. In this case the device is said to be *cutoff*. The relationship between I_D, V_{DS}, and V_{GS} for a typical MOSFET, termed the output characteristics, is shown in Fig. 2.2-3, along with the ohmic and saturation regions of operation. The cutoff region is the $I_D = 0$ line in this figure. The gate current remains at $I_G = 0$ in all three regions of operation.

The value of the threshold voltage is determined by the concentration of the p-type impurities in the substrate. If some n-type impurities are added to the region under the gate near the surface of the substrate, the threshold voltage will decrease. If sufficient impurities are added, the region itself will become n-type and the threshold voltage will become negative. An n-channel device with a positive threshold voltage is termed an *enhancement MOSFET* and those with a negative threshold voltage are termed *depletion MOSFETs*. MOS devices formed in a p-substrate (or tub) and thus having n-type drain and source diffusions and an n-type channel are termed *n-channel transistors*. Those formed in an n-type substrate (or tub) with p-type drain and source diffusions and a p-type channel are termed *p-channel transistors*. In contrast to the convention introduced above for n-channel transistors, p-channel transistors with a negative threshold voltage are termed enhancement devices and those with a positive threshold voltage are termed depletion devices.

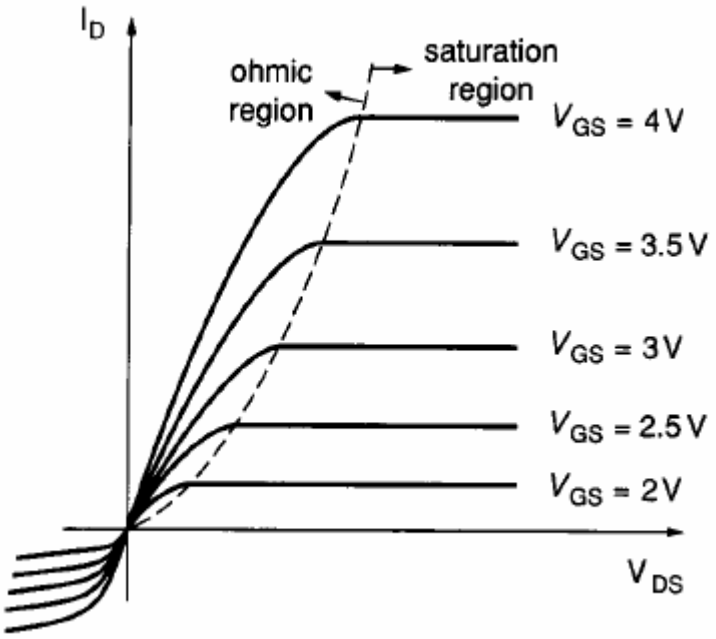


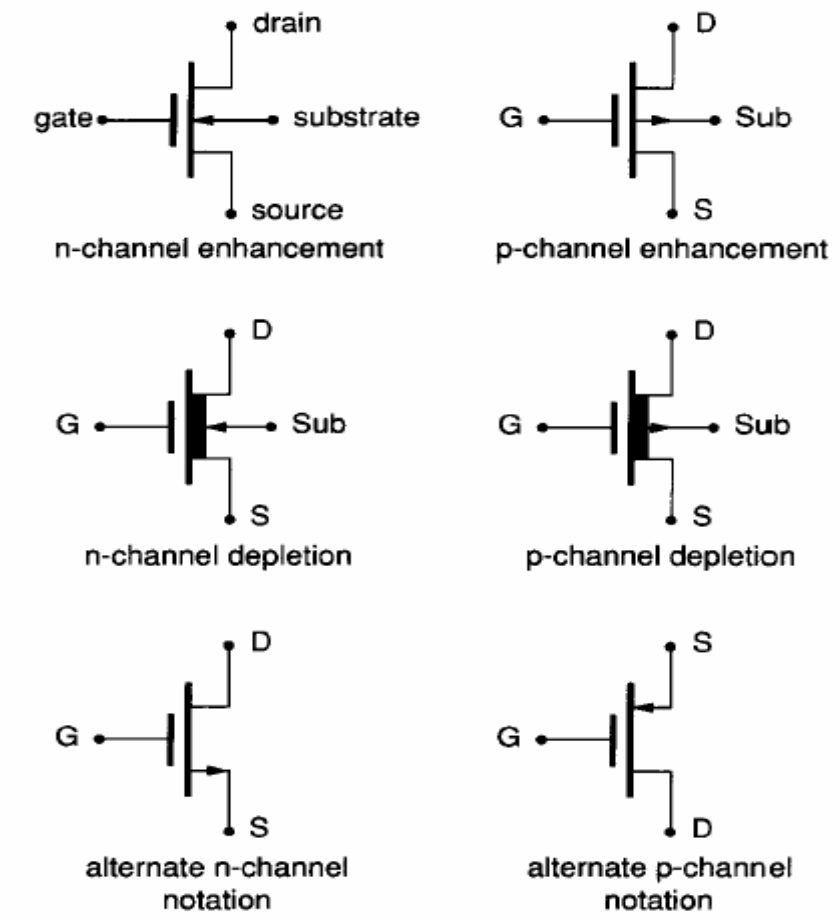
FIGURE 2.2-3
Typical output characteristics for an n-channel MOSFET.

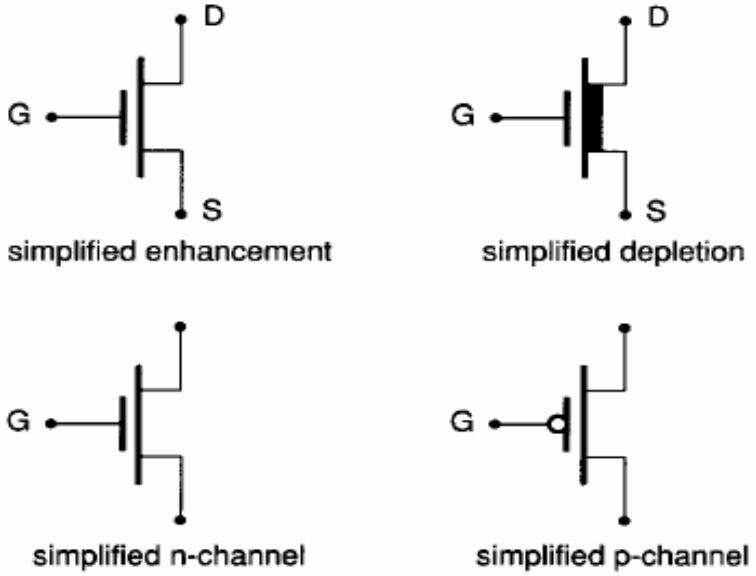
Commonly used symbols for enhancement and depletion n- and p-channel devices are shown in Fig. 2.2-4. Since the polarity of the substrate is often known for either the NMOS or PMOS processes, the simplified notation shown in Fig. 2.2-4, which does not maintain this information in the device symbol, has been widely adopted for both NMOS and PMOS devices.

2.2.1*a* NMOS Process

Although both NMOS and PMOS processes are currently available, the NMOS process has been used more extensively in recent years. The NMOS process is preferred because the characteristics of the n-channel MOSFET are preferable to those of the p-channel MOSFET. This is attributable to a higher mobility for electrons than for holes. The discussion that follows will be based upon the NMOS process. Modifications of this presentation to describe the PMOS process are straightforward and are thus left to the reader.

A discussion of a generic double-polysilicon, self-aligned silicon gate NMOS enhancement/depletion process follows. This can be considered as a typical standard process although processes that offer more as well as less flexibility are also standard. This process is similar to a widely used MOSIS NMOS process augmented by a second polysilicon layer. The same basic approach used here is used for other NMOS processes. The MOS transistors in this process are similar to those depicted in Fig. 2.2-2, with the exception that polysilicon (a good conductor) is used instead of metal for the gate. Field effect transistors with polysilicon gates are also called MOS transistors or MOSFETs even though the acronym MOS is no longer completely descriptive.





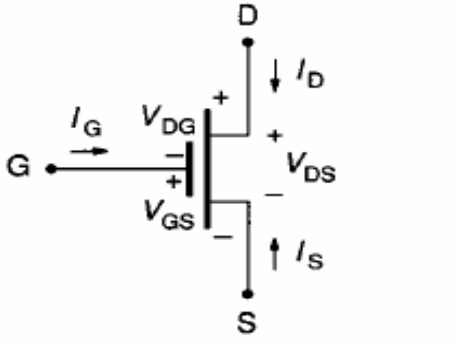


FIGURE 2.2-4
Symbols for MOS transistors.

The devices that are available in this process are

1. n–channel enhancement MOSFETs.
2. n–channel depletion MOSFETs.
3. Capacitors.
4. Resistors.

The method of physically constructing each of these components in this process and interconnecting them to form the simple circuit shown in Fig. 2.2-5 will now be addressed. Both top views and cross-sectional views are presented in Fig. 2A.1 of Appendix 2A. Cross sections are along sections AA′ and BB′ of Fig. 2A.1*a*. A summary of the major process steps appears in Table 2A.1 of Appendix 2A. Additional information about this process relating to layout sizing rules, physical feature sizes, and electrical characterization parameters of the generic NMOS process can be found in Tables 2A.2–2A.5 of Appendix 2A. (Table 2A.3 appears in Plate 5 in the color plate insert.) The circuit designer must present the top view of each mask level for fabrication but must have a firm understanding of the cross-sectional view for effective design.

The method of physically constructing each of these components in this process and interconnecting them to form the simple circuit shown in Fig. 2.2-5 will now be addressed. Both top views and cross-sectional views are presented in Fig. 2A.1 of Appendix 2A. Cross sections are along sections AA′ and BB′ of Fig. 2A.1*a*. A summary of the major process steps appears in Table 2A.1 of Appendix 2A. Additional information about this process relating to layout sizing rules, physical feature sizes, and electrical characterization parameters of the generic NMOS process can be found in Tables 2A.2–2A.5 of Appendix 2A. (Table 2A.3 appears in Plate 5 in the color plate insert.) The circuit designer must present the top view of each mask level for fabrication but must have a firm understanding of the cross-sectional view for effective design.

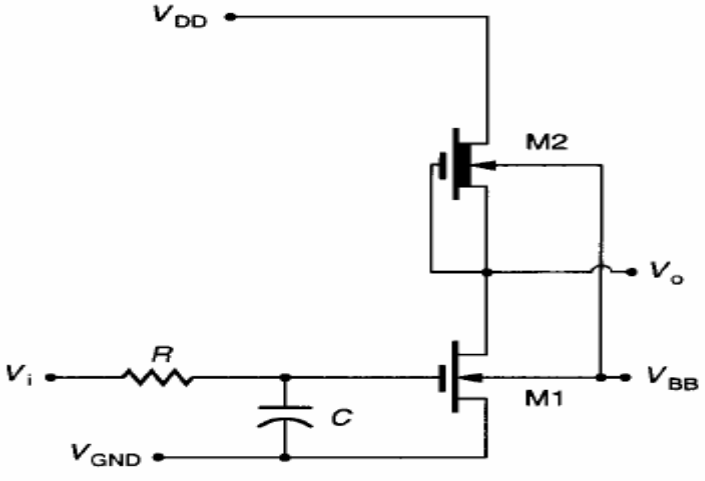


FIGURE 2.2-5
A simple NMOS circuit.

For the NMOS process, the starting point is a polished p-type silicon disc. The thickness of the disc is typically around 500 μ. A layer of SiO_2 in the neighborhood of 1000 Å thick is first added to the entire wafer using the oxidation process. On top of this a layer of Si_3N_4 (about 1500 Å thick) is applied by the CVD process. Following the application of a layer of photoresist, Mask #1 is used to pattern the surface. Mask #1, which is often called the *moat*, or n^+ *diffusion* mask, defines in photoresist the drain, source, and channel regions of all transistors as well as any other regions where n^+ implants are desired. After exposure, development removes the photoresist layer in areas that are not to be moat (i.e., the *complement of the moat*, or the *antimoat*). A top (Mask #1 pattern) and cross-sectional view at this stage of what will be the two transistors appear in Fig. 2A.1*a*. The Si_3N_4 is then etched from the areas not protected by the photoresist. A high-energy implant of p-type impurities (typically boron) is then applied to the entire wafer. The remaining photoresist protects the moat regions from this implant. This heavy implant is used to raise the threshold voltage in the antimoat region (often called the *field*) and to provide electrical isolation between adjacent devices. After this field implant and a drive in diffusion, the remaining photoresist is stripped. A thick layer of SiO_2 (about 10,000 Å) is then thermally grown by the oxidation process over the wafer. This layer is formed in the field, but no oxidation can take place in the region protected by the Si_3N_4 because Si_3N_4 does not oxidize. The thick field oxide layer is termed a local oxidation layer and is often called *LOCOS*. The oxidation consumes some of the substrate silicon. The second cross section in Fig. 2A.1*a* shows the state of the wafer following growth of the field oxide. This corresponds to Step 10 in the process scenario of Table 2A.1. Following removal of the Si_3N_4, the thin layer of SiO_2 under the Si_3N_4 is stripped and another SiO_2 layer is grown.

With the moat now protected only by the very thin SiO_2 layer, a light n-type implant over the entire wafer can be applied (optional) to set the threshold voltage of the enhancement devices. This implant is light enough so that all p-type regions remain p-type. If used, the implant is applied to the entire wafer to avoid the need for an additional mask.

A heavier selective implant is required in regions that are to serve as the channels of depletion transistors. To achieve this, a second layer of photoresist is applied to the entire wafer, and the second mask, Mask #2 (termed the *implant* mask), is used to pattern the photoresist so that only the channel regions of depletion transistors are unprotected. Another n-type implant is used to make the exposed regions n-type with the remaining photoresist serving as an implant mask. After stripping the photoresist, the wafer is as shown in Fig. 2A.1*b*. This corresponds to the status of the wafer at Step 19 in Table 2A.1.

Direct contacts between the lower polysilicon layer and moat are termed buried contacts. In the process scenario of Table 2A.1 this is listed as an optional step and is not used in the layout shown in Fig. 2A.1 although it could be used, if available, to reduce the area required for contacting the gate of M2 to its source. To make a buried contact, the thin gate oxide must be patterned and etched to remove the insulating SiO_2 layer and create paths (vias) through which the following polysilicon layer can contact the moat. Mask #A is used to pattern the buried contact vias. Although the buried layer contact can reduce area, the additional processing costs needed to provide this feature are often not justified; hence the buried contact feature is often not available in NMOS processes.

After stripping of any thin oxides that may be present at this stage in the moat region, a uniform thin layer of SiO_2, often termed *gate oxide* (200 to 1000 Å thick), is grown on the surface of the wafer. Stripping and regrowing provide better control of the critical gate oxide thickness. A layer of polysilicon (termed POLY I), which is about 2000 Å thick, is then deposited on the surface of the entire wafer. This is covered with photoresist, patterned with Mask #3, and etched to remove unwanted POLY I. The POLY I layer is used as gates for both enhancement and depletion transistors, as a plate for capacitors, as a conductor, and for resistors. The formation of a capacitor, a resistor, and enhancement and depletion transistors can be seen in Fig. 2A.1*c*. This corresponds to Step 27 in the process scenario of Table 2A.1. Note that the POLY I layer is over gate oxide in cross section AA′ and above field oxide in cross section BB′.

The remaining uncovered thin layers of SiO_2 are stripped and another thin layer (500–1000 Å) of SiO_2 is again grown over the entire surface. This serves as the dielectric for POLY I–POLY II capacitors, as an insulator for POLY I–POLY II crossovers, and as the gate oxide for transistors that use the POLY II layer as the gate. The thickness of this oxide layer is often ideally the same as that of the first gate oxide layer. By stripping the unexposed oxide and regrowing, a buildup in the depth of the oxide layers is prevented and more uniformity is attained. A second layer of polysilicon (termed POLY II) is then deposited, followed by photoresist and patterning with Mask #B. In the circuit shown in Fig. 2.2-5, the POLY II layer is used only for the upper plate of the capacitor, as shown in Fig. 2A.1*d* (Step B.9 of Table 2A.1). Note that it is slightly smaller than the underlying POLY I layer. This difference is standard practice when trying to accurately match capacitors to make one plate a little smaller than the other so that the smaller plate will effectively define the capacitor area independent of slight misalignments of the two plates. Additional practical considerations that further improve matching will be discussed later.

After stripping the thin SiO_2 layers, an n^+ diffusion is applied to the entire wafer. The field oxide and polysilicon layers serve as masks to the diffusion and prevent impurities from reaching the substrate in the protected areas. The n^+ diffusion creates the n-type drain and source regions of all transistors and makes any other unprotected moat areas n-type. The portion of the light depletion diffusion that is not protected by the polysilicon gates also becomes more heavily doped. Note that although no mask is used at this step, the n^+ diffusion mask, which was used as the first masking step, has essentially determined the n^+ diffusion regions fabricated at this step. The n^+ diffusion also penetrates into any exposed polysilicon layers, increasing the conductivity in these regions. The n^+ diffusion depth is about 5000 Å. It will be seen in the section discussing process parameters that the sheet resistance for POLY I and POLY II layers is about the same but that the sheet resistance of POLY I under POLY II is higher. This is due to the absence of the additional n^+ diffusion in the lower layer. Since the polysilicon gates serve as masks for the n^+ drain and source diffusions, the process is said to be *self-aligned*. In a self-aligned process, small misalignments of the gate (POLY I or POLY II) masks will not affect the gate geometry or dimensions, nor will they make the transistors nonfunctional.

Next, another insulating layer is deposited over the wafer surface. Doped deposited oxide, such as PSG, is often used for this purpose. This rather thick layer, ~ 6000 Å, serves as an insulator between the uppermost polysilicon layer and the subsequent metal layer. The field oxide depth is further increased with this deposited oxide layer. The entire wafer is again covered with photoresist, and Mask #4 (actually the fifth or sixth mask if POLY II and/or buried contact options are available) is used to pattern *contact openings* for the purpose of obtaining electrical contact from the top with the desired components. After the photoresist is developed, an etch that attacks the insulating layer but does not affect polysilicon or silicon makes the required openings. This etch is stopped in the vertical direction only by polysilicon or the single-crystal silicon of the substrate. The wafer takes the form shown in Fig. 2A.1*e* (Step 34 of Table 2A.1).

Metal (typically aluminum) is then deposited over the entire wafer, followed by another layer of photoresist. This metal layer is typically about 7000 Å thick. This photoresist is patterned by Mask #5, followed by an etch to remove unwanted metal. The metalization is used to interconnect components and provide external access to the integrated circuit. A metalization that interconnects the four basic components to form the circuit shown in Fig. 2.2-5 is shown in Fig. 2A.1*f* (Step 40 of Table 2A.1).

Large, square metal areas, called *bonding pads*, are needed to allow for contact with the IC package. Small bonding wires will later be connected from these pads to the pins on the IC package. These pads are also patterned with the metalization mask but are not shown in Fig. 2A.1*f* because of the large amount of area required for bonding pads relative to that needed for the components in Fig. 2.2-5.

A bonding pad is shown in Fig. 2A.1*g*. Four of these would be needed to interface the circuit of Fig. 2.2-5 with the IC package. The V_{BB} contact comes from the bottom side of the substrate. The bonding pad size has remained relatively constant for a long period of time even though considerable reductions in feature size of geometries on the die itself have been experienced. This is because the methods of physically mounting the die in packages and interconnecting the bonding pads to the pins in the package have not changed much. With bonding wires typically about 1 mil in diameter, it is difficult to reduce bonding pad size significantly.

The entire surface is finally covered with a passivation layer (often called glass or p-glass) to provide long-term stability of the IC by minimizing atmospheric contamination. A layer about 10,000 Å thick is often used for passivation. Since this layer is also an electrical insulator, it is necessary to again pattern it and make openings above the metal pads to allow for attaching the bonding wires. The final mask, Mask #6, is used for this purpose and is shown in Fig. 2A.1*g*.

For the simple circuit of Fig. 2.2-5, the area required for the bonding pads dominates that needed for the circuit itself. For simple circuits this is generally the case but as the complexity of the circuit increases, the percentage of the total area required for bonding pads becomes quite small.

Giving the information for each mask separately, as was done in Fig. 2A.1, makes it difficult to perceive the entire circuit and determine layer to layer alignment. Sophisticated software packages termed layout editors are widely used, in which layers are color-coded and displayed simultaneously on high-resolution monitors. A single layout that simultaneously shows all mask information is shown in color in Plate 2. A color convention has been established for distinguishing separate layers. The color convention adopted in Plate 2 corresponds to that used in the MOSIS process and is discussed in more detail later in this chapter.

An interesting observation can be made from the layout of the MOS transistors of Fig. 2A.1. The MOS devices are totally geometrically symmetric with respect to drain and source and so must also be electrically symmetric. The designation of *drain* and *source* is thus arbitrary. In many applications a convention has evolved for convenience and consistency in device modeling in regard to drain and source designation. This convention will be discussed in Chapter 3. When appropriate, we will follow the established convention throughout this text.

In the process described, several alternative methods for constructing resistors and capacitors are available. For example, a region of moat with two contacts can be used as a resistor, and a capacitor can be made between POLY I and metal.

Processing step modifications such as omission of one polysilicon layer, omission of the depletion mask, substitution of metal gates for the polysilicon gates, and addition of another mask and implant to create enhancement transistors with two different threshold voltages are possible and common. Process procedure modifications such as using diffusions instead of implants; changing types of impurities; varying the thickness of oxide, polysilicon, or metal layers; including or excluding oxide stripping and regrowing steps; and changing types of photoresist are widespread and play major roles in yield, fabrication costs, and performance. The IC design engineer must be familiar with the process steps that will be used in fabrication when embarking on a new design.

For the process just described, it is the responsibility of the circuit designer to provide all information necessary to construct the seven masks shown in Fig. 2A.1. The size, shape, and spacing of the components are judiciously determined. The size and shape affect the performance of the circuit and are at the control of the circuit designer for optimizing performance, within constraints of minimum allowable size as determined by the capabilities of the process and maximum size as determined by economics. The spacing is also constrained by the capabilities of the process itself. The spacing and sizing specifications are obtainable from the design rules of the process, which are discussed later in this chapter. A process engineer will typically be responsible for providing design rules for a particular process.

2.2.1*b* CMOS Process

A discussion of a typical generic single-polysilicon silicon gate, p-well, n-substrate CMOS process follows. As in the NMOS case, variants in this process—such as a second metal layer, a second polysilicon layer, additional implants, oppositely doped substrate, or metal gates—are also well established. The devices available in the CMOS process under consideration are

1. n-channel MOSFETs.
2. p-channel MOSFETs.
3. Capacitors.
4. Resistors.
5. Diodes.
6. npn bipolar transistors.
7. pnp bipolar transistors.

The diodes and bipolar transistors are often considered parasitic components and are generally not extensively used as components in the circuit design itself. The process is tailored to maintain optimal characteristics in the n- and p-channel MOSFETs at the expense of poor characteristics for the bipolar transistors.

A method of physically constructing each of the first four components in the list will be considered. The approach followed here is similar to that followed for the NMOS process except that all mask details are included on the single layout of Fig. 2B.1 of Appendix 2B. A color version of this figure appears in Plate 3. Fewer details about oxide growth, photoresist application and patterning, and so on are provided since these steps are very similar to the corresponding steps for the NMOS process. Cross-sectional views along AA′ and BB′ in Fig. 2B.1*a* after each major step are shown in Fig. 2B.1. Interconnections follow the approach used in Section 2.2-1*a* for the NMOS process and are not discussed here. A summary of the major process steps appears in Table 2B.1. Additional information about this process relating to layout sizing rules, physical feature sizes, and electrical characterization parameters of the generic CMOS process can be found in Tables 2B.2–2B.5 of Appendix 2B. The process described here is very similar to the 3 μ CMOS/bulk process available through MOSIS.[7]

The starting point of this CMOS process is a polished n-type silicon disc. A layer of SiO_2 is first grown on the entire disc, followed by the application of a layer of photoresist. This photoresist is patterned with Mask #1 to provide openings for a p-tub (alternatively, p–well), which will serve as the substrate for the n-channel MOS devices. Either a deposition or implant is used to introduce the p-type impurities that form the tub. This diffusion is quite deep (about 30,000 Å). The remaining photoresist and SiO_2 are then stripped, a thin layer of SiO_2 regrown, and the entire surface covered with a layer of Si_3N_4.

Mask #2, termed the *moat mask* or the *active mask* by MOSIS, is used to pattern the Si_3N_4 layer. The Si_3N_4 layer is then etched away except above the regions that are to be the n^+ and p^+ diffusions or channel regions for the n-channel and p-channel MOSFETs. These diffusions will be added by subsequent processing steps to form drain and source regions for MOSFETs as well as to form guard rings. These protected regions are again termed moat. After the Si_3N_4 layer is opened, the remaining photoresist is stripped. Figure 2B.1*b* depicts the wafer after Step 15 of the process scenario of Table 2B.1.

An optional field threshold adjust step may be introduced at this point. This field threshold adjust would be used to raise the threshold voltage in the n-type

substrate in regions that will not contain devices. This will provide increased isolation between the p-channel transistors. Although an additional mask is required for this field adjust (Mask #A1 of Table 2B.1), the mask would be the complement of the union of the p-well mask and the active mask, Masks #1 and #2. As such, this mask information would be generated automatically and need not be separately provided by the designer.

A thick layer of field oxide (typically 10,000 Å) is grown in the regions not protected by the remaining Si_3N_4 that was patterned with Mask #2. The Si_3N_4, along with the remaining SiO_2 that was under this layer, are then stripped. The wafer at this stage is as depicted in Fig. 2B.1*c*. This corresponds to Step 18 in the process scenario of Table 2B.1. Note that along cross section AA′ several isolated areas are not protected by the field oxide. These areas will be used for fabricating transistors and guard rings. No breaks in the field oxide appear in the BB′ cross section. This corresponds to the region where the resistor and capacitor will appear; these devices are fabricated on top of the field oxide.

Next, a thin, uniform layer of SiO_2 (200 to 1000 Å), called in this case gate oxide, is regrown. A layer of polysilicon (typically 2000 Å) is then deposited, covered with photoresist, and patterned with Mask #3. This polysilicon layer, termed POLY or POLY I, is used for the gates of all transistors, as a plate on capacitors, for resistors, and for interconnects. Following etching and stripping, the wafer takes the form shown in Fig. 2B.1*d*. This corresponds to Step 25 in the process scenario of Table 2B.1.

An optional second polysilicon layer, termed POLY II, could be included here, as provided in the process scenario. The second polysilicon layer is not depicted in Fig. 2B.1. This second polysilicon layer would be separated from the first by a thin (500 to 1000 Å) insulating layer of SiO_2. The main purpose of the second polysilicon layer would be for the formation of capacitors with POLY I and POLY II as electrodes, although this layer would also find some use in interconnects and crossovers if available. The capacitance density and electrical characteristics of the poly–poly capacitors are more attractive than those obtainable with other capacitors available in this process. An additional mask, termed the POLY II or *electrode mask*, is needed to pattern this polysilicon layer. The etch of both the POLY I and POLY II layers produces an abrupt, sharp edge, making reliable coverage of this edge with thin material difficult. Since the oxide between POLY I and POLY II is thin, crossing of a POLY I boundary with POLY II may result in either a break in the POLY II or a shorting of POLY I and POLY II. To circumvent these problems, the crossing of a POLY I boundary with POLY II is often not permitted.

At this stage the drain and source diffusions for both the n-channel and p-channel transistors are added. Although two different types of diffusions and hence two separate masks are required, the designer need specify only one of the two masks. In this process, only those areas not protected by field oxide are capable of accepting any diffusion impurities. This is termed the moat, or active, region. It is further provided in this process that any moat area that is not exposed to n-type impurities will be exposed to p-type impurities. Consequently, the designer selects those moat regions that are to become p-type with Mask #4, which is termed the p^+ select mask. The n^+ select mask (Mask #5), which is used to pattern those regions of moat that are to become n-type, is automatically generated from the complement of the p^+ select mask intersected with the moat (active) mask. p^+ select is used in the substrate to form p-channel transistors and interconnects and is used in the p-well to provide ohmic contact to the p-well as well as for guard rings.

Correspondingly, n^+ select is used in the p-well to form n-channel transistors and interconnects, and in the substrate to make top ohmic contacts as well as additional guard rings. Further comments about guard rings and their role in latch-up protection appear in Section 2.4. As was the case in the NMOS process, the polysilicon layer or layers are patterned prior to the p^+ and n^+ diffusions. The polysilicon that lies in the moat serves as a diffusion mask for these diffusions and provides self-alignment of the gate with the drain and source regions.

The n^+ and p^+ diffusions are much shallower than the p-well diffusion and are typically in the 5000 Å and 7000 Å ranges, respectively. Following the p^+ and n^+ diffusions, which occur prior to Step 36 in the process scenario, the cross-sectional profile is as shown in Fig. 2B.1*e*. The n-channel and p-channel transistors, along with the ohmic contacts and guard rings, are clearly visible at this stage. A thick insulating layer, which is a deposited oxide (often PSG), is then placed over the entire wafer. This insulating layer is about 6000 Å thick and serves as an insulator between the uppermost polysilicon layer and the subsequent metal layer. This causes a further thickening of the field oxide and is depicted above the dashed interface of the field oxide layer shown in Fig. 2B.1*f*.

Mask #6 is used to pattern contact openings. Areas unprotected by photoresist after patterning are etched away. This etch will consume insulating layers but is stopped by either polysilicon or the silicon substrate. This provides for metal contact of either a polysilicon layer or a p^+ or n^+ diffusion depending on which is the uppermost layer present.

After the contacts are opened, metal is applied uniformly to the wafer and patterned with Mask #7. This is termed the *metal mask* (or, if subsequent metal layers are to be added, the *metal 1 mask*). This corresponds to Step 48 in the process scenario and the cross section of Fig. 2B.1*f*.

An optional second metal can be added at this stage. This requires two additional mask steps, one for making contact with underlying metal 1 and the other for patterning the second metal layer. The mask used to pattern the contact openings, or vias, between the two metal layers is termed the *via mask*. Polyimide is often used as the insulating layer between the two metals because it offers advantages in step coverage over other commonly used insulating layers.

Following application of a passivation layer, often termed p-glass, the passivation is opened above the bonding pads to provide for electrical contact from the top with Mask #8. This is often termed the *glass mask*. The pad layout is similar to that discussed for the NMOS process and depicted in Fig. 2A.1*g*.

This completes the CMOS processing steps for the generic CMOS process scenario of Table 2B.1. The resistor, capacitor, n-channel MOSFET, and p-channel MOSFET should be apparent from Figs. 2B.1*a* and 2B.1*f*.

2.2.1*c* Practical Process Considerations

The equipment needed for the CMOS process is basically the same as is needed for the NMOS process previously described. With this equipment the minimum feature size for the CMOS process is comparable to that for the NMOS process. It should be noted that eight masks and considerably more processing steps than are required for the basic six-mask NMOS process are needed for this CMOS process. In addition, it will be seen later that considerably more area is required for the same number of devices in a CMOS process than in an NMOS process with the same feature size. The increase in size is due largely to the required size of the large p-tubs and the n- and p-type guard rings. These increases in area are, however, often offset by less complicated designs and/or the superior performance that is attainable with the CMOS process.

Several physical and processing-dependent material characteristics cause the physical MOSFET to differ from the ideal. The processing-dependent material characteristics will be considered first.

WIDTH AND LENGTH REDUCTION. A typical cross section of the n-channel MOSFET along EE′ and FF′ of Fig. 2B.1*a* is compared with the ideal in Fig. 2.2-6. These cross sections are intentionally not to scale so that they will better illustrate the actual characteristics.

It will be seen later that the width and length of the MOSFET are key parameters at the control of the designer that play a major role in device performance. The width, W, is the width of the moat, or active, region as depicted in Fig. 2.2-6*a*, which corresponds to the EE′ cross section of the MOSFET, and the length L is the distance between the drain and source diffusions, as indicated in the FF′ cross section of Fig. 2.2-6*c*. It should be emphasized that the device dimensions are determined by the size of the *intersection* of the poly mask and the active mask and not by the dimensions of the poly pattern that forms the gate.

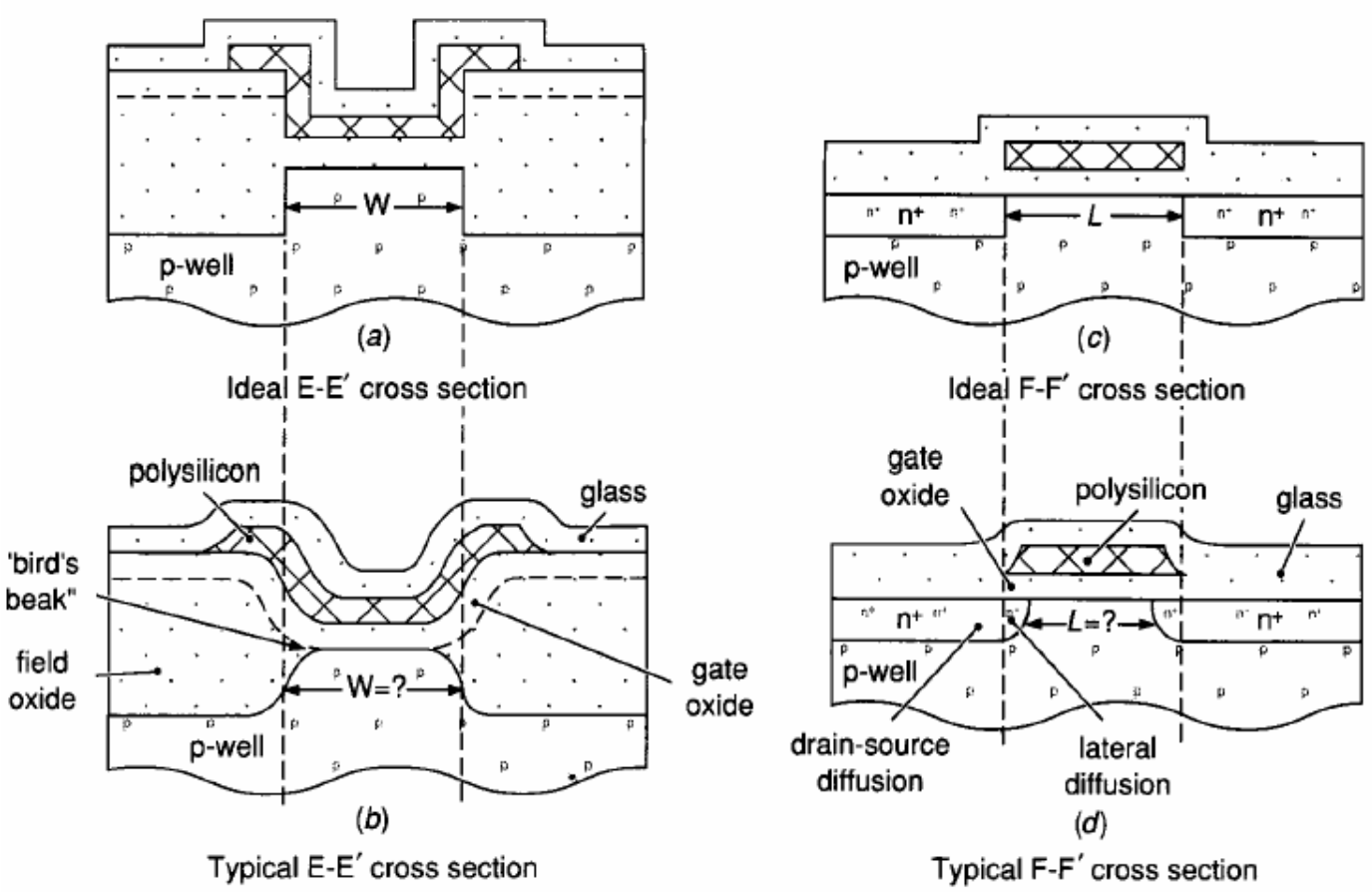


FIGURE 2.2-6
Width and length reduction in MOSFETS.

In the typical cross section of Fig. 2.2-6*b*, it can be seen that during the field oxide growth, encroachment into the active region effectively reduced the width of the transistor. This oxide encroachment is termed *bird's beaking* due to the distinctive shape of the encroachment. This is particularly troublesome because the width of the transistor is no longer precisely defined and because the exact amount of width reduction is not easily controllable. A second factor that affects the effective width is the accuracy with which the protective Si_3N_4 layer used to pattern the field oxide can be controlled. The effective width of this layer is affected by both the patterning of the photoresist and the problems associated with etching that were discussed in Section 2.1. In addition, since a thin SiO_2 layer (200–800 Å) is applied prior to the Si_3N_4 to minimize mechanical stress at the Si_3N_4 interface, the encroachment of the SiO_2 growth also limits accuracy.

Similar problems in controlling the length of the transistor exist, as indicated in Figs. 2.2-6*c* and *d*. The major source of length reduction is associated with the lateral diffusion of the drain and source diffusions, which are difficult to precisely control. Assuming the lateral and vertical diffusion rates are equal and that the diffusion depth is 5000 Å, the total length reduction due to lateral diffusion, since it diffuses in from both ends, would be around 1 μ. This is very significant and problematic in short channel transistors. Other factors that affect the effective length are the accuracy in patterning the photoresist that defines the polysilicon gate length and the accuracy in controlling the polysilicon gate etch itself.

In summary, both length and width reduction are inherent with existing processing technologies. Although they can be partially compensated for by considering these reductions during design or automatically adjusting (termed *size-adjust*) the geometrical database to over- or undersize the appropriate mask geometries, these effects are difficult to precisely control, and the exact width and length of the device are difficult to define. These effects are particularly troublesome for small geometries with device dimensions in the 1 μ or smaller range. Partial compensation with the mask size-adjust is often provided, thus allowing the designer to assume that the nominal value of the actual dimensions on silicon agree with those specified on the design.

LATERAL WELL DIFFUSION. Lateral diffusion associated with the creation of the p-well also deserves mention. The depth of the p-well is about 3 μ and, the lateral diffusion associated with the well formation is comparable. Although not a major factor limiting device performance, this lateral diffusion consumes considerable surface area and forces the designer to leave a large distance between isolated p-wells and between any p-well and p^+ diffusion in the substrate, thus increasing chip cost by increasing die area. One way to partially minimize the impact of the large amount of area associated with well boundaries is to group large numbers of n-channel transistors into a single p-well when the wells for these transistors are to be tied to the same potential. Tradeoffs between these area savings and the corresponding increase in interconnect area must be made.

LATCH-UP. The physics of layered doped silicon is also problematic. It is well known that a four-layer sandwich of doped material, npnp or pnpn, forms a Silicon Controlled Rectifier (SCR). Once an SCR is "fired" (switched to on conducting state), it continues to conduct until the gate signal is removed and current flow is interrupted. Several parasitic bipolar transistors and an SCR are identified on the cross section of Fig. 2.2-7 which is based on Fig. 2B.1*f*. Several diodes can also be identified. Although this CMOS process has not been optimized for obtaining good performance of these bipolar devices, there are limited practical applications of some of the diodes and bipolar transistors. The SCR, however, is very undesirable and if it is caused to fire, excessive current will usually flow, causing destructive failure of the integrated circuit. The firing of the SCR is termed *latch-up* in CMOS circuits. The CMOS designer must make certain that latch-up cannot occur in any design.

Latch-up problems are strongly layout dependent. A theoretical treatment of the latch-up problem is beyond the scope of this text, but a thorough understanding of this problem is not needed for successfully designing CMOS circuits provided the designer is familiar with layout techniques that circumvent the problem. Guard rings are widely used to prevent latch-up. Exact requirements for guard ring

placement will be determined once a particular CMOS process is defined. In some processes, separate and additional n^+ and/or p^+ diffusions are included specifically for guard ring formation. This requires additional masks and additional processing steps. In the CMOS process discussed in this section, no additional masking or processing steps are required since the normal drain and source diffusions are also used to fabricate guard rings.

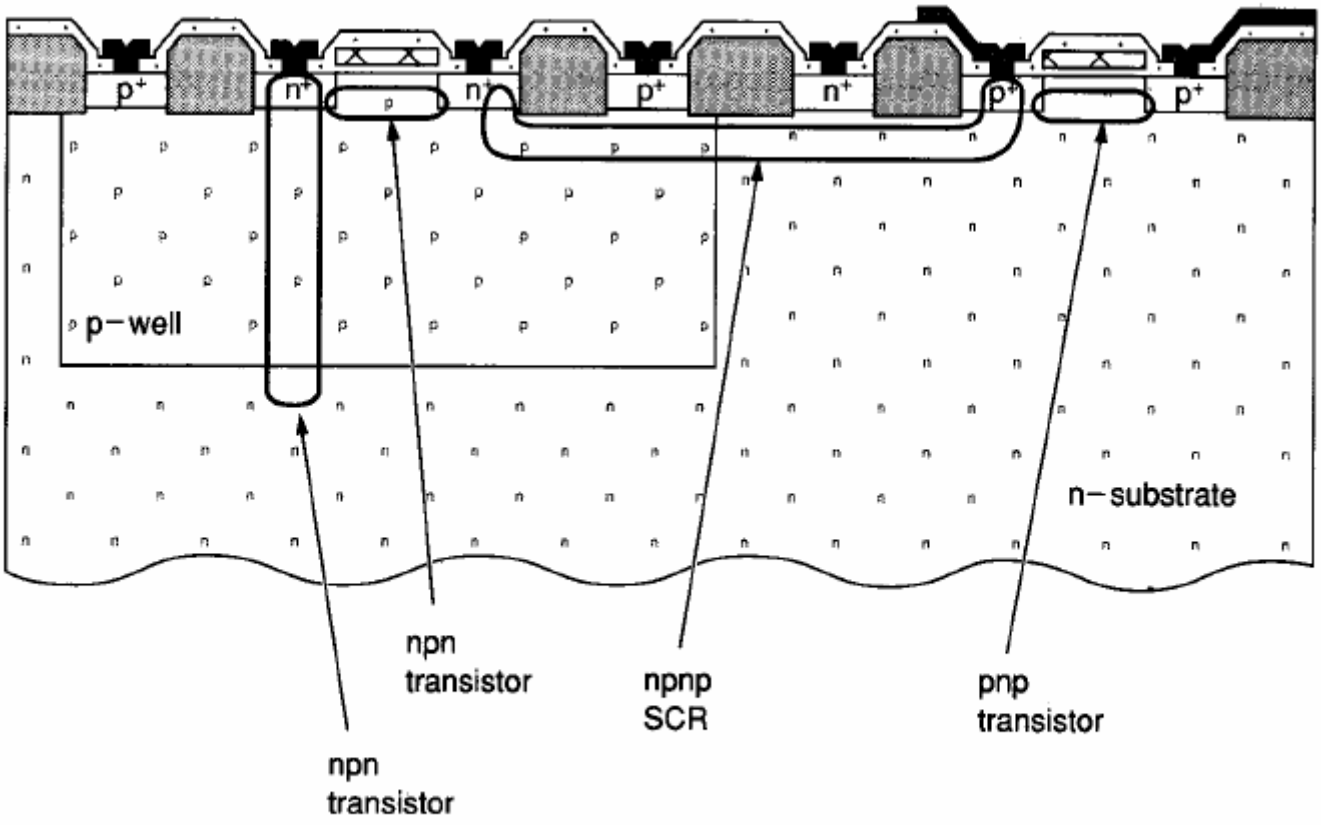


FIGURE 2.2-7
Parasitic transistors in a p-well CMOS process.

One way to obtain latch-up protection in the generic CMOS process of this section is to completely encircle every p-well with a p^+ guard ring. Such a guard ring is shown in Fig. 2B.1*a* around the periphery of the p-well. Metal contact is made as often as possible to this guard ring to further reduce resistance. The guard ring will then typically be connected via metallization to the lowest dc potential in the circuit—V_{SS} or ground, for example. Although not shown in Fig. 2B.1*a*, encircling the p-well with an n^+ guard ring provides additional protection and is also desirable. A partial n^+ guard ring separating the p-well from the p-channel substrate transistor can be seen in the same figure. As before, numerous metal contacts are made to this guard and it is subsequently tied to the highest potential in the circuit.

Breaks in a guard ring must be avoided since these breaks could provide a path for breakdown. Consider first the p^+ guard ring in the p-well. Breaks can occur one of two ways. The most obvious is to exclude a segment from either the moat mask or the p^+ select mask. The other way is to cross the guard ring *anywhere* with polysilicon in the process described in Table 2B.1. Such a crossing will cause a break because the polysilicon is patterned prior to the p^+ diffusion and serves as a mask to this diffusion. The break would thus occur under any polysilicon crossing of the intended guard ring. The exclusion of polysilicon crossing of the guard ring is undesirable from a circuit designer's viewpoint because it complicates interconnection between devices in the p-well and those outside the p-well; all interconnection crossings must be made of metal to avoid breaking the guard ring. Polysilicon crossing of guard rings in other process scenarios where a separate p^+ guard ring diffusion is available may be permissible. Correspondingly, breaks in the n^+ guard ring will occur if a segment is omitted from the active mask, if it is crossed with p^+ select, or if it is crossed with polysilicon.

Although complete enclosure of the p-well with the n^+ guard ring is desirable, some designers using the generic CMOS process described in this section use only the p^+ guard ring or have the p^+ guard ring and include the n^+ guard material only between the p-channel transistors and the p-well, as depicted in Fig. 2B.1*a*.

INPUT PROTECTION. Static breakdown is also of concern, and protection of inputs must be provided to prevent destructive breakdown when handling the devices. The major sources of concern are inputs that have a direct connection to a region separated from the rest of the circuit only by thin oxide, such as gate oxide or poly–poly oxide, with no direct connection to any diffused region. Such inputs would include the gates of any transistors or any connection to a floating polysilicon capacitor electrode. The breakdown is due to a destructive breakdown of this thin oxide due to electric fields that exceed the oxide breakdown voltage. As stated in Chapter 1, silicon dioxide will break down when electric fields are in the 5 MV/cm to 10 MV/cm range. With 1000 Å gate oxides, this would occur for voltage inputs in the 50 V to 100 V range. Although these are beyond the maximum allowable input voltages specified for a typical 3 μ CMOS process, these voltages are much less than the static voltages experienced when handling these chips. The problem is even worse for thinner gate oxides. Such breakdown is destructive and must be prevented. Input protection circuitry is used for this purpose. This input protection must not interfere with the normal operation of the circuit. A single simple protection circuit is typically developed and is used repeatedly by connecting it to each pad that requires protection.

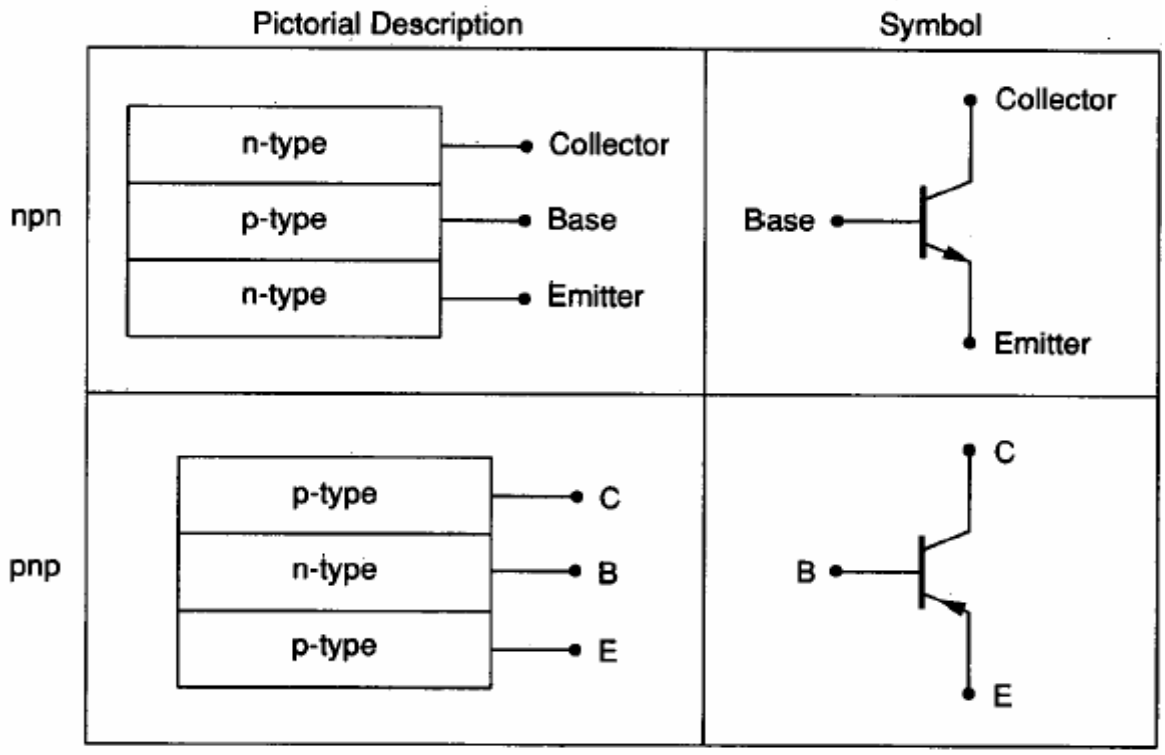
One common protection scheme involves connecting the input pads through a small polysilicon resistor to a reverse-biased diode that nondestructively breaks down at voltages below the critical gate oxide breakdown voltage. The node that is to be protected then becomes an internal node coincident with the node corresponding to the interconnection between the protection resistor and the diode. The resistor is used to safely limit peak current flow in the protection diode. In the CMOS process described in this section, this diode would be constructed by putting an n^+ diffusion in a p-well with a p^+ select guard ring around the periphery of the well. This guard ring would be connected to the lowest potential in the circuit, typically ground or V_{SS}, and the n^+ diffusion would be connected to the pad that is to be protected through the polysilicon resistor. This protection circuit provides protection through the reverse breakdown voltage of the diode if the input is positive and through normal forward-biased diode conduction if the input is negative. For the NMOS process, single diode protection can be attained by connecting the critical node through a polysilicon resistor to an n^+ diffusion. No guard ring is available or required in an NMOS circuit.

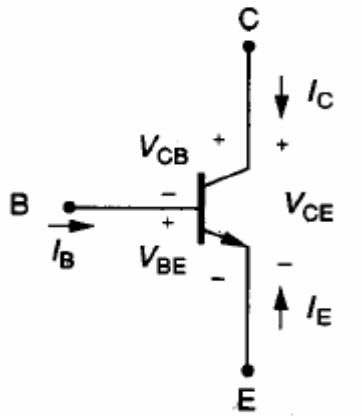
An alternative that provides all protection through normal forward-biased diode conduction is obtained if a second diode of opposite polarity shunts the diode just described. This diode is constructed from a p^+ diffusion in the substrate, with the n-substrate connected to the highest potential in the circuit and the p^+ diffusion connected to the intersection node of the first diode and the polysilicon resistor. It is recommended that this p^+ diffusion be encircled by an n^+ guard ring, which also would be connected to the substrate. Under normal operation the diodes in the input circuitry do not conduct, so the input protection is ideally transparent to the user. Actually, the diodes do contribute to a small amount of leakage current. They also contribute to a small parasitic capacitance connected to an ac ground, which may be of limited concern in some applications.

Although the input protection schemes discussed could be used on any input or output pad, such circuitry is generally not required on pads that are already directly connected to a diffusion region, even if they are also connected to layers separated by thin oxide from other nodes in the circuit, because the diffused region itself forms part of the diode and thus provides inherent self-protection. Nevertheless, care should always be exercised when handling any MOS devices, even if good circuit-level protection has been included, to reduce the chance of destroying the integrated circuit by static breakdown.

2.2.2 Bipolar Process

The basic active devices in the bipolar process are the npn and pnp transistors. These names are descriptive since the devices are constructed with three layers of n- or p-type semiconductor material, with the middle layer different from the other two. These layers can be fabricated either laterally or vertically. A simplified pictorial description of these transistors, including the established symbols for the devices, appears in Fig. 2.2-8. Several excellent references discuss the basic operation of the BJT.[3,12–17] The modeling of the BJT is discussed in Chapter 3. As will be seen later, the characteristics of the collector and emitter regions as well as their geometries are intentionally different and as such the designation of the collector and emitter contacts is not arbitrary. The convention that has been established for designating the collector and emitter contacts will be discussed in Chapter 3.





electrical variable convention (npn or pnp)

FIGURE 2.2-8
Bipolar transistors.

The components available in the bipolar process are

1. npn bipolar transistors.
2. pnp bipolar transistors.
3. Resistors.
4. Capacitors.
5. Diodes.
6. Zener diodes.
7. Junction Field Effect Transistors (JFETs)—not available in all bipolar processes.

A familiarity with the process is crucial to utilizing this wide variety of components in the design of an integrated circuit. Unlike discrete component circuit design with these same devices, whose characteristics can be specified over a wide range and which can be connected in any manner, the basic characteristics of the devices available for bipolar integrated circuits are determined by the process and the range of practical values and parameters is severely limited. In addition, the methods of interconnection are strictly limited, the basic devices have characteristics that are quite temperature dependent, and the passive component values are typically somewhat dependent on the signal applied. In spite of these restrictions (to be discussed later), very clever analog and digital bipolar integrated circuits have evolved. The bipolar process is used for the popular TTL, ECL, and I^2L digital logic families as well as a host of linear integrated circuits, including the 741 operational amplifier, the 723 voltage regulator, and the 565 phase locked loop. Although minor variances in the processing steps are common, the major differences are in device sizes and impurity concentrations and profiles. The discussion of a typical seven-mask bipolar process follows. The major process steps are outlined in Table 2C.1 of Appendix 2C.

The construction of npn transistors, pnp transistors, resistors, and capacitors will be considered. The location of these components can be seen in the top view containing mask information shown in Fig. 2C.1*a* of Appendix 2C.

The starting point in this bipolar process is a clean, polished p-type silicon wafer. A layer of SiO_2 is first grown over the wafer. Following application of a layer of photoresist, Mask #1 is used to pattern the n^+ buried layer. The n^+ buried layer serves the purpose of decreasing collector resistance and minimizing the parasitic current flow from collector to substrate in npn transistors. It also helps decrease the base resistance in lateral pnp transistors. Either a deposition or implant, followed by a drive in diffusion, can be used to introduce the n-type impurities into the substrate through the openings provided by Mask #1. A layer of oxide, which grows during the diffusion, is then stripped. After the n^+ diffusion the wafer takes the form shown in Fig. 2C.1*b*.

An n-type epitaxial (crystalline) layer is then grown over the entire wafer. The thickness of this layer typically varies between 2 μ and 15 μ, with the thinner layers used for digital circuits and the thicker layers for analog circuits. This layer will be used for the collector region in npn transistors. The epitaxial layer is shown in Fig. 2C.1*c*. Note that some of the impurities in the buried layer have migrated (or *out-diffused*) into the epitaxial layer during its growth. A thick layer of SiO_2 (typically 5000 Å) is then grown over the entire surface.

Mask #2 is used to pattern the SiO_2 layer for the p^+ isolation diffusion. SiO_2 is etched from the areas not photographically protected by Mask #2 to allow for this drive in diffusion following a p^+ deposition. The p^+ isolation diffusion is used to electrically separate adjacent transistors. It is wide and deep since it must completely penetrate the epitaxial layer to provide the required isolation. The wafer at this stage is as shown in Fig. 2C.1*d*. Although the isolation diffusion is shown with vertical edges in the figure, lateral diffusion, typically comparable to the vertical diffusion, causes significant out-diffusion laterally under the oxide layer, thus making the top of the channel stop considerably wider than the bottom. Following this diffusion, another thick layer of SiO_2 is grown over the entire wafer.

An optional shallow, high-resistance p-diffusion could be added at this step. This is not depicted in Fig. 2C.1 but is listed as an option in Table 2C.1. This step would provide a mechanism for making practical diffused resistors in the 1 kΩ to 20 kΩ range. A typical sheet resistance of this region would be 1 kΩ/□ to 2 kΩ/□ . Mask #A in Table 2C.1 is used to pattern these regions.

Mask #3 is used to pattern the SiO_2 layer and define the base regions for the npn transistors as well as the collector and emitter regions for lateral pnp devices. A p-type deposition and a subsequent drive in diffusion create these regions in the unprotected areas. This diffusion is much shallower than was the isolation diffusion and must not penetrate the epitaxial layer. The isolation mask openings provided by Mask #2 are typically reopened with Mask #3 to provide a few additional p-type impurities. The wafer at this stage is shown in Fig. 2C.1*e*.

Following growth of another layer of SiO_2, Mask #4 is used to pattern the emitter regions for the npn transistors. An n^+ deposition followed by a drive in diffusion creates the emitter regions. Openings are also made in the oxide above the collector to add small n^+ wells in the lightly doped collector region to provide for better electrical contact from the surface. The integrated circuit at this stage is as depicted in Fig. 2C.1*f*. The emitter diffusions must be shallow so as not to penetrate the relatively shallow p-type base regions already created. The amount and profile of the impurities in the n^+ emitter regions and the thickness of the p-type base region, which is now sandwiched between the n^+ emitter and the n-collector, strongly influence the gain of the transistor.

Mask #5 is used to pattern contact openings to allow for top contact of the circuit with the metallization. The entire circuit is then covered with a thin layer of metal. Mask #6 is used to pattern the metal, followed by the addition of a passivation layer. The completed cross-sectional view of the four components under consideration is shown in Fig. 2C.1*g*. Mask #7 patterns pad openings to allow for electrical contact to the bonding pads.

Two modifications of this process deserve mention. One involves adding a *deep collector* diffusion. This requires an additional masking step and is used to diffuse impurities under the area where the collector contacts are to be made. This step would occur either before or after the isolation diffusion and is used to extend n^+ impurities all the way from the surface to the buried layer. Since this is such a deep diffusion, an area penalty in the collector is experienced. The deep diffusion is used to reduce collector resistance in high-current applications. The second modification involves adding an additional p-diffusion in the p-channel stops. This also requires an additional mask step and is used to avoid surface inversion in high-voltage parts. With the exception of open collector circuits, this step is not common in basic logic parts.

The npn and pnp transistors in this process are depicted in Fig. 2.2-9. The npn transistor is called a vertical npn device since the emitter, base, and collector regions are stacked vertically. It can now be seen that the n^+ buried layer decreases the collector resistance that must be modeled in series with the collector.

The pnp transistor is called a lateral device since it is stacked laterally (horizontally). The base width cannot practically be made as narrow and the base area is not as accurately controllable as for the npn device. In addition, the emitter and collector regions must have the same impurity profile. The characteristics of the lateral transistors are generally considered poorer than those of the vertical devices. Other pnp transistors, not shown in Fig. 2C.1, can be constructed by using the p-type base diffusion as the emitter, the n-type epitaxial layer as the base, and the p-type substrate as the collector. These devices are called substrate transistors. The performance of these devices is also mediocre, and applications are restricted since all collectors of substrate transistors are common.

The capacitor that was constructed in Fig. 2C.1 may at the outset appear to be merely a diode. It would serve the purpose of a diode, even though the area is considerably more than may be required in most applications. When reverse biased, however, the depletion layer forms the dielectric, and the p and n regions on either side form the capacitor plates. Capacitors made like this, with total

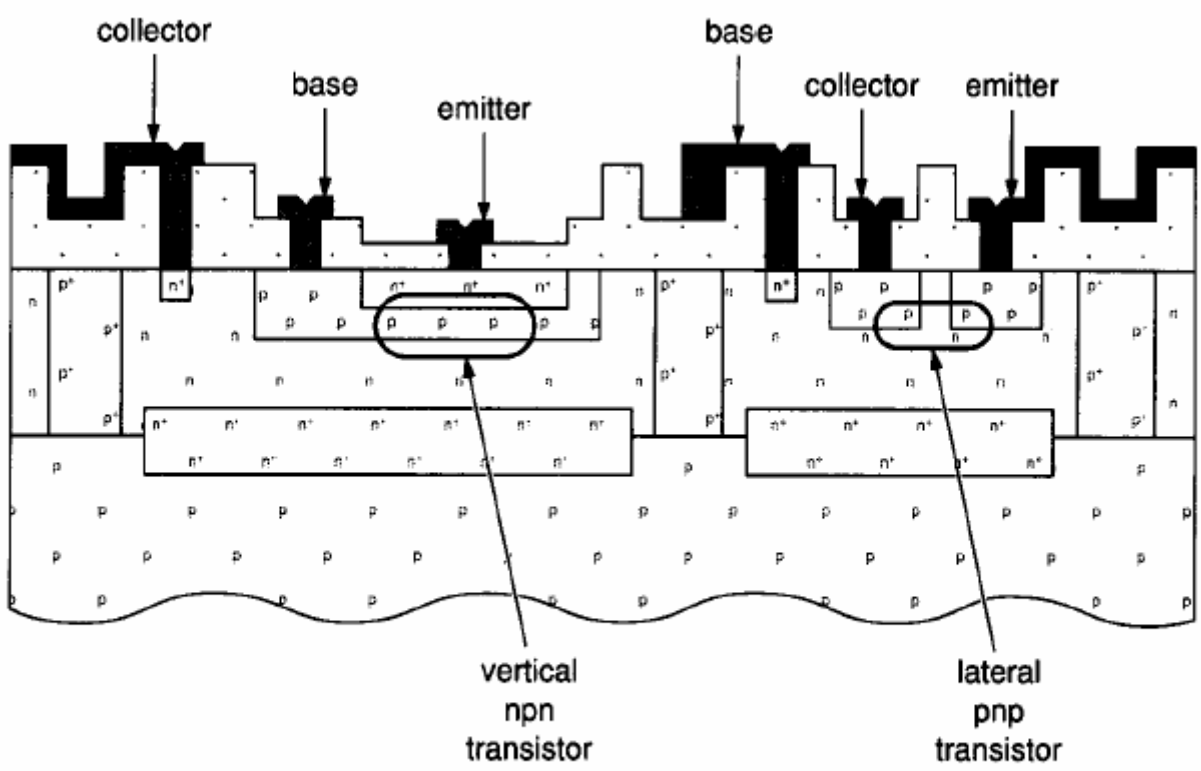


FIGURE 2.2-9
Vertical and lateral transistors in a bipolar process.

capacitances from the sub-picofarad to the 100 picofarad range, have proven practical. The capacitors are not without limitations, however. The requirement that the junction must always be reverse biased severely limits the interconnection flexibility of this device. The width of the depletion layer is voltage dependent, making the capacitance nonlinear. Temperature also affects the capacitance value. Finally, the base–emitter junction typically breaks down with a reverse bias of about 7 V, limiting the maximum voltage that can be applied to the capacitor. The base–collector junction can also be used as a capacitor. Its characteristics are very similar to the base-emitter capacitor with the exception that it offers an increased reverse breakdown at the expense of a lower capacitance density. Junction capacitors will be discussed in more detail in Chapter 3. An alternative to the junction capacitor would be a metal–oxide–semiconductor capacitor formed between the metal and the n^+ emitter diffusion. Although the characteristics of this capacitor would be better than those of the junction capacitors, an extra mask step is generally required to provide a means of selectively stripping the thick oxide above the emitter region so that a thin oxide can be regrown. The thin oxide is needed to get the capacitance density up to a practical level.

The resistor of Fig. 2C.1 is actually just a serpentined strip of the lightly doped p-type base diffusion. The underlying n-type epitaxial layer is generally contacted and taken to the highest potential in the circuit to prevent current flow into this region. Several other techniques for fabricating resistors in this process are available. They will be discussed in Chapter 3.

Although minimum feature sizes are comparable for the bipolar and MOS processes, standard bipolar processes require more area per device than do the NMOS processes. A major reason for this increased area is the deep and wide p^+ channel stops that are required for device isolation in standard bipolar processes. An alternative bipolar process using trench isolation[23] is available which offers a significant improvement in component density over the standard bipolar process.

2.2.3 Hybrid Technology

The hybrid approach to integrated circuit design involves attaching two or more integrated circuit dies (typically of different types), along with some discrete components in some cases, in a single package to form what is called a hybrid integrated circuit. It is often, and desirably, transparent to the consumer whether the circuit is monolithic or hybrid; in some cases, however, the hybrid packages are considerably larger. The hybrid integrated circuit is typically more costly than the monolithic structures. The extra cost and size of hybrid integrated circuits is offset, in some demanding applications, by improved performance capabilities.

Hybrid circuits containing discrete components occupy considerably less area than the conventional PC board/discrete component approach. They have played a major role in demanding analog signal processing applications such as high-resolution A/D and D/A converters and precision active filters. Tolerances, temperature dependence, and area-induced component value limits for resistors and capacitors in standard MOS and bipolar processes have limited the development of monolithic integrated circuits for precision continuous-time signal processing. Thick film and thin film passive components have reasonable tolerances, are easily trimmable, have acceptable temperature coefficients that can be tailored for tracking, and offer reasonable tradeoffs between area required and component values. These thick film and thin film networks are commonly used for the passive components in hybrid integrated circuits. A discussion of thick film and thin film processing technologies follows.

THICK FILM CIRCUITS. The thick film technology is relatively old, requires considerable area compared to monolithic circuits, can be used for relatively high-power applications, and can be applied at relatively high frequencies (up to 1 GHz) although it is typically limited to a few MHz. The increased area required by the thick film circuits is offset by the reduced cost in equipment and processing materials required for the thick film process, the latter being a small fraction of that required for either bipolar or MOS processes.

The components available in a thick film process are resistors and capacitors along with conducting interconnects. Layers of different material are successively screened onto an insulating substrate. These materials are used for resistors and conductors as well as for the dielectrics of capacitors.

The number of resistive layers varies but practical limitations generally restrict this to at most three. Typical thickness of these layers (called pastes or inks) is about 20 μ, but they may vary considerably by design. The actual thickness of these layers is not accurately controllable ($\pm$ 30%) due to limitations in the screening process itself.

The thick film process offers the most advantages for resistor fabrication. Although capacitors are often included, the electrical characteristics of thick film capacitors are not outstanding and the capacitance density is quite low. Discrete chip capacitors, which have much better characteristics than their thick film counterparts, are often bonded to thick film resistive networks in hybrid circuit applications.

The minimum conductor width in a typical thick film process is about 250 μ, and minimum resistor widths are about 1250 μ. It can be seen that these are orders of magnitude larger than the corresponding minimum feature sizes for the MOS and bipolar processes (1–5 μ).

Screening involves forcing the paste through small holes in a tightly stretched piece of fabric called a screen, typically constructed of stainless steel. The grid is quite regular. The spacing of the holes can be specified, but practical physical limitations relating to both the mechanical characteristics of the steel and the physical characteristics of the inks prevent the use of extremely fine meshes. Screens with a grid spacing ranging from 100 to 300 filaments/inch are typical. This spacing restricts thick film resolution to somewhere around 500 μ. Where paste is not desired, holes in the screen are plugged by a mask. A squeegee is used to force the ink through the unrestricted areas. Following screening, each layer is fired to harden it.

Inks are available with sheet resistances that satisfy the equation

$$1\ \Omega/\square < R_{\square} < 10\ \text{M}\Omega/\square \qquad (2.2\text{-}9)$$

for fired layers 20 μ thick. This large latitude in ink characteristics allows for a wide range of resistor values. Since only one type of ink can be used for each resistor layer, the tradeoffs between area and sheet resistance must be considered when specifying the ink sheet resistances. Even though adjusting the length is a convenient means of establishing resistor values, long thick film resistors are to be avoided because they develop "hot spots" and are difficult to trim. The "hot spots" are caused at regions where the resistive layer is a little thinner and/or narrower than surrounding regions. This causes an increased resistance in this small region, which under constant current causes increased local power dissipation. This power dissipation causes heating, which typically further increases the resistance and power dissipation. Heating causes deterioration of the film layer at these points. Deterioration in these regions can eventually result in device failure. Short, wide resistors are also to be avoided. The main problem with short, wide resistors is the inability to accurately specify the size of the resistor since the contacts will overlap with a considerable portion of the device. A reasonable rule of thumb for the allowable width (W) / length (L) ratio for a rectangular resistor is

$$\frac{1}{10} < \frac{W}{L} < 3 \qquad (2.2\text{-}10)$$

Although the W/L ratio is constrained, the values for W and L remain to be specified within the design rules of the process. It is a good practice to make the resistors large if the area is available to minimize edge roughness effects, increase power dissipation capability, and make trimming easier.

Serpentined patterns such as shown in Problem 2.11 should also be avoided with thick film technology. This is due to the increased current density that will result from current crowding at the inner corners of serpentined structures.

A capacitor is constructed by screening a conductive layer, followed by a dielectric, followed by another conductor. The dielectric is generally applied in two coats to minimize pinholes, which would short the capacitor plates together. With the two layers of dielectric, the chances of a pinhole coincident with both layers are greatly reduced. Since a thick film capacitor is actually a parallel plate capacitor, the capacitance is given by

$$C = \epsilon_R \epsilon_0 \frac{A}{t} \qquad (2.2\text{-}11)$$

where $\epsilon_0 = 8.854$ pF/m, A is the area of the capacitor plates, t is the dielectric thickness, and ϵ_R is the relative dielectric constant. Inks with relative dielectric constants from 10 to 1000 are available. The high dielectric constants offer a reasonable capacitance density at the expense of large and nonlinear temperature coefficients. The lower dielectric materials offer improved performance but are restricted to applications requiring small capacitors due to a low capacitance density. As in the MOS and bipolar processes, the upper plate of thick film capacitors is typically a little smaller than the lower to minimize capacitance changes caused by minor misalignments. The two conductive layers used for the capacitor plates also serve as interconnects. If crossovers of two conductors are required, the dielectric layer can be used as an insulator at the expense of creating a small parasitic capacitor at the crossover.

The screen geometries, along with a cross-sectional view of a typical thick film process, are depicted in Fig. 2D.1 of Appendix 2D. This process has two resistive screenings as well as two conductive layers and a dielectric for capacitor fabrication. The major process steps are listed in Table 2D.1. Process parameters and characteristics, along with design rules for a typical thick film process, are also given in Appendix 2D.

Thick film resistors can be trimmed with a laser or by abrasion. These trims, which can be very accurate, remove part of the thick film layer and thus increase the resistance. For this reason, resistors that are to be trimmed are typically targeted to be undersized in value by about 40% to guarantee trimmability in spite of process variations.

Thick film capacitors can also be trimmed by abrasively removing part of the upper plate (along with some dielectric). Since this decreases the capacitance, the thick film capacitors are typically targeted to be oversized by about 40%.

THIN FILM CIRCUITS. The components available in thin film processes are resistors and capacitors, although often only resistors are included due to both the specific applications which naturally benefit from thin film technology and the practical limitations of thin film capacitors. Thin film circuits are much smaller than thick film circuits. They are similar to thick film circuits in that successive layers are applied to an insulating substrate as contrasted to the MOS and bipolar processes, where some of the processing steps involve diffusions that actually penetrate the substrate. For conductors, thin film thicknesses are typically from 100 to 500 Å although thicknesses of several thousand angstroms are occasionally used if a high conductivity is needed. Film thicknesses from 100 to 2000 Å for resistors and film thicknesses in the 3000 Å region for dielectrics of capacitors are common. Note that these film thicknesses are comparable to the thicknesses of layers applied in the MOS and bipolar processes but are orders of magnitude thinner than the 20 μ (200,000 Å) typical of thick films. The sheet resistance range for thin film resistors is typically from 50 Ω/□ to 250 Ω/□, which is considerably less than is available in thick film processes. The thin film layers are applied by uniformly coating the entire wafer with the film. Then unwanted areas are selectively patterned and etched with a photolithographic process similar to that used in the MOS and bipolar cases.

The minimum feature sizes for the thin film components are comparable to those of the MOS and bipolar processes. The temperature characteristics and performance of thin film components are quite good with the exception of the dielectrics for capacitors, which are quite lossy. "Hot spots," which were a problem with thick film circuits for long resistors, are not a major problem with thin film resistors since the thin films are typically more uniform and since thin film applications generally require smaller current flow.

Thin film circuits are much more expensive to produce than their thick film counterparts because of the sophisticated equipment that is needed for both the photolithographic process and the film depositions and etching. They are used extensively in telecommunication circuits at low frequencies but also find applications at higher frequencies (up to 30 GHz) as well.

Thin film resistors can be accurately trimmed by a laser. Thin film capacitors are not well adapted to a continuous trim although binarily weighted capacitors connected with laser-fusible links are trimmable in quantized decrements.

Thin films can also be applied on top of monolithic structures, offering considerably more performance capability than is attainable with either the thin film or monolithic approaches themselves. Problems with film technology and the decreased yield per wafer associated with both increased die area and an increased number of processing steps have slowed the development of such processes.

2.3 DESIGN RULES AND PROCESS PARAMETERS

Design rules are generally well-documented specifications listing minimum widths of features (conductor, moat, resistor, etc.), minimum spacings allowable between adjacent features, overlap requirements, and other measurements that are compatible with a given process. Factors such as mask alignment, mask nonlinearities, wafer warping, out-diffusion (lateral diffusion), oxide growth profile, lateral etch undercutting, and optical resolution and their relationships with performance and yield are considered when specifying the design rules for a process. It is not our intention in this text to rigorously investigate the technical details about how design rules are derived but rather to consider the design rules and process parameters as a set of constraints within which the circuit designer must work. This is justifiable because the basic format of the design rules and process parameters remains relatively fixed, with changes in the process contributing only to numerical perturbations of the design rules and process parameters. The design rules, the process parameters, and their relationship with device characteristics serve as an interface between the process engineers and the circuit designers. Both groups, along with representatives from marketing (since yield is affected by the design rules), have input into the evolution of these interfaces.

Although the minimum feature sizes, which ultimately determine the design rules, have been steadily decreasing with time to the benefit of yield and production costs, it is important that designers adhere to the design rules once a process has been selected for a particular project. Most large semiconductor houses have developed or purchased sophisticated computer software to verify that layouts violate no design rules. In the process of verifying design rules, it is often the case that layout errors are also detected since these errors will often violate a design rule. For large designs that involve thousands of transistors, it is crucial that these verifications be made since a single design rule violation or layout error will often be fatal (i.e., the circuit won't work). The importance of adhering to design rules and utilizing verification programs for simple as well as complicated designs cannot be overemphasized. If design rules are intentionally violated, the verification software cannot be fully utilized and perhaps not utilized at all.

Typical sets of design rules and process parameters for the NMOS, CMOS, bipolar, thick film, and thin film processes discussed in Sec. 2.2 are summarized in Appendices 2A–2E. Some of the parameters listed in those appendices have not been defined yet but will be discussed in Chapter 3. The 3 μ NMOS parameters and the 3 μ CMOS parameters are very similar to those provided by MOSIS for their 3 μ NMOS and 3 μ CMOS processes in 1988. We have attempted to maintain most of the notation established by MOSIS to aid students who will be doing designs in a MOSIS process.

CMOS DESIGN RULES. The CMOS design rules are listed in Table 2B.2 of Appendix 2B. These rules are depicted graphically in Table 2B.3.

The design rules list guidelines (actually restrictions) about how each of the geometrical figures on each mask level align relative to each other and to other mask levels. Unless specifically stated to the contrary with the comment "exactly," all rules are minimum spacings between the corresponding geometrical figures. In general, the designer may exceed these minimum spacings to whatever degree is deemed appropriate. It should be emphasized, however, that from both cost and performance viewpoints, the die area should be as small as is practical. At this stage one might be tempted to conclude that a margin of safety, or improved reliability, could be obtained if a more conservative set of rules were established by the designer. With the exception of some matching considerations that will be discussed later in this book, few benefits are derived from following this strategy since an economically motivated margin of safety was considered when establishing the design rules. The rules were derived under the assumption that large circuits with many devices sized at the minimum allowable levels must have good performance and high yield.

In Table 2B.2, two sets of dimensions are specified. The first corresponds to those specified in a 3 μ CMOS process provided by MOSIS. The second set is in terms of the scaling parameter, λ, which characterizes the feature size of the process. The feature size (minimum poly width, active width, and metal width) is 2λ. This parameterization is used so that the design rules do not need to be rewritten as the feature size of the process shrinks. Substituting $\lambda = 1.5$ μ will give a process very similar to the 3 μ process characterized in the first column of this table. Although similar in intent, the scalable parameters listed in this table are not identical to those characterizing the MOSIS scalable CMOS process.

The mask geometries themselves may differ somewhat from the geometries specified for the corresponding geometrical feature by the designer. The exact geometries specified by the designer are termed *drawn* features. The change in feature sizes on a mask from those specified by the designer is termed *size adjust* and is undertaken in cooperation with those responsible for the processing. This allows for precompensation of effects, such as lateral etching or out-diffusion, that make the physical dimensions on silicon different from the mask dimensions. Size adjust is often used so that the "effective" dimension (the physical dimension realized after fabrication) is nominally equal to the drawn dimension. Specific comments about selected key rules follow.

p-well. The spacing (Rule 1.2) between two p-wells at different potential is very large. This allows for accommodation of the lateral diffusion. Since the p-well is very deep (3–4 μ), the lateral diffusion is also significant.

Via. The via level is used for interconnecting Metal 1 and Metal 2. The via design rules are very similar to those for the contact openings. Both involve making openings in thick oxide layers.

Pads. The dimensions for the metal bonding pads have not scaled with decreases in feature size of the processes. The size of the capillaries used for attaching bonding wires, which are several mils in diameter to accommodate the nominal 1 mil bonding wire, has not decreased significantly for a long while, thus necessitating both large bonding pads and large spacing. Smaller probe pads are often included on designs to facilitate diagnostic probing during the debugging stage.

Active. Size adjust, sometimes termed mask bias, is used to preadjust feature sizes on the active mask so that the targeted effective feature sizes are close to the drawn feature sizes. This allows for compensation of the field encroachment into the active region. Also, this size adjust thus compensates for the width reduction experienced during processing. Note that a large spacing between active and p-well is provided to accommodate for the lateral p-well diffusion.

Poly. The poly overlap rules are primarily to provide compensation for mask alignment errors between poly and active. Although the process is self-aligned, this self-alignment is achieved only for modest misalignment of devices. If, for example, the misalignment is so bad that the poly does not entirely cover the active region, then the subsequent n^+ or p^+ drain and source diffusions will create a conductive region between the drain and source that cannot be controlled by the gate. Design Rule 3.4 governs this concern. Size adjust is used on the POLY mask to do length adjustment for lateral diffusion.

If a second poly layer is available, it may not cross a boundary of POLY I. The sharp edge of POLY I would make the step coverage of this edge with the thin oxide unreliable. The integrity of the POLY II over this step would also be in question. Nor is POLY II permitted for transistor gates although such devices would likely be functional. The main purpose of POLY II is as a second plate on poly–poly capacitors although some applications as an interconnect medium or as a resistor may exist.

p^+ select. The p^+ select rules are used primarily to allow for mask misalignments with the active mask and the poly mask.

Contact. The contact openings are specified exactly rather than minimally. Although it is often the case that a large contact between two regions is desirable to reduce contact resistance, a large single contact opening is not permitted; rather,

numerous separate contacts must be used. Multiple individual contacts were used for contacting the p-well in Fig. 2B.1*a*. The reason for restricting the contact opening size and making all essentially the same can be best appreciated by considering what the contact-opening etch must accomplish. The contact openings must penetrate a thick (5000 to 7000 Å) layer of oxide. Thus, this step must consume much more oxide than is required during the thin oxide stripping steps. If a large contact opening were permitted, the central areas of these openings would be etched away before the oxide was completely removed for the smaller contact openings. As the etching continued in the small openings, any pinholes in the large open area could be further attacked by the etchant. Since the underlying poly layer thicknesses and diffusion depths are comparable in thickness to the layer that must be removed during contact openings, and since underlying thin oxides are *much* thinner, these pinholes could cause device failure. Although the probability of failure due to a single larger contact may be very low, the probability of a single failure that would render a circuit defective if a large number of these large openings were permitted may be unacceptably large. Even if the pinholes did not cause shorting, the reliability of such devices deteriorates with increased risks of premature device failures after the part is in use. For these reasons, contact openings on the gates of transistors (no contact to poly inside active) are usually not permitted either.

Contact openings to POLY II on top of POLY I are permitted. Although this type of contact is also plagued by the pinhole problem, the POLY II layer is generally used in analog applications as an upper plate of a capacitor. The total number of capacitors in these circuits is generally quite small compared to the number of transistors in a large digital circuit, thus minimizing (in the probabilistic sense) the failures due to the pinhole problem (see Problem 2.12).

2.4 LAYOUT TECHNIQUES AND PRACTICAL CONSIDERATIONS

Once a circuit design is complete, it becomes necessary to provide an area-efficient layout of the circuit to generate the masks necessary for fabrication. Although at the outset it appears that the circuit designer's job is complete at this point and that the layout can be undertaken by a draftsman (as is commonly done with PC board versions of discrete component designs), this is far from the case in IC design. Some companies provide draftsmen for this purpose who interact closely with the designers, whereas other philosophies leave this task entirely to the design engineer. In either case, the design engineer is still involved at this stage because component sizing and spacing, as well as the parasitics associated with integrated circuit components, must typically be considered in the design itself since their effects are often significant. This is particularly important in designs including analog circuitry. Even the opening sentence of this paragraph is an oversimplification of the situation since the initial design itself will likely not be complete until the layout is finished—as was indicated in Fig. 1.3-1. Although the design engineer is typically not responsible for any steps in the fabrication process once the mask information has been delivered, he or she is generally still responsible for the project until the product is in production. For complicated circuits, first silicon (the physical integrated circuits produced by the initial design) will generally not be acceptable for marketing because of either the circuit's failure to meet some specifications or total circuit failure caused by (1) a design or layout error, (2) failure of the designer to adequately account for all relevant parasitics, or (3) unacceptably low yield due to failure to center the design parameters appropriately in the actual process window.

One of the first considerations in the layout is sizing the devices as well as the interconnections. For the long rectangular resistor shown in Fig. 2.4-1, the total resistance is obtained from the sheet resistance by the expression

$$R = (L/W)R_{\square} \tag{2.4-1}$$

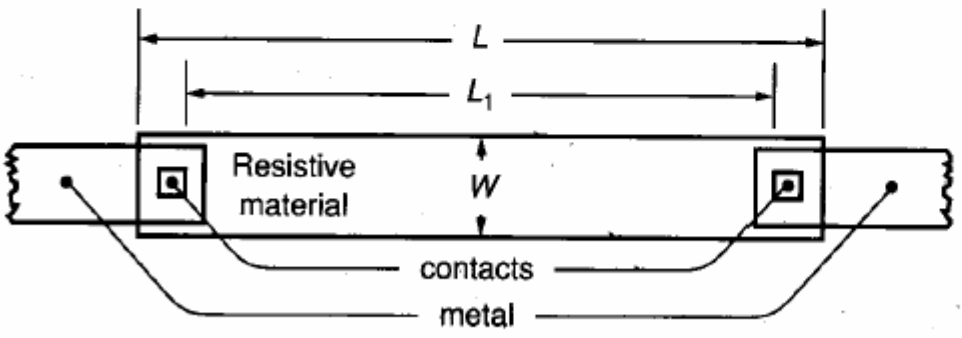


FIGURE 2.4-1
Resistor component sizing considerations.

provided that L is long enough so that the difference between L and L_1 is negligible (because $L - L_1$ is fixed by the design rules). Since the L/W ratio rather than either L or W determines the resistance, it may seem to make little difference what the values of L and W actually are, but this is often not the case. In addition to the difficulty in determining the effective length (L or L_1?) for short resistors, the edges will typically be somewhat rough due to unevenness in processing. This unevenness will cause variations in resistance. Making W and L larger reduces the relative effects of both the edge variations and the difference between L and L_1. In addition, the power-handling capability will be increased with larger devices. These improvements with larger devices are obtained at the expense of increased area, which will reduce the number of dies per wafer.

For MOSFETs, it will be shown in Chapter 3 that the length/width ratio, rather than the length and width themselves, plays one of the major roles in the device model. Again, increasing the area for a fixed length/width ratio will reduce the effects of unevenness in the edges, but this is obtained at the expense of increased area and increased gate capacitance.

As mentioned earlier, it is common practice when laying out capacitors to make one plate (typically the upper) a little smaller than the other so that the smaller plate effectively defines the plate area even if minor misalignments in the masks occur. This also makes the edge field effects easier to account for. Although the area of the smaller plate essentially determines the capacitance, the geometry of this plate itself remains to be determined. If the relationship between this capacitor value and others in the circuit is not of major importance, the shape of the capacitor will likely be selected to conform to available spaces around adjacent components, thus minimizing circuit area.

Many applications require that resistor or capacitor ratios be accurately determined. This is particularly common in analog signal-processing circuits. Ratio matching requirements of 1% to 0.1% or better are common. Although absolute component value tolerances better than 1% (or even 10%) are not currently feasible without trimming in any of the processes discussed in Sec. 2.2, the ratio accuracy specified above is attainable in some processes and is maintainable over a wide range of temperature. For resistor layouts both the individual L/W ratios, as well as the area and shape, become design parameters available to the circuit designer. The area of the resistors should be large enough to make the effects of edge roughness acceptably small, but they should be small enough to make the circuit economical and to avoid deviations caused by global variations in processing characteristics.

An example of realizing a resistor with a 3:1 ratio to R1 by three different techniques is shown in Fig. 2.4-2. Since conductors are quite good, R4 offers several distinct advantages over R2 and R3 for attaining this ratio. Comments about the different approaches follow.

1. The long resistor, R2, often cannot be conveniently placed on the circuit in an area-efficient manner. Furthermore, the question of exact length remains open and the fact that the number of contacts are not related by the 3:1 ratio limits the accuracy of R2/R1.

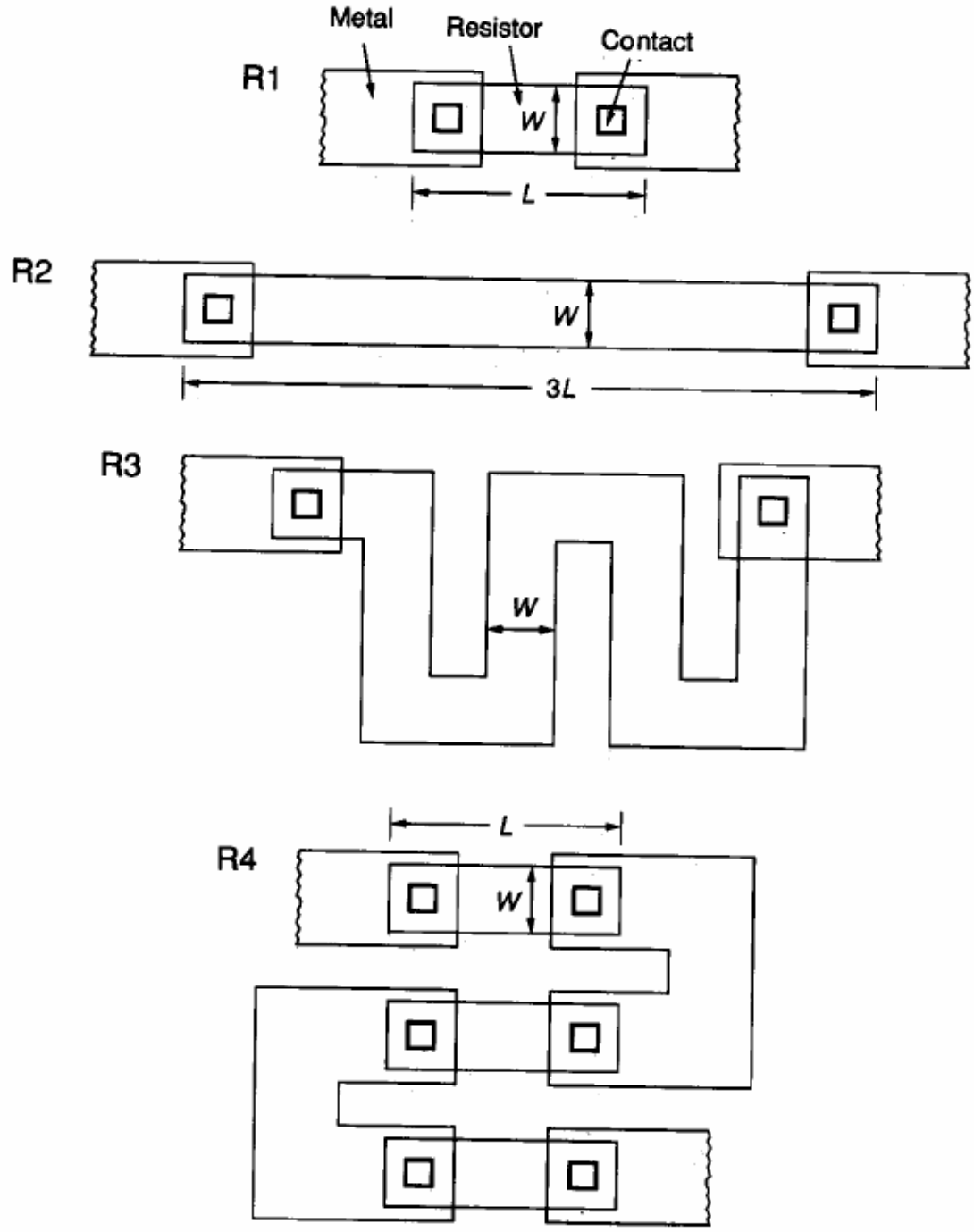


FIGURE 2.4-2
Resistor ratio-matching considerations.

2. The serpentine pattern used for R3 is quite common to keep overall aspect ratios practical. However, the difficulty in accurately accounting for the corners (using the .55 rule) and the differences in periphery length will generally make the R3/R1 ratio the least accurate of the schemes shown in the figure. The contact resistances are also not accounted for in the ratio with R1.
3. Even though the exact "length" is difficult to define, the three serpentined resistors in R4 are ideally identical to R1, so the ratio accuracy is maintained. This approach also accounts for any contact resistance associated with the contacts themselves as well as differences in temperature characteristics of the resistive and contact regions.

For realizing capacitor ratios, the area should be large enough to make the effects of edge roughness acceptably small. If the areas are too large, however, the circuits become impractical both because of the area requirements and because of the increased failure rate due to an increased likelihood of dielectric defects (one or more pinholes), which will short the capacitor plates together. Variations in dielectric thickness must also be considered if large areas are involved. In addition to maintaining the required ratio, the periphery lengths should also adhere to the ratio if possible. Fig. 2.4-3 shows three different methods of realizing a 3:1 capacitance ratio to C1. Several comments about these approaches follow.

1. C2 maintains the same geometry but the length of the perimeter differs somewhat. This will limit ratio accuracy due to variations in etching of the POLY

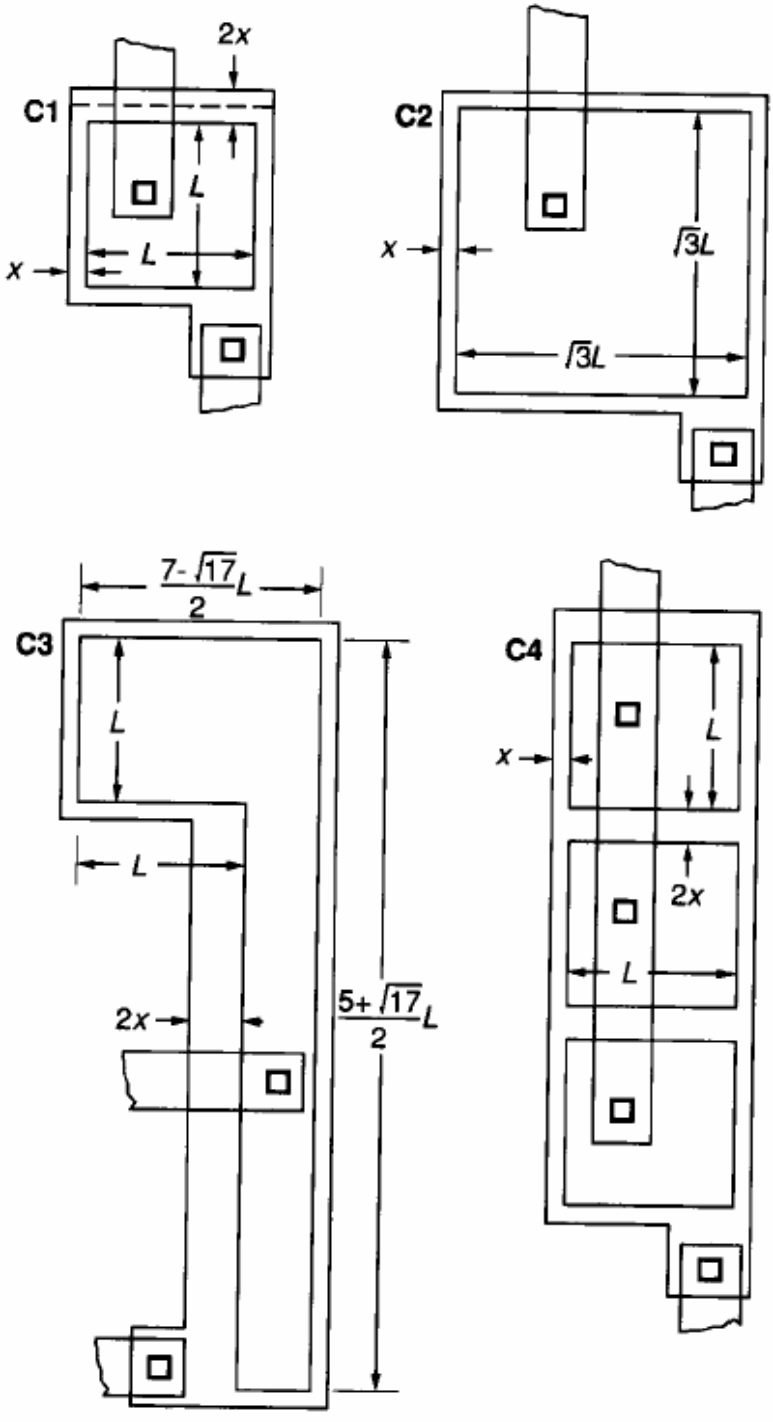


FIGURE 2.4-3
Capacitor ratio-matching considerations.

II edge. Furthermore, the ratio of the small capacitance from the conductor to the lower plate of C2 is 1:1 instead of 3:1.

2. C3 has a periphery that is three times that of C1, but the small capacitance from the conductor to the lower plate is still 1:1 with that of C1. The number of inside and outside corners in C1 and C3 does not ratio by 3:1, thus further limiting accuracy.
3. C4 has the same periphery as C1 as well as the same parasitics from the conductor to lower plates. This will give the best ratio of the three approaches. If the capacitor areas are large, even this approach will be affected by variations in dielectric thickness. Since the oxide layer is typically quite uniform locally, the ratio accuracy for large areas can be improved if each of the capacitors is further subdivided in an identical manner and the smaller capacitors for C1 and C4 are interleaved.

Parasitics can cause significant deviations in circuit performance and should be minimized during layout. Some of the common parasitics encountered are (1) the resistances associated with polysilicon and doped semiconductor regions when used as conductors, and (2) the capacitances associated with any crossover, from any conductor to substrate, and with any depletion region in a reverse-biased pn junction. Unfortunately, these resistive and capacitive parasitics can be comparable in magnitude to the desired component values to which they are connected if good layout rules are not established. Even with good layout rules, the values of these parasitics may be significant. Clever design techniques and inclusion of the unavoidable parasitics in the analysis when possible, however, help overcome some of the parasitic limitations. The parasitics associated with a depletion region of a reverse-biased pn junction are particularly troublesome since they are voltage dependent and thus difficult to properly account for in analysis and design. Even if accounted for, the parasitic capacitances often cause unwanted *cross talk* between signal paths, and the nonlinear capacitors can cause nonlinear signal distortion that may be unacceptably large.

Cleverness in layout will also often save a considerable amount of area. Even though the concern for minimizing area is always present, it is especially important to minimize area in small digital blocks that will be repeated thousands of times in high-volume VLSI circuits. Three different techniques for connecting the gate to the drain of an enhancement MOSFET are depicted in Fig. 2.4-4. The layout of Fig. 2.4-4*a*, which uses a conventional metal interconnect, requires considerable area. The circuit of Fig. 2.4-4*b* is more area-efficient, but it is not allowed in many processes because of the concern of reduced yield associated with having pinholes, which cause device failure when gate contact is made in the channel. The connection of Fig. 2.4-4*c*, which is termed a *butting contact*, is the most area-efficient, although butting contacts are only available in some processes. Note that a single contact opening is used for the butting contact.

A doped semiconductor region, a second conductor, or a polysilicon strip can be used as a conductor or as a crossover, provided that an insulating layer exists between the devices. Crossovers are often required since jumper wires are totally impractical in integrated circuit design.

In the double-poly NMOS process described in Sec. 2.2, three-level conductor stacking (metal, POLY I, and POLY II) is possible although only a two-level crossover of Poly I and metal is permitted since Poly II cannot cross a Poly I boundary. Metal over a moat diffusion will also serve as a crossover. In the CMOS process previously described, poly–metal, metal–p-well, and metal–moat (either n^+ or p^+) overlaps can all serve as two-level crossovers. In the bipolar process described metal–n^+ or metal–p (base diffusion) overlaps make

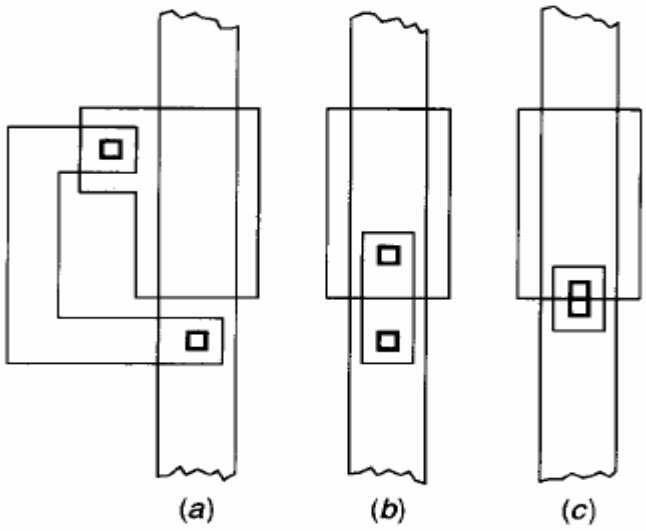


FIGURE 2.4-4
Poly-to-moat contacts: (*a*) Contact off of channel, (*b*) Contacts above channel, (*c*) Butting contacts.

practical two-level crossovers. A metal–epitaxial crossover is also possible, but the resistance associated with the epitaxial layer may be too high to make this practical in many applications.

Most design groups do the layout on an interactive graphics computer system, which serves as a workstation. As capability increases and price decreases, the trend is toward providing each design engineer with a dedicated workstation. Any or all mask levels can be displayed simultaneously, and rapid zooms and scrolls are generally available to provide both local and global information. Once the individual polygons or macros characterizing a mask feature are placed on the CRT, they are added to the geometrical database. After the layout in the CRT is in its final form, so is the database.

As an alternative to the engineer's part in layout, considerable research effort in the past 10 years has been devoted to automatic layout and routing programs. With this approach the computer will place the components and provide all appropriate interconnects (route) directly from the electrical description of the circuit. In addition to saving the layout time, layout errors can be eliminated with these programs. Programs that perform this task are available today and are quite useful for low-volume products requiring a fast turnaround where the area of the IC itself is not of major concern. Widespread use of automatic routing programs is particularly apparent with gate array products, although some of these programs get "stuck" in complicated regions and need human interaction to overcome these problems. Handpacked designs are generally denser than those obtainable with the automatic layout and routing programs. The challenges associated with the automatic layout and routing problem were recognized years ago from research on PC board layout.

Interactive layout and routing routines are also useful since the computer can rapidly do the drafting required in nonchallenging portions of the circuit, whereas the operator can often make better choices for placement or routing in tight situations. Although it may appear to the unwary that the development of such a program is straightforward, the challenge of such a program is attested to by the fact that many millions of dollars have been spent on the development of the routing and placement programs over the past decade, and research activity in this area is still intense.

Layout verification programs, such as Design Rule Checkers (DRCs), are useful for confirming that no design rule violations occur and for detecting some layout errors (since these errors often cause a design rule violation). For large circuits, considerable amounts of computer time are required for most sophisticated verification programs. Some layout editors incorporate the DRC algorithms into the layout program and do local design rule checks each time a feature is added to the database, thus ensuring that the database violates no design rule as it is created. One of the more popular programs in this class is MAGIC from the University of California at Berkeley. Automated schematic extraction or hand generation of a circuit schematic by someone other than the initial designer (who may be too familiar with the design and thus more likely to pass over an error again), followed by comparison with the original schematic, is often useful for detecting layout errors.

CHƯƠNG 3. DESIGN AUTOMATION AND VERIFICATION

INTRODUCTION

Design automation and design verification are the keys to effective use of large-scale integrated circuit technology today. When circuits consisted of only a few transistors or gates, layout and checking of circuits by hand were reasonable. As circuit complexity increased to thousands and tens of thousands of transistors, manual tools were no longer sufficient for design, causing computer-based design aids to become prominent. With present integrated circuits containing hundreds of thousands of transistors, heavy dependence on design automation and design verification is necessary to design these circuits.

This chapter describes the nature and use of basic design automation and design verification tools as applied to the design of integrated circuits. *Design automation tools* are defined here as those computer-based tools that assist through automation of procedures that would otherwise be performed manually, if at all. Simulation of proposed design functionality and synthesis of integrated circuit logic and layout are just two examples. *Design verification tools*, on the other hand, are those computer-based tools used to verify that circuit design or layout meets certain prescribed objectives. A geometrical design rule checker for examining layout characteristics is an example, and a logic simulator with a specific set of input vectors and corresponding desired output vectors is another. Note that simulation can be classified in either category according to its purpose. Both design automation tools and design verification tools are included in the more general class known as CAD (computer-aided design) tools.

Both design automation and design verification tools require computer-readable descriptions of the underlying circuit function and structure to operate. These computer-based descriptions vary from simple geometrical specification languages such as CIF[1] (Caltech Intermediate Form) to high-level functional description languages such as VHDL[2] (a hardware design language). Initially the focus of this chapter is on a description of design tools related to or based on geometrical layout. A simplified geometrical specification language will be examined. Required functionality provided by tools that input and display integrated circuit layout will also be described. Then, tools that check layout geometries and extract circuit net list information will be detailed.

Design tools for higher-level design description and verification are described next. Circuit, switch, and logic simulation for digital circuits are introduced and compared. Timing analysis is examined as a way to verify the temporal operation of digital circuits. Hardware design languages such as VHDL and EDIF[3] (Electronic Design Interchange Format) are introduced with simple examples provided to clarify important concepts.

The descriptions of design verification and design automation tools provided here use MOS examples primarily. The concepts are directly applicable to bipolar designs, although some changes in specific tool capability may be required by different technologies. The chapter concludes with an introduction to automated methods of generating layout from high-level descriptions of digital circuits via silicon compilers.

INTEGRATED CIRCUIT LAYOUT

Historically, integrated circuit design and integrated circuit layout functions were performed by separate groups. The circuit design task resulted in mixed logic and transistor-level circuit diagrams describing the intended circuits. A circuit description like that of Fig. 10.1-1 was given to layout artists, who were experts at converting circuit diagrams to geometrical layouts such as the one shown in Fig. 10.1-2. For early commercial products, the layout drawings were transferred to rubylith masks by hand. Later, layouts were drawn on vellum—a tough, semitransparent drafting material—to withstand the many design modifications

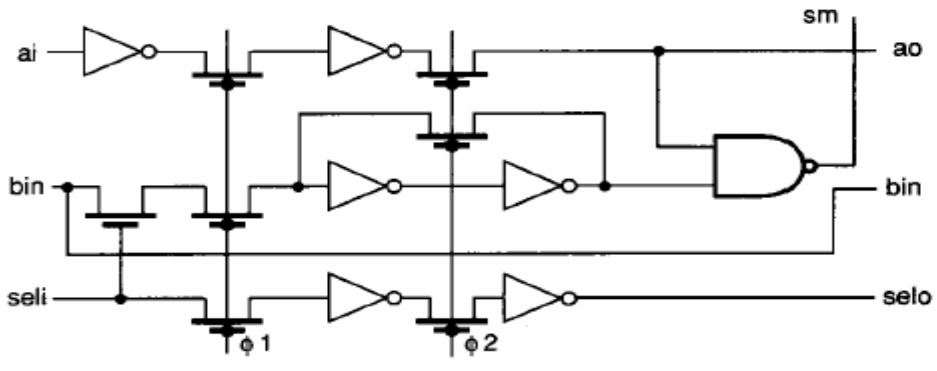


FIGURE 10.1-1
Partial circuit diagram for bit-serial multiplier of Fig. 10.1-2.

that are inherent in the normal design process. The layouts from the large vellum plots were digitized to computer-readable form to allow automated checks and to provide input to the mask-making process. Although this method worked for many years, including the early days of microprocessors, the large number of devices required in modern integrated circuits causes fully manual layout to be too time-consuming and prone to error. However, even today, critical sections of the newest microprocessors are still handcrafted to pack the circuit into the smallest possible area.

Many modern methods of integrated circuit layout include both synthesis of control logic and handcrafting of critical building blocks that will be repeated. These layout pieces are entered into a computer at an early stage to allow mechanized help with replicating, checking, and plotting the complete integrated circuit layout. Design layouts may be entered via tools that help convert graphic layout information to computer-readable form. An early tool, shown in Fig. 10.1-3, is called a *digitizer* and was used to enter layout coordinates directly into a computer from a layout plot. Sometimes layout is converted directly to text input in the form of a geometrical specification language. Most often, geometrical layout information is entered through a color graphics workstation to specify the desired integrated circuit layout.

Geometrical specification languages for integrated circuits allow computer-readable definition of the geometries for the mask layers required to fabricate an integrated circuit. These specification languages contain primitive structures such as wires and boxes to specify geometrical shapes and layout levels. Organizational constructs are also provided to allow placement and repetition of the geometrical structures. A geometrical specification language is much like a computer programming language, with the geometrical shape primitives corresponding to instructions and the organizational constructs corresponding to procedures with parameter values.

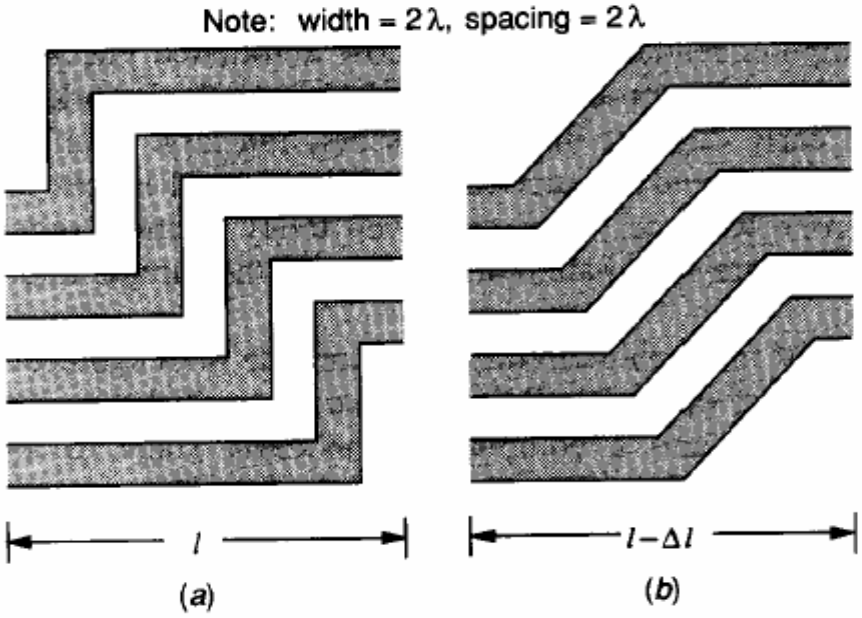


FIGURE 10.1-4
Layout Styles: (*a*) Manhattan, (*b*) Diagonal.

A simplified geometrical specification language for Manhattan style designs for a general MOS process is used here to illustrate relevant concepts. A *Manhattan design style* is one that supports only horizontal and vertical geometries. The name arises because Manhattan style layouts resemble an aerial view of the street layout of New York's Manhattan borough. This style precludes diagonal structures, such as interconnection jogs, that are sometimes used within circuit layouts to minimize area. Figure 10.1-4 shows layout styles with and without diagonal structures. The potential area savings with diagonal structures must be weighed against the increased complexity of programs used to verify the final design. Many commercial integrated circuit manufacturers allow diagonal layout structures but limit these to 45° angles from horizontal and vertical structures.

The simplified geometrical specification language defined here provides only two primitive statements. The two primitives are *boxes* and *levels*, while the organizational constructs include *macros* and *calls*. A macro is like a high-level language (HLL) procedure, and a call is like an HLL procedure call. Table 10.1-1 provides the syntax for these primitives and organizational constructs.

All parameter values are integers. Lengths are in terms of λ, a measure related to the characteristic resolution of the process and the layout design rule set. Macro numbers, layout levels, and orientations are limited to positive integers. A minimum set of layers for a typical NMOS n-well CMOS process is defined in Table 10.1-2. Appendices 2A and 2B define corresponding layers for a double polysilicon NMOS and a p-well CMOS process, respectively.

TABLE 10.1-1
Simplified geometrical specification language

B *x y dx dy*	Box structure with length *dx*, width *dy*, and lower left-hand corner placed at *x*, *y*
L *n*	Layout level for the box definitions that follow
M *n*	Start of macro number *n*
E	End of a macro
C *n x y m*	Call for macro number *n* with translation *x*, *y* and orientation *m*
Q	End of layout file

MOS layer definitions

Layer	CMOS	NMOS
1	n-diffusion	n-diffusion
2	p-diffusion	Ion implant
3	Polysilicon	Polysilicon
4	Metal	Metal
5	Contact	Contact
8	n-well	—
9	Overglass	Overglass

The orientation represents possible rotations of the geometrical figure after translation. The relative order of translation and rotation is important (see Prob. 10.3). Here, rotation is performed first with translation following. The possible orientations are defined in Table 10.1-3 and demonstrated with the block letter P in Fig. 10.1-5.

This simple geometrical specification language will suffice to specify any MOS Manhattan integrated circuit layout if the necessary layout levels are defined. The description is based on alphanumeric characters and is easily displayed, edited, or transferred between computer systems.

TABLE 10.1-3
Rotations of geometries

Orientation	Description
1	No rotation
2	Rotate 90° CCW
3	Rotate 180° CCW
4	Rotate 270° CCW
5	Mirror about y-axis
6	Rotate 90° CCW and mirror about y-axis
7	Rotate 180° CCW and mirror about y-axis
8	Rotate 270° CCW and mirror about y-axis

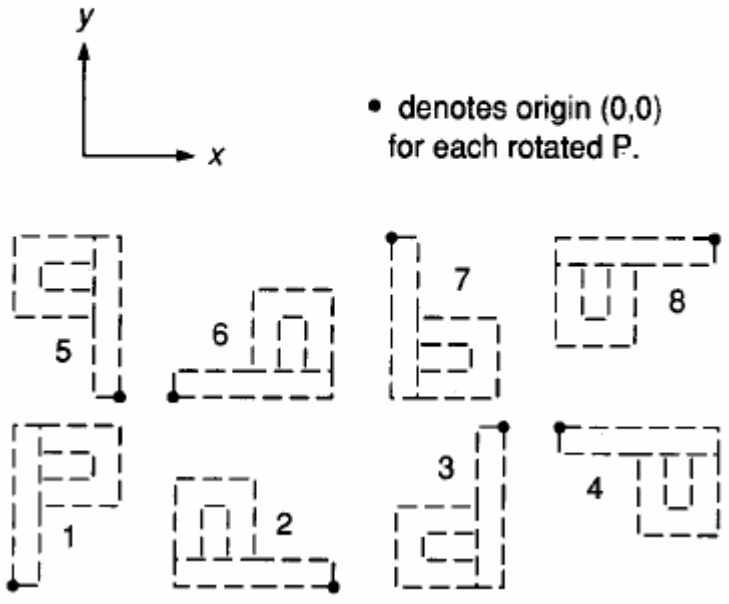
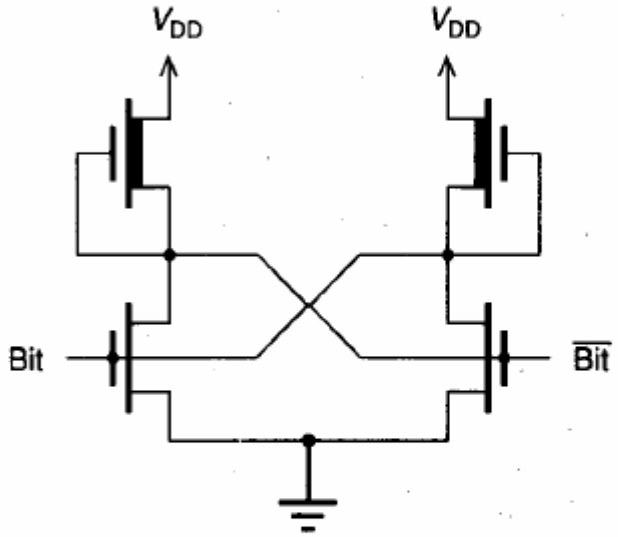


FIGURE 10.1-5
Cell orientations.

```
M  5
L  1
B  11  0  4  5
B  5  3  6  2
B  1  3  4  7
B  1  10  2  3
B  1  13  14  2
L  2
B  4  1  8  6
L  3
B  6  1  4  8
B  8  6  4  6
B  10  12  2  7
L  4
B  0  0  16  2
B  1  6  11  4
B  0  13  16  2
L  5
B  12  0  2  1
B  2  7  2  2
B  9  7  2  2
B  7  14  2  1
E
M  8
C  5  0  0  1
C  5  16  30  3
E
C  8  0  0  1
Q
```

(*a*)



(*b*)

FIGURE 10.1-6
Static memory cell definition: (*a*) Geometrical specification file, (*b*) Circuit diagram.

An example of the geometrical specification file for a static memory cell composed of two inverters tied back to back is shown in Fig. 10.1-6 along with the corresponding circuit diagram. A single inverter consisting of an enhancement pulldown transistor and a depletion pullup transistor is defined by macro 5. This inverter is placed twice, once in a rotated and translated position, to create the static memory cell defined as macro 8. Macro 8 is placed once to create the layout plot shown in Fig. 10.1-7.

Layout Styles

In spite of high labor costs, handcrafted layout is still used within the semiconductor industry because of the necessity to minimize the area required by high-volume integrated circuits. Even automated layout methods such as silicon compilation and standard cell synthesis use handcrafted layout to optimize the primitive cells that are combined through automated techniques. Frequently the basic form for the integrated circuit is sketched and optimized on paper prior to entry into

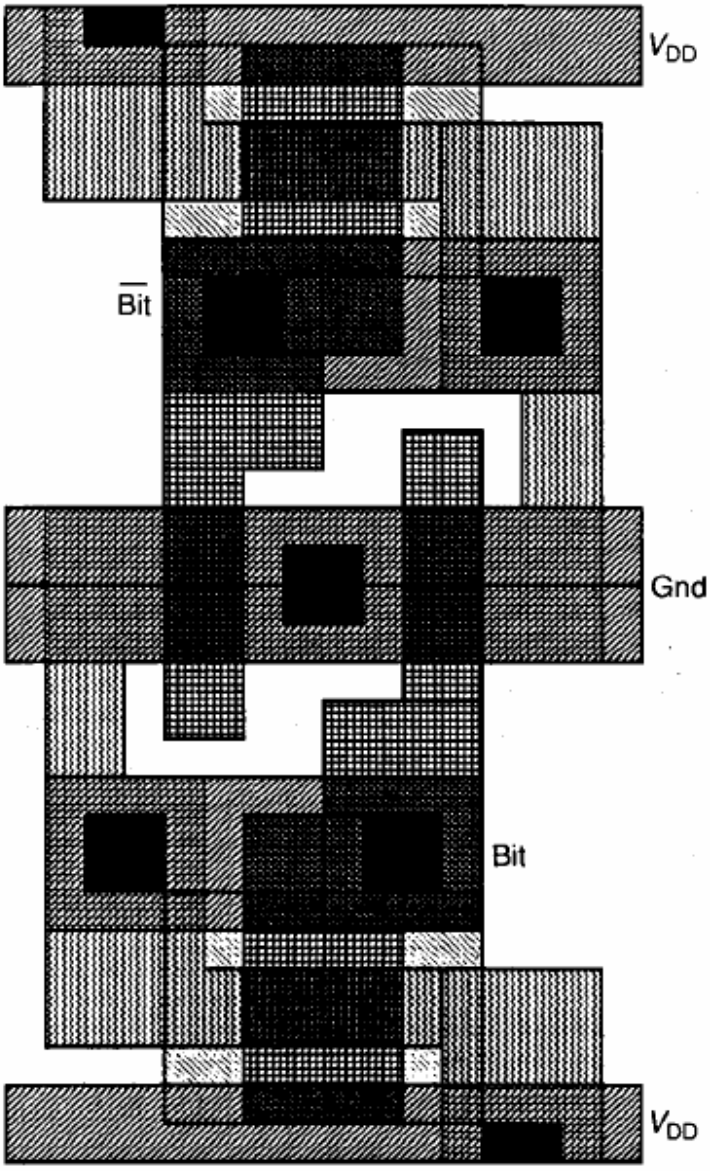


FIGURE 10.1-7
Static memory cell layout.

a computer. The resulting geometrical layouts are digitized, sometimes through use of a symbolic layout language but primarily with the help of an interactive CRT graphics editor.

Handcrafted layouts can be entered directly into a computer in geometrical form through use of an interactive CRT graphics editor. A mouse or joystick is used in conjunction with a cursor to size and position geometrical objects such as boxes on a high-resolution CRT display. A corresponding data file is kept in computer memory to describe the displayed geometries. With an operator's command, this data file may be converted to a geometrical specification language description or can be saved for further use. Several advantages of this graphical editor accrue from bypassing the need to input numerical data to a computer with a text editor or digitizer and from the ease with which geometries can be changed or duplicated.

The graphics editor called Magic,[4] currently popular with universities, uses the painting idiom to create geometrical objects on a color CRT display. The user chooses a color (layout level) from a palette on the screen and paints areas on the screen by specifying two opposite corners of a rectangular field.

The chosen color fills the area. The final result is the same as for other layout methods: a geometrical specification file is created in computer memory, saved, and ultimately transferred to the mask shop.

Graphical editors in both industrial and university environments typically maintain their own unique memory and disk representations of layout geometries. For reasons of efficiency (fast editing response and minimum memory requirements), these representations are often highly optimized binary data structures. In the university environment, the geometrical specification language CIF was defined as a common interchange format among universities and between universities and the MOSIS fabrication service. In industry, EDIF was defined as the interchange format. Most industrial CAD tools provide conversions between their internal format and EDIF. In addition, most industrial CAD tools convert their internal format to a special binary format for submittal to the mask shop. The Berkeley Oct tool set[5] provides conversion from its internal format (Oct) to and from both CIF and EDIF.

In summary, both specification of layout geometries and designer entry of layout geometries are described in this section. Many different geometrical specification languages have been defined and used. A very simple one was defined here for demonstration purposes only. In the university environment, CIF is the predominant interchange format, and EDIF is the interchange standard in industry.

SYMBOLIC CIRCUIT REPRESENTATION

Descriptions of integrated circuit layouts can take many forms. The geometrical specification language of the previous section provides a primitive textual description of a circuit. Other, more symbolic, forms of representation are often used by designers to specify layouts. A hierarchy of these, including a parameterized layout representation, parameterized module generation, a graphical symbolic representation, and logic equations, is described here.

Parameterized Layout Representation

A symbolic layout language[1] (SLL) allows a textual description of circuit layout in a form that is more easily generated and understood by humans than the geometrical specification language of the previous section. In the past, an SLL was used to represent design layouts that were drawn by hand on graph paper and then digitized. Two main characteristics differentiate an SLL from the geometrical specification language described previously. First, the SLL uses descriptive identifiers for the parameters necessary to specify a geometrical layout. Examples are BOX, POLY, and DX for the geometrical shape, the layer, and the width in the x direction, respectively. This provides a readable description of geometries that specify a layout. Thus, the SLL description is easily entered into a computer using the designer's favorite text editor. Second, symbolic entries are allowed in addition to the numerical data of the geometrical specification language. For example, the x and y position of a geometry might be specified by the variables XPOINT and YPOINT. This allows the final placement of the geometry to depend on the

placement of other cells. At some point in the design process, the SLL must be converted to a geometrical specification language for use by other CAD tools and for transmittal to the mask shop. XPOINT and YPOINT must be assigned numerical values to specify the location of the geometry before this conversion takes place.

In addition to the use of symbolic parameters in an SLL, programming constructs such as loops and conditionals can provide additional capability in the specification of a cell's layout. The use of an SLL to describe layout is much like the use of assembly language to describe the machine language (binary) program for a computer. An assembly language program uses mnemonics for the instructions and symbols for variables to simplify and expedite the process of programming a digital computer. Both forms describe the same end object; the binary representation provides the most concise description, while the assembly language is a preferable working medium for programmers.

An SLL description for the layout of the CMOS inverter of Fig. 7.5-5 is given in Fig. 10.2-1. Note the verbose nature of this description compared to the geometrical specification file of Fig. 10.1-6. The description of Fig. 10.2-2 demonstrates the use of variables to allow the inverter cell of Fig. 7.5-5 to be stretched in either the VERT (vertical) or HORZ (horizontal) directions. Also, a REPEAT statement is included to allow the cell to be repeated NR times. RX and RY are the repeat distances along the x and y axes, respectively. If the variables VERT and HORZ are each set to a value of 0 and NR is set to 4, the inverter cascade of Fig. 10.2-3 is produced. The two variables VERT and HORZ can be used to stretch the inverter cell to match the pitch of adjacent cells by

```
CELLNAME CMOSINV;
BOX NDIF X=3 Y=0 DX=4 DY=4;
BOX NDIF X=3 Y=4 DX=2 DY=4;
BOX NDIF X=3 Y=8 DX=4 DY=4;
BOX PDIF X=3 Y=20 DX=4 DY=4;
BOX PDIF X=3 Y=24 DX=5 DY=4;
BOX PDIF X=3 Y=28 DX=4 DY=4;
BOX POLY X=0 Y=5 DX=7 DY=2;
BOX POLY X=0 Y=7 DX=2 DY=18;
BOX POLY X=0 Y=25 DX=10 DY=2;
BOX POLY X=4 Y=14 DX=8 DY=4;
BOX MET1 X=0 Y=0 DX=12 DY=4;
BOX MET1 X=0 Y=28 DX=12 DY=4;
BOX MET1 X=3 Y=8 DX=4 DY=16;
BOX MET1 X=7 Y=14 DX=1 DY=4;
BOX CONT X=4 Y=1 DX=2 DY=2;
BOX CONT X=4 Y=9 DX=2 DY=2;
BOX CONT X=5 Y=15 DX=2 DY=2;
BOX CONT X=4 Y=29 DX=2 DY=2;
BOX CONT X=4 Y=21 DX=2 DY=2;
BOX NWEL X=0 Y=18 DX=12 DY=16;
END CMOSINV;
```

Figure 10.2-1
Symbolic layout language description of CMOS inverter of Fig. 7.5-5

```
CELLNAME CMOSINV;
BOX NDIF X=3 Y=0 DX=4 DY=4;
BOX NDIF X=3 Y=4 DX=2 DY=4;
BOX NDIF X=3 Y=8 DX=4 DY=4;
BOX PDIF X=3 Y=20 DX=4 DY=4;
BOX PDIF X=3 Y=24 DX=5 DY=4;
BOX PDIF X=3 Y=28 DX=4 DY=4+VERT;
BOX POLY X=0 Y=5 DX=7 DY=2;
BOX POLY X=0 Y=7 DX=2 DY=18;
BOX POLY X=0 Y=25 DX=10 DY=2;
BOX POLY X=4 Y=14 DX=8+HORZ DY=4;
BOX MET1 X=0 Y=0 DX=12+HORZ DY=4;
BOX MET1 X=0 Y=28+VERT DX=12+HORZ DY=4;
BOX MET1 X=3 Y=8 DX=4 DY=16;
BOX MET1 X=7 Y=14 DX=1 DY=4;
BOX CONT X=4 Y=1 DX=2 DY=2;
BOX CONT X=4 Y=9 DX=2 DY=2;
BOX CONT X=5 Y=15 DX=2 DY=2;
BOX CONT X=4 Y=29+VERT DX=2 DY=2;
BOX CONT X=4 Y=21 DX=2 DY=2;
BOX NWEL X=0 Y=18 DX=12 DY=16+VERT;
END CMOSINV;
CELLNAME FOURINV;
REPEAT CMOSINV NR=4 RX=12+HORZ RY=0;
END FOURINV;
```

FIGURE 10.2-2
Parameterized symbolic layout language description for inverter cascade of Fig. 10.2-3

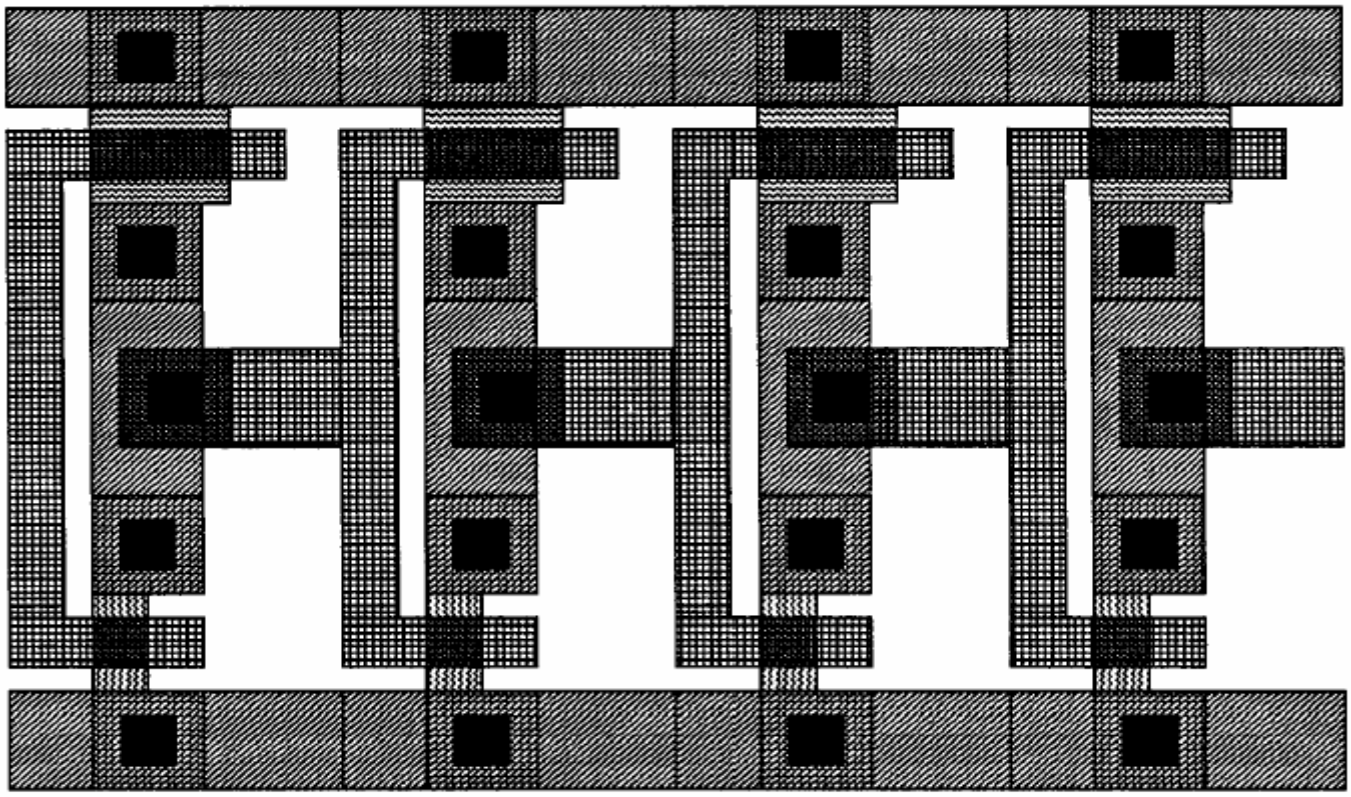


FIGURE 10.2-3
Inverter cascade created from parameterized symbolic layout language description.

specifying positive values for one or both of VERT and HORZ. The use of a programmatic description of layout greatly expands the capabilities of a layout designer in specifying the geometrical structure of a circuit.

Parameterized Module Generation

A recent advance in the area of symbolic layout descriptions is the use of parameterized module generators. A parameterized module generator is a software procedure that can generate many different cell layouts depending on values that are specified when the generator program is executed. Parameterized module generators have been written for RAMs, ROMs, PLAs, multipliers, adders, and data paths, for example. Many of these generators use input parameters to specify the width or number of bits in the generated layout.

As an example, three separate designs might require an 8-bit adder for the first design, a 16-bit adder for the second design, and a 32-bit adder for the third design. Typical design style would use an interactive graphics editor to create each of these adders separately. If a parameterized generator for the adder module could be defined, however, a single module generator could be used to produce an N-bit adder where N is a parameter that can be set to 8, 16, 32, or some other integer value. Then each of the three adders could be created from the same parameterized description. A parameterized module generator is particularly well suited to modern integrated circuit design styles, which commonly utilize regular structures such as rows of cells and arrays of cells.

Parameterized module generators use many of the powerful constructs of high-level programming languages to describe layout structure, position subcells, and fit the overall layout of a larger cell together. Parameterized variables are used with their values bound to a specific value when a module is generated. Conditional statements allow creation of specialized edge cells and programming of memory and PLA contents. For example, a parameterized module generator for an array of cells might include conditional statements such that if both the x and y indices were equal to 0, then an upper-left corner cell would be generated. If the x and y indices were each between the smallest and largest values, a center cell would be generated, and so forth.

The use of high-level programming language techniques also provides a disadvantage for many parameterized module generators. That is, the layout cannot be visualized until the generation program has been compiled and linked to instantiate the layout for a module. These potentially time-consuming steps may hinder the use of interactive layout in designing a module generator for a new cell. To circumvent this problem, there is ongoing research on ways to provide interactive graphical feedback as the geometrical structure of a cell is defined.[6]

With such a tool, a silicon layout specialist can create the parameterized modules that are required in a design. Then a circuit or logic designer can use these blocks by specifying parameters appropriate to the design task. Recently, parameterized module generators were used to specify the layout of a commercial RISC processor (see Sec. 10.11). An interesting, but unsolved problem, is to prove that the output of a parameterized module generator is correct over the valid range of parameters for the module generator.

Graphical Symbolic Layout

The parameterized layout representation described previously provides little insight into the geometrical relationships between circuit elements. This important insight can be provided by another symbolic form for integrated circuit description, called graphical symbolic layout. An early form of graphical symbolic layout is called Sticks.[7] Sticks and related symbolic methods provide an abbreviated, graphical description that combines circuit connectivity with layout topology information. In the Sticks symbology, circuit connections are shown with colored (or weighted) lines representing layout levels, while transistors are formed by the intersection of the lines representing polysilicon and diffusion. The entire layout diagram is composed of simple line symbols that show both connectivity and topology but not actual or relative size for geometrical constructions.

The combination of connectivity and topological information is important in the generation of integrated circuit layouts, as is shown with the aid of the circuit diagram for the quasi-static memory cell of Fig. 10.2-4*a*. This circuit diagram shows a forward path from the first inverter to the second inverter and a clocked feedback path from the second inverter to the input of the first inverter. The circuit diagram does not indicate topological requirements to realize this path.

The geometrical layout of the memory cell of Fig. 10.2-4*a* requires decisions on changes of layout levels to prevent unwanted transistors and connections. The Sticks diagram of Fig. 10.2-4*b* retains all the circuit connectivity information

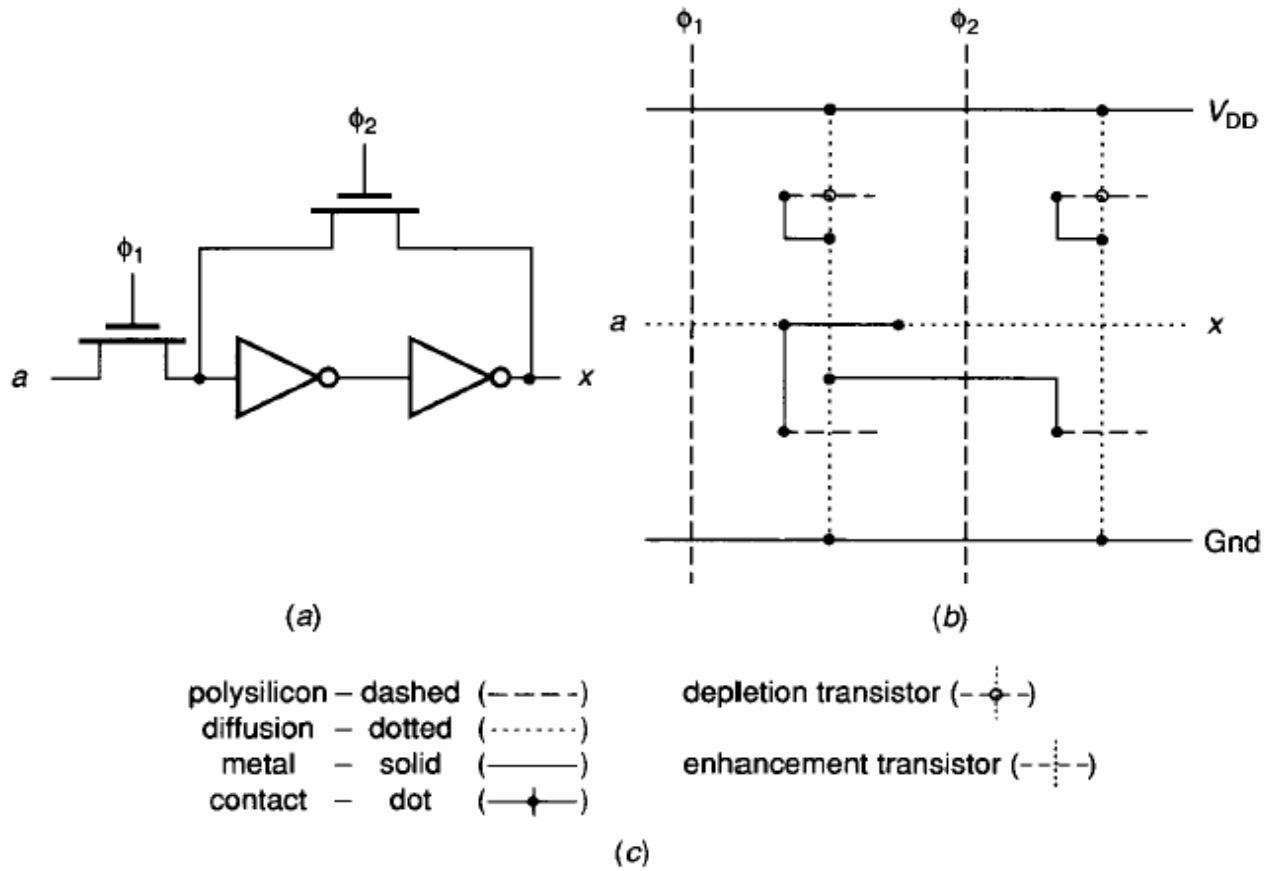


FIGURE 10.2-4
Quasi-static memory cell: (*a*) Logic diagram, (*b*) Symbolic diagram for NMOS layout, (*c*) Layer legend.

for the memory cell and also symbolically specifies the topology of the final integrated circuit layout. In particular, it shows that the feedforward path must be changed from the diffusion layer to the metal layer to cross the polysilicon ϕ_2 clock line without creating an unwanted transistor. Also, the Sticks diagram shows that the feedback transistor is conveniently formed by allowing the polysilicon ϕ_2 line to cross the diffusion feedback path. The diagram shows that power and ground are provided in metal and that the input and output signals are both in the diffusion level. The utility of Sticks and other graphical symbolic layout methods is derived from the simple abstracted notation for layout topology and circuit description.

Once a graphical symbolic layout for a circuit is generated, it is often simple for a designer to convert to a full layout form. The layout task has been simplified to the process of fattening connection lines and compacting the layout, especially if required transistor length-to-width ratios have been noted on the graphical symbolic layout. In fact, this process is simple enough to be automated.[8] If the graphical symbolic layout description has been entered into a computer, perhaps through an interactive graphics terminal, a symbolic compiler program can convert the symbolic layout to a full layout by expanding the line symbols according to a technology specification and then compacting the resulting layout.

As with most automated layout aids, a symbolic compiler usually trades reduced designer efforts for increased silicon area. An increase in the area for a layout generated with a program is not uncommon when compared to a handcrafted layout. As a result, high-volume integrated circuits such as microprocessors and memory continue to utilize handcrafted layout of replicated cells as a major design component. This does not, however, minimize the value of the symbolic representation to the designer. Capturing layout topology in symbolic form early in the layout design prevents later problems such as isolation of a circuit from direct metal connection to power buses.

Logic Equation Symbology

If the function of a digital integrated circuit can be captured by a set of Boolean logic equations, these equations suffice to generate an integrated circuit layout. Thus, logic equations represent a fourth symbolic means to describe a combinational logic circuit. One frequently used means to convert logic equations into layout topology is with a PLA generator, as described in Chapter 9. Two other methods for generating geometrical layouts from logic equations are discussed next: the Weinberger array[9] and SLAP[10] (a methodology for silicon layout).

A Weinberger array uses a regular structure of NOR gates to implement combinational logic in an integrated circuit form. This array structure was introduced in Chapter 9. Figure 10.2-5 shows a Weinberger array used to implement the full adder carry function described by

$$K = AB + AC + BC \tag{10.2-1}$$

Since the final structure is regular, it is not difficult to construct a computer program to generate the array layout using logic equations as program input. By

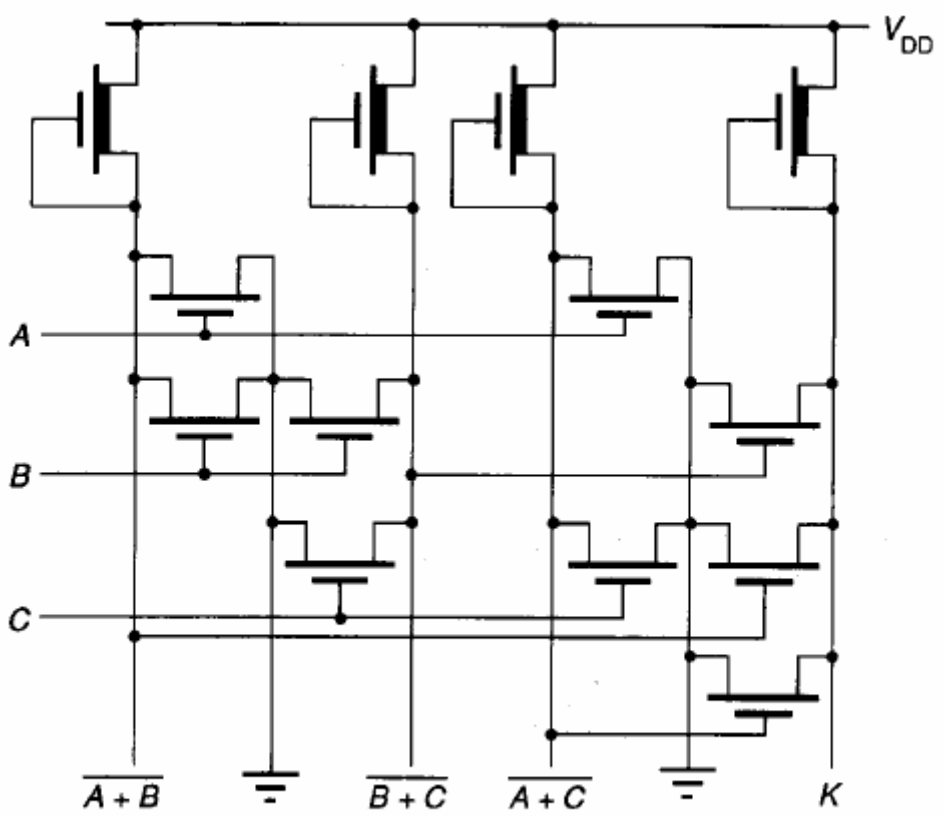


FIGURE 10.2-5
Weinberger array for full-adder carry.

use of DeMorgan's theorem, any combinational logic function can be realized using only NOR gates. In fact, the Weinberger array requires at most a series path of three NOR gates between an input and an output to realize a combinational logic function. Remember that a single-input NOR gate is an inverter. Thus, a first NOR gate may be required to provide the complement of an input while the final two levels use the NOR-NOR logic form to realize the logic function in product-of-sums form.

The use of NOR gates for a Weinberger array allows a constant size for the pullup devices even though the number of inputs and their corresponding pulldown devices may differ for each gate. Careful design allows adjacent gates to share a single ground path, as shown in the layout of Fig. 10.2-6. This array structure can be easily expanded by adding input variables at the bottom and NOR gates to the right without changing the existing structure.

A comparison of the Weinberger array with the PLA yields an interesting result. Even though the logic of a PLA is realized entirely with NOR gates, the AND-OR logic form corresponding to a sum-of-products description is normally used. The AND-OR logic form can be realized with NOR gates only by inverting both the inputs and outputs. This requires a series string of four or five NOR gates between the PLA inputs and outputs, thus causing more delay for a PLA implementation of logic than for a Weinberger array implementation which requires only three levels of logic.

In contrast to the PLA and the Weinberger array, both with predefined array structures, a third method called SLAP has been proposed to compile logic equations into layout form. SLAP first converts logic equations into a directed graph with a graph level for each level of the logic equations. If double-rail inputs

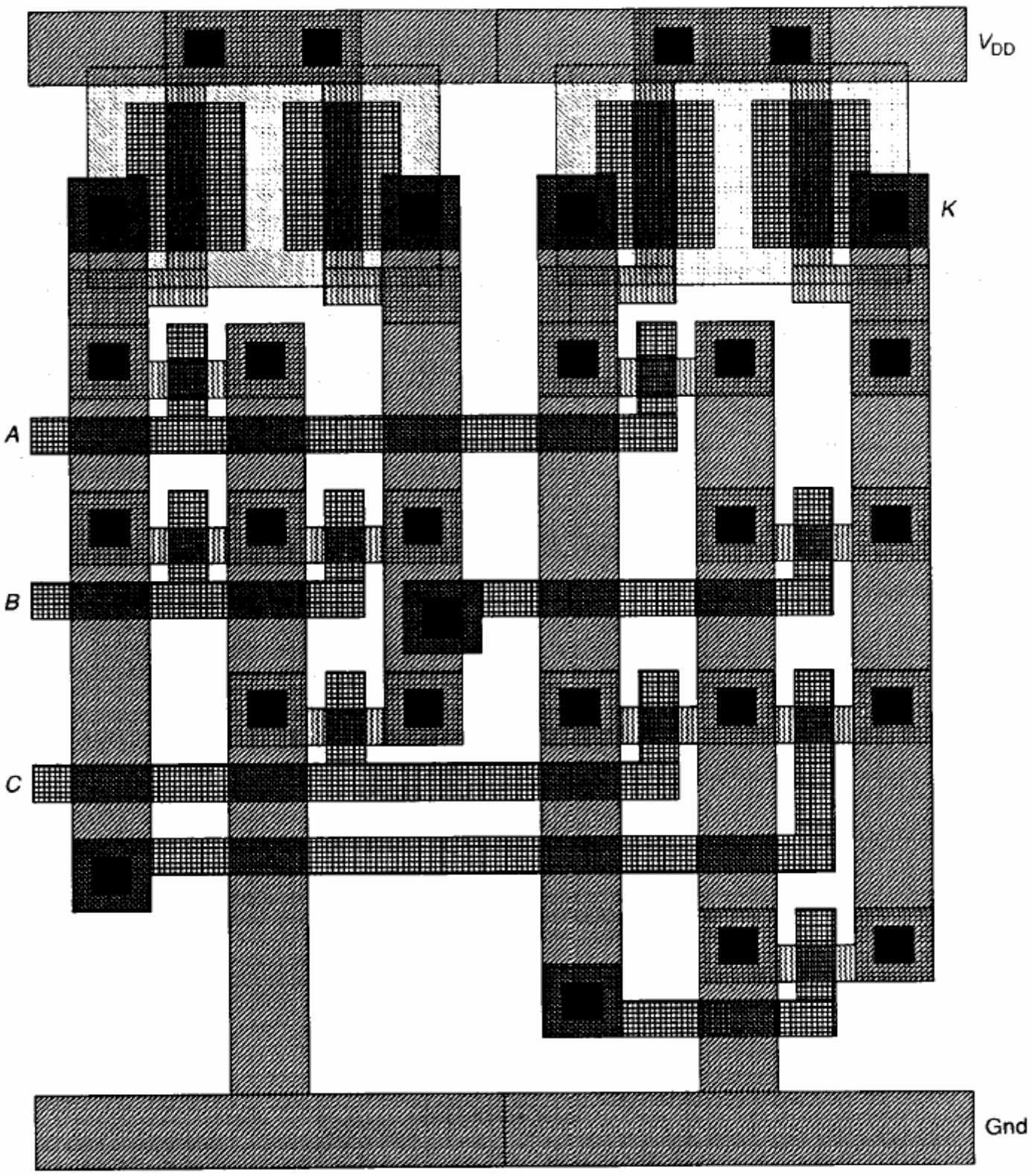


FIGURE 10.2-6
Layout for Weinberger array.

are available, at least two levels of gates are required to implement a general logic function. The SLAP methodology, however, allows realization of intermediate outputs that may then be used as inputs for other logic functions. A graph with an arbitrary number of levels may be required, depending on the particular representation for the logic. Figure 10.2-7 shows the directed graph for the logic functions of the following equations.

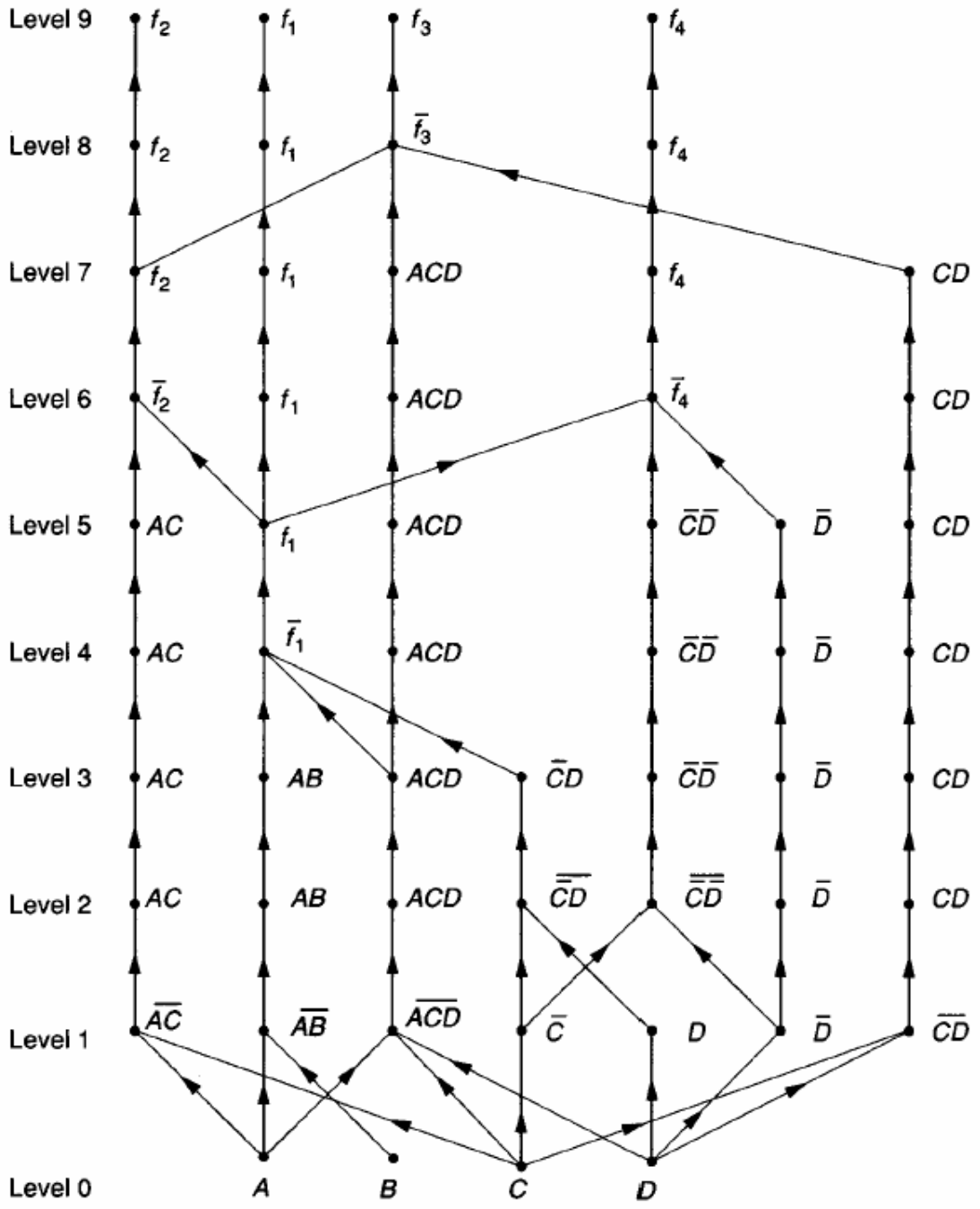


FIGURE 10.2-7
Directed graph for Eqs. 10.2-2 through 10.2-5.

$$f_1 = AB + \overline{C}D + ACD \tag{10.2-2}$$

$$f_2 = AC + f1 \tag{10.2-3}$$

$$f_3 = ACD + CD + f_2 \tag{10.2-4}$$

$$f_4 = \overline{C}\,\overline{D} + \overline{D} + f_1 \tag{10.2-5}$$

This directed graph is formed by placing logic gates with external inputs at the first level, secondary logic gates at the next level, and so on. Heuristics are then used to improve the organization by reducing the number of required levels, if possible, and to reduce the resulting layout area required. The layout density achieved with this method is about the same as that accomplished with gate array

structures. An important characteristic of the SLAP methodology is that general logic structures can be compiled directly into a geometrical layout, whereas the PLA format forces a two-level logic realization.

In this section, four methods of generating layout from symbolic representations were introduced. Of the first two, parameterized layout representation and parameterized module generation, the second is growing in popularity for layout of today's designs. Graphical symbolic layout also enjoys success as a technique for layout of random logic. Synthesis of layout directly from the fourth symbolic form, logic equations, is fast becoming a widely used technique for generating integrated circuit layout.

COMPUTER CHECK PLOTS

Generation of a layout plot from a geometrical specification file for an integrated circuit is often desirable. In the past, large-scale plots, some almost big enough to cover one end of a basketball court, were generated so that visual checking of circuit layout could be performed. Most of these visual checks can now be performed directly from a computer-based geometrical specification file without manual intervention. A computer program can verify fixed rules for the millions of geometrical figures used to describe VLSI circuits without tiring and without error—a task that is essentially impossible for humans. However, human capability to critique overall structure or to detect inconsistencies in an otherwise regular design is difficult to duplicate with computer-based checks. As a result, hardcopy plots of integrated circuit designs are still used for finding errors, for promotional literature, and for many other purposes. Such plots are called *computer check plots*.

Computer check plots for integrated circuit designs are created in both soft- and hardcopy form on CRTs, printers, and plotters using color or black on white representations for the layout artifacts. Check plot devices range from monochrome CRTs, with only 24 × 80 character resolution for the entire display, to laser printers with 300 dots per inch or higher resolution. To compare the maximum usable display capability over this range of resolution, an example using a static memory cell is examined next.

The static memory cell of Fig. 10.1-7 has dimensions of 16 $\lambda \times 30\ \lambda$ for an area of 480 λ^2. A monochrome alphanumeric CRT using character graphics with 24 lines by 80 columns can display an area of 1920 λ^2, although the effective area is somewhat less because of the 1:3 aspect ratio of the CRT display resolution. All details of the static memory cell are visible in the CRT display, as shown in the hardcopy plot of Fig. 10.3-1, but the cell's relation to other cells is lost. As a second example, a dot matrix drawing normally requires a resolution of at least 5 dots per λ to define the smallest details of a circuit. For a printer with a resolution of 100 dots per inch, the static memory cell requires a plot that is about 0.8 × 1.5 in. to show the details of the circuit. Figure 10.3-2 provides a plot of this size for the memory cell of Fig. 10.1-7. If the memory cell were part of a 1K-bit memory (32 cells × 32 cells), a high-resolution plot of the entire memory array would require 25.6 × 48 in. Of course, the general form of the memory area could be discerned with a much smaller plot. Figure 10.3-3 shows a plot at one-tenth this scale (2.56 × 4.8 in.) for the 32-by-32 cell array.

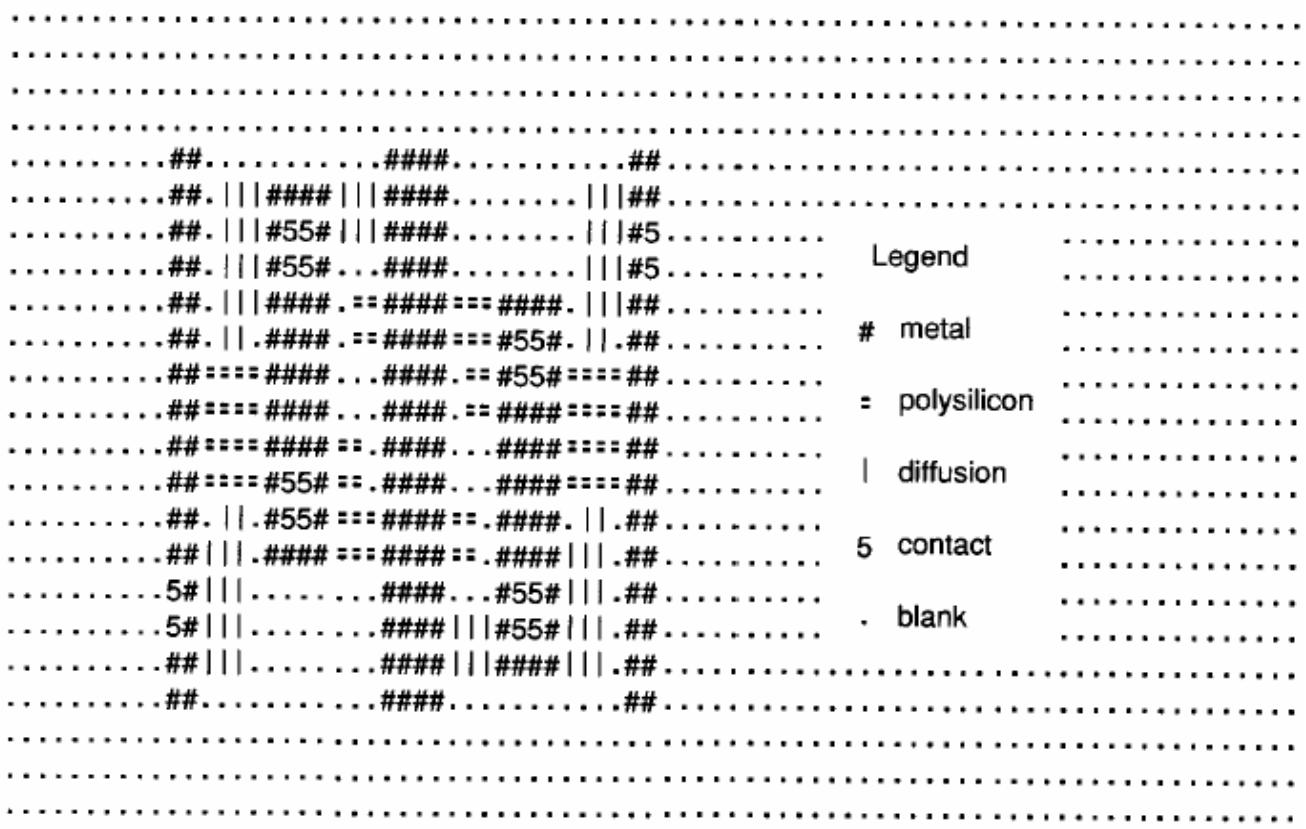


FIGURE 10.3-1
Hardcopy plot of SRAM cell as displayed on 24 line by 80 character CRT (λ = one character width).

A typical graphics CRT display with a 19 in. diagonal screen (15 in. horizontal $\times$ 11 in. vertical) might have a resolution of 760 by 480 dots. This is roughly 50 dots per display inch. Based on the analysis above, the details of a 152 λ by 96 λ circuit could be displayed in its entirety on the screen. This would correspond to about a five-by-six array of the memory cells described above. Figure 10.3-4 shows a hardcopy plot of the memory cells that could be seen on the CRT display. Of course, an entire chip can be displayed if the layout is scaled so that the finer details of the chip are lost. Figure 10.3-5 shows the entire layout for a 220 $\times$ 230 mil image-processing chip composed of sixteen, 12-bit serial multipliers with associated circuitry and input/output pads.

Color displays and plots are always a higher-cost feature than black and white; where color is available, each integrated circuit layer is represented using

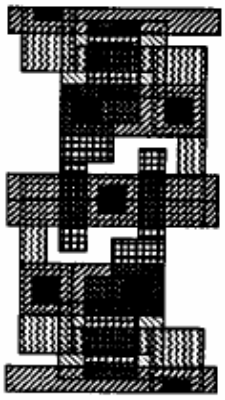


FIGURE 10.3-2
Minimum size plot for Fig. 10.1-7 with 100 dots/inch resolution.

a different color. Aside from their aesthetic appeal, color renditions of circuits show higher information content per unit area, allowing display of larger circuits in a given area. For a color display, only 2 to 3 dots per λ of resolution are necessary to delineate circuit details. Additionally, individual color levels can be used to show labels, flag geometrical design rule violations, or highlight specific features of a circuit. Most modern graphics workstations provide color displays.

When black on white plots are generated, two primary methods are used to distinguish individual layers. Line drawings, with each layer represented by a different style of line (solid, dotted, dashed, dot-dash, etc.) are producible on almost any printer with dot graphics capability (see Fig. 10.3-6). Filled drawings with different layers shown by characteristic area fill pattern (fine dots, heavy dots, diagonal lines, vertical lines, etc.) are popular, even at the expense of increased computer time to generate the plots, greater wear on the printer mechanism, and longer time to print the plots. Laser printers provide good resolution (300 dots per inch) and are frequently used for area fill check plots. A primary advantage of the filled drawing of Fig. 10.1-7, compared with the line drawing of Fig. 10.3-6, is that the concept of area for integrated circuit layers is quickly conveyed to the viewer by the filled drawing. This concept is important to the designer since the fabrication process operates on contiguous areas rather than the individual boxes used to describe them.

In this section, a short summary of integrated circuit display media and their corresponding resolution requirements was presented. It is important to have high-resolution display and hardcopy capability for integrated circuit layout design.

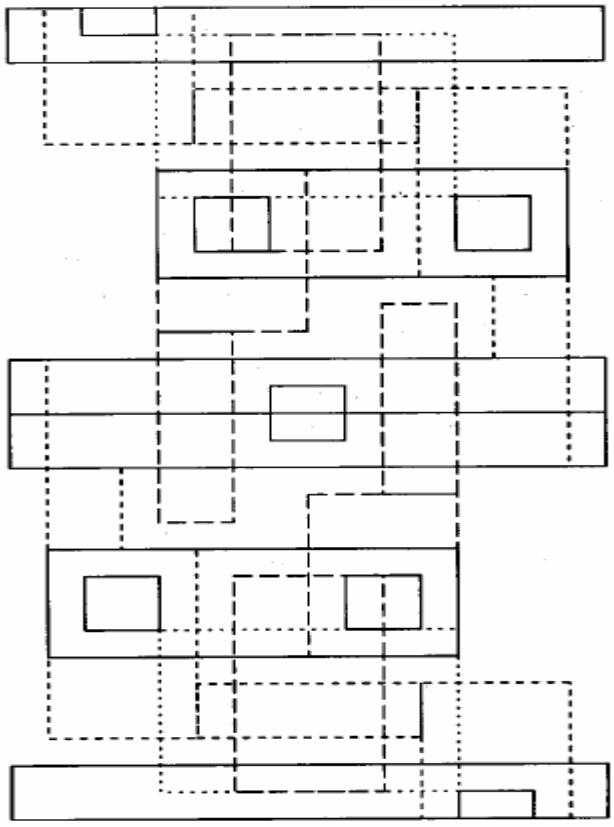


FIGURE 10.3-6
Line check plot of layout of Fig. 10.1-7.

DESIGN RULE CHECKS

Integrated circuits are created from several layers whose geometrical structures are defined by photolithographic masks. At any given time, a minimum resolution exists for the structures that can be fabricated on silicon because of lithographic and processing constraints. Any attempt to define structures that require higher resolution or accidental specification of a higher resolution through carelessness may lead to nonfunctional circuits. Also, violation of certain geometrical relationships among layers may cause failures because of processing constraints. For each process, a set of guidelines called *design rules* is specified to encapsulate geometrical fabrication constraints. The design rules for the CMOS process described in Table 2B of Appendix 2 are used as the basis for the following discussion. However, most of the rules are determined by general lithographic and processing constraints so that similar rules apply to other processes as well.

Geometrical Design Rules

A conceptual explanation of geometrical design rules is provided in this section. Design rules were introduced in Sec. 2.3 of this text. Geometrical design rules for a single integrated circuit layer are simple; they involve only spacings and widths. Figure 10.4-1 demonstrates a 2 λ spacing between polysilicon conductors

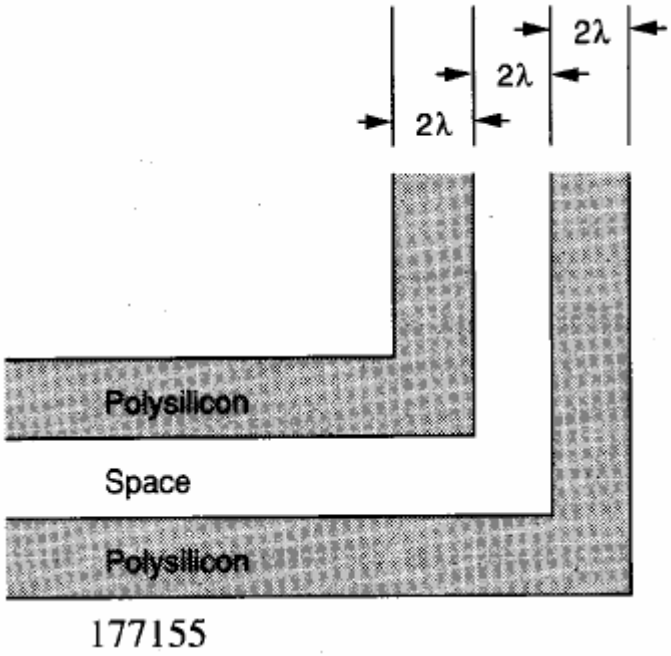


FIGURE 10.4-1
Minimum width polysilicon conductors.

that are each 2λ wide. It is worth noting that if a mask layer is complemented, all widths become spacings. This is shown in Fig. 10.4-2, where the complemented polysilicon conductor widths from Fig. 10.4-1 appear as spacings. Therefore, if width is considered in terms of the complement of the layer definition, all single-layer rules can be treated as simple spacing rules. This means that the same computer algorithm can be used to check for both width and spacing errors.

An interesting conceptual understanding of design rules was provided by Lyon.[11] His explanation is based on the scalable parameter λ, which is said to describe the minimum resolution of the fabrication process. In practice, fabrication processes are usually characterized by their minimum transistor length. The parameter λ is normally specified as half the minimum transistor length. Thus, a 2 μ process has a minimum gate length (and width) of 2 μ, and λ would be set to 1 μ. Thus, λ is not directly a measure of process resolution, but rather is proportional to the minimum device length. With this in mind, the following two meta rules (a meta rule is a rule about rules) were proposed by Lyon to generalize geometrical design rules in terms of λ.

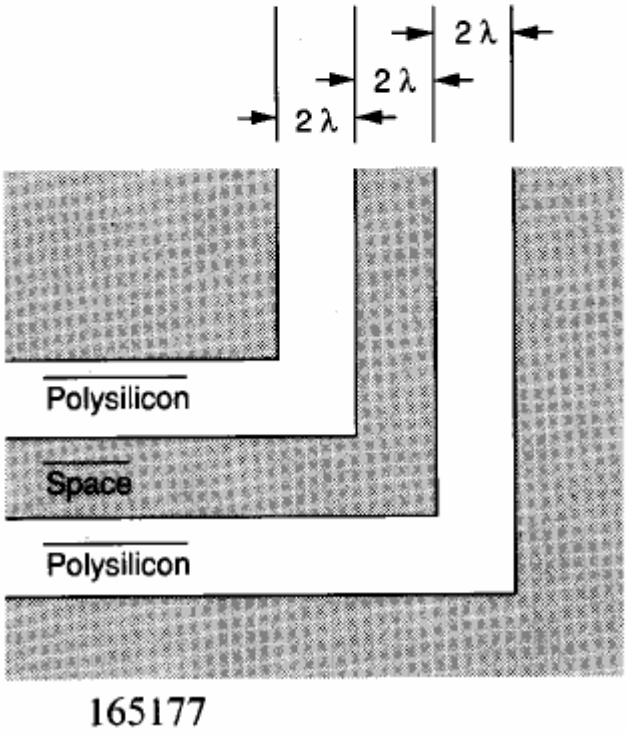


FIGURE 10.4-2
Complemented layout, where spacings become widths.

1. A 1 λ error should not be fatal, although the intended performance of the integrated circuit may be degraded.
2. A 2 λ error may be fatal and almost certainly will degrade the performance of the integrated circuit.

Consider the minimum width of 2 λ for the polysilicon conductor shown in Fig. 10.4-3*a*. If the width of the actual polysilicon that is fabricated on the chip is 1 λ less than this minimum, as in Fig. 10.4-3*b*, the polysilicon will still conduct, although its resistance will double. If the fabricated polysilicon conductor is 1 λ wider than the minimum, as in Fig. 10.4-3*c*, the resistance is lowered, but the polysilicon still functions as a conductor. Thus, a change in width of 1 λ does not cause an obviously fatal problem for the polysilicon interconnection.

Now consider a 2 λ deviation from the design width of 2 λ. If a minimum width polysilicon conductor is narrowed by 2 λ, as in Fig. 10.4-4*a*, there is no conductor left—certainly a fatal error unless the connection was redundant. If the width is increased by 2 λ as in Fig. 10.4-4*b* and the minimum polysilicon spacing is 2 λ, there is a chance that the polysilicon conductor will contact an adjacent polysilicon conductor, causing a short circuit—also a fatal error.

Other design rules involve more than one level and are harder to remember and to verify. As an example of a two-level rule, consider that a transistor is created by the area common to polysilicon and diffusion. This transistor area must satisfy the 2 λ minimum length rule, so the smallest transistor size is 2 λ by 2 λ. The diffusion areas for the source and drain of a transistor also must satisfy a 2 λ minimum length. This rule is sometimes confusing from a layout viewpoint since the source, the drain, and the transistor gate area appear as one contiguous diffusion area. Thus, a source area 1 λ long combined with a transistor area 2 λ long and a drain area 2 λ long, shown in Fig. 10.4-5*a*, appears as a diffusion area 5 λ long and does not seem to violate the 2 λ diffusion length rule. However, Fig. 10.4-5*b* shows that a source only 1 λ long could disappear as a result of a 1 λ alignment error between polysilicon and diffusion—thus a 2 λ rule must be specified for the transistor source/diffusion dimensions. Typical design rule sets for several processes, including NMOS and CMOS, are provided in Appendix 2.

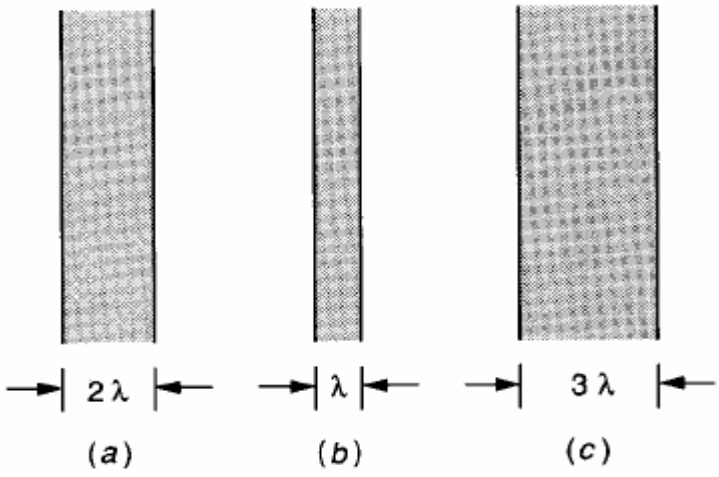


FIGURE 10.4-3
DRC degradation meta rule.

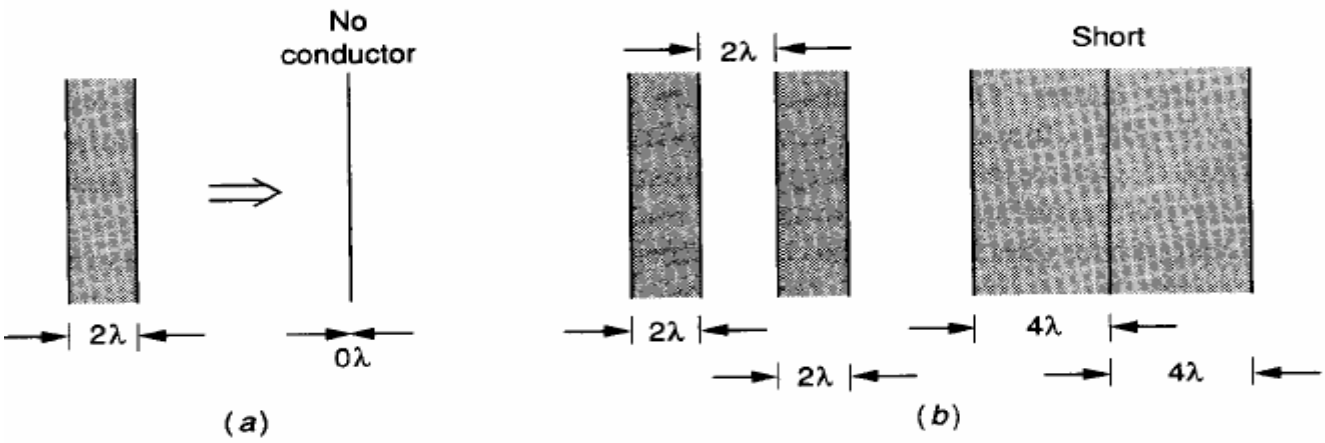


FIGURE 10.4-4
DRC fatality meta rule.

Computer Design Rule Checks

If a designer creates or changes a geometrical specification file manually, a *design rule check* (DRC) is required. Because of the large number of geometries and the wide variation in number and style of geometrical design rules in today's circuits, computer-based DRCs are necessary. Two different styles of DRC programs are in wide use. These can be categorized as polygonal checks and raster scan checks. Both styles will be described briefly.

Polygonal design rule checks are widely used within the semiconductor industry. The geometrical specification file is expanded to produce polygons defining all connected areas for the layer(s) of interest. Note that the layer of interest may be a composite area such as active transistor area or perhaps depletion transistor area. Or it may be a difference area such as the ion implantation overhang created by subtracting the depletion transistor area from the ion implan-

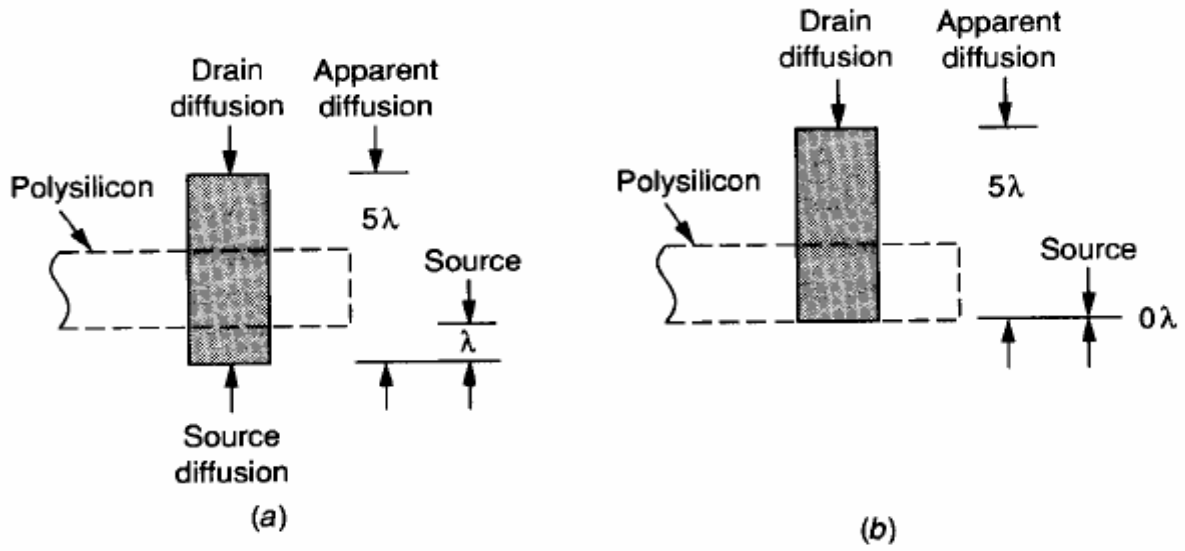


FIGURE 10.4-5
Transistor source width.

tation area. These special areas can be defined by logical operations on primitive layers. Once the polygonal definitions are formed, they can be analyzed for width and spacing errors. One valuable feature of encircling a connected area with a single polygon is that electrical connectivity information is immediately available. Polygonal design rule checks require substantial computing resources because of the many mathematical operations that must be performed during the check.

Design rule checks can also be performed in a relatively simple way as raster scan checks by passing small filters over a rasterized image of the integrated circuit. To allow this, an entire geometrical specification file is instantiated (expanded into the geometries and layers that represent the layout) within a two-dimensional array where the dimensions represent the x and y coordinates of a point and the contents are binary variables to indicate the presence or absence of each layout level. The resolution of the x and y coordinates limits the precision of the design rule checks. Filters such as a 4×4 array,[12] a "plus" symbol, or a circled "plus" symbol[13] have been used to scan the instantiated layout to check for design rule violations. These methods are conceptually simple and computationally clean, but lack the accuracy and connectivity information of the polygonal methods.

Design Rule Checker Output

To demonstrate the results from a raster scan DRC program, several errors were placed in a geometrical specification file. The layout for this file is shown in Fig. 10.4-6. The resulting output from the DRC program is shown in Fig. 10.4-7. The DRC program outputs a heading that gives the name of the file, the date and time, the bounding box coordinates for the checked area, and the macro number. Below the heading, a list of all vertical and horizontal errors is provided. This particular sample contains three vertical and four horizontal errors. Each violation is shown by a one-line entry containing the identification of the violated rule, the x and y coordinates of the violation, the violation or error distance, and the length over which the violation occurred. The resolution of the layout of Fig. 10.4-6 and the DRC results of Fig. 10.4-7 is 0.5 λ.

Definitions of the seven rule violations from Fig. 10.4-7 are given in Table 10.4-1. In each case these errors involve a spacing violation. For example, Rule 6.2 is a metal spacing error. A glance at the upper left corner of Fig. 10.4-6 shows a T formed by a long horizontal metal section and a short vertical metal section separated from the horizontal metal (top of the T) by about 1 λ. From Rule 6.2, the spacing must be at least 3 λ unless the two metal sections should be joined, in which case the spacing would be zero. As an exercise, the reader should find the location of each of the errors listed in Fig. 10.4-7.

Once the cause of an error is determined, corrective action must be initiated. Since the DRC output gives the exact x and y coordinates of the violation, it is usually relatively simple to use an interactive graphics CRT to display the error. Actually correcting the error may not be so simple. If the layout is loosely packed, correction in place by adjusting a single geometrical figure can possibly be done. For some layouts, however, an error will occur in a space-critical area, requiring changes of a large number of geometries. For this reason, it is crucial to generate a correct layout through automatic means or, in the case of a handcrafted design, to check the layout frequently for geometrical design-rule errors as it is generated. With care, errors are caught early before correction causes difficult problems.

TABLE 10.4-1
Design rule error definitions

Rule	Length	Definition
1.2	3 λ	Diffusion spacing
4.2	2 λ	Polysilicon spacing
4.3	λ	Polysilicon-to-diffusion spacing
5.3	λ	Polysilicon larger than contact
5.6	λ	Metal larger than contact
6.1	3 λ	Metal width
6.2	3 λ	Metal spacing

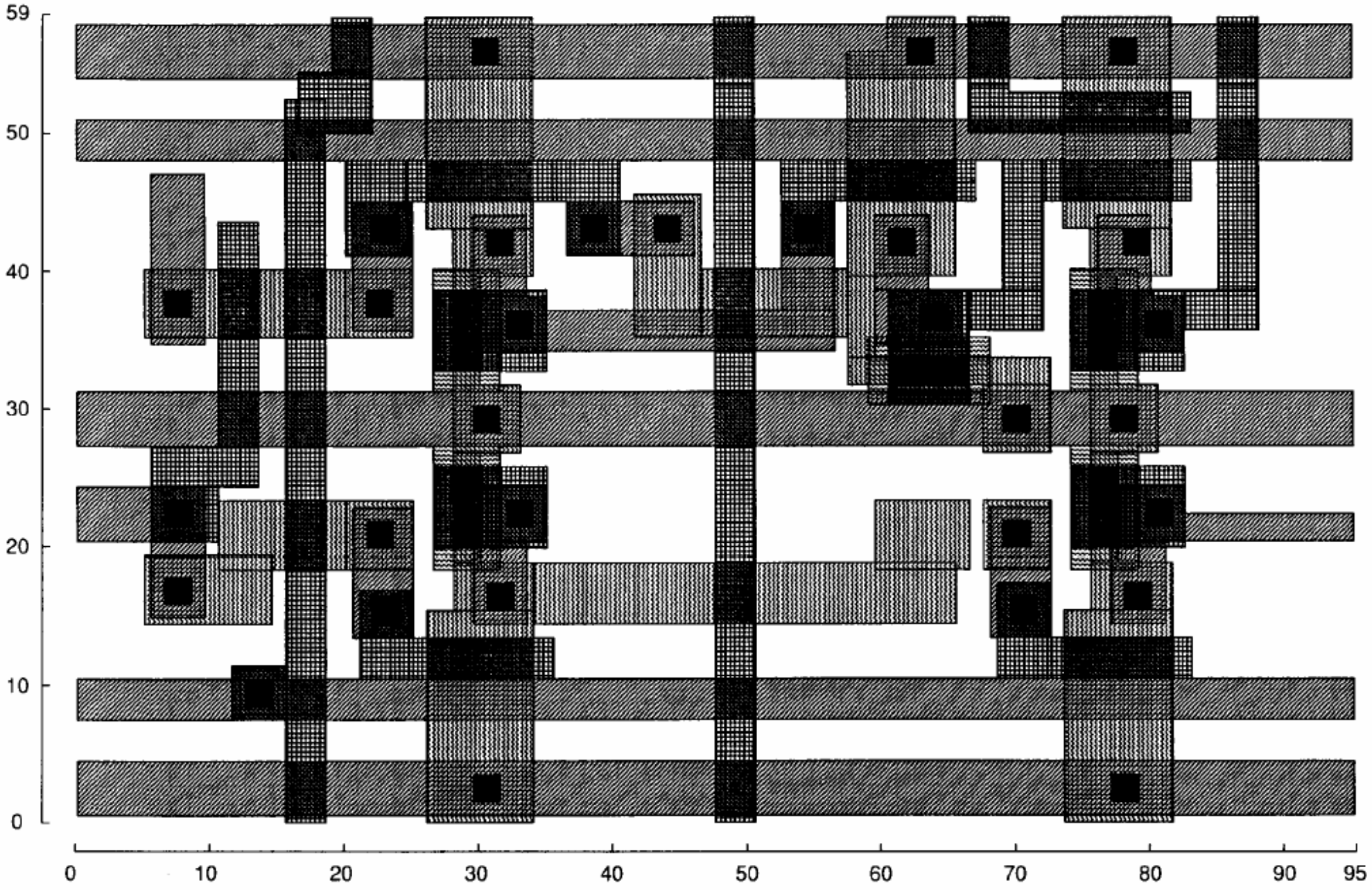


FIGURE 10.4-6
Sample layout for DRC.

LDRC version 3.115

Design rule check of file: BGA.TAM
Date 9-MAR-89 Time 21:08:05
X min = 0.0 X max = 95.0
Y min = 0.0 Y max = 59.0

Macro name is BGMLT
Macro number is 99

Vertical errors

Rule	X loc	Y loc	Error	X len
6.2	5.5	47.5	1.0	4.0
5.3	22.0	17.0	0.5	2.0
6.1	82.5	20.5	1.0	12.5

Vertical error count: 3

Horizontal errors

Rule	X loc	Y loc	Error	X len
4.3	10.5	20.5	0.0	3.0
4.2	18.5	41.5	0.5	7.0
1.2	66.5	18.5	1.0	5.0
5.6	80.0	41.5	1.0	2.0

Horizontal error count: 4

Total number of Design rule violations: 7

Design-Rule Checker Execution:

CPU Time 0: 0:26.06
Page Faults 354

FIGURE 10.4-7
DRC output for Fig. 10.4-6.

The DRC program used here was run in the batch mode on a computer after the layout was complete. Many CAD systems allow DRCs as geometries are entered through an interactive graphics CRT using an incremental DRC program. Either the designer is prevented from placing geometries that would violate design rules, or a pending violation is flagged immediately by an error message. This minimizes the need for major changes after the layout is almost complete.

DRCs are one of the more time-consuming, yet important, design verification steps. Both polygonal and raster scan DRCs are possible. A good DRC program provides output that accurately identifies the type and location of each error. A good interface between the DRC program and an interactive graphics editor is important for displaying and correcting DRC errors.

CIRCUIT EXTRACTION

After the design and layout process is complete, MOS circuits are characterized by a machine-readable specification prior to the mask-making step. This specification is usually a geometrical specification file as described earlier. This file contains all the information about the geometries, levels, and placements for the circuit to be fabricated. Because geometrical specification files contain large quantities of detailed information about the integrated circuit, it is difficult for a designer to determine whether this information accurately describes the circuit that was intended. Fortunately, computerized methods exist to extract the circuit information from the geometrical specification file. The process of extracting the circuit information from the geometrical description is called *circuit extraction*.

A circuit extraction program expands the geometrical specification file of the integrated circuit into a layer-by-layer description of the geometries and their placements. This description is then scanned to locate all transistors and interconnections for the circuit. A result of the circuit extraction program is a net list. A *net list* is a set of statements that specifies the elements of a circuit (for example, transistors or gates) and their interconnection. Individual transistors are described along with the nodes to which they connect. This information allows creation of a circuit diagram based on the actual geometrical specification file. Most importantly, the extracted circuit can be compared with the original circuit specified by the designer so that differences are annotated. A difference usually indicates an error that must be corrected. This comparison is called an LVS (layout versus schematic) design verification step.

In addition to providing the details of circuit interconnections, circuit extraction is useful for calculating layout areas and perimeters for each integrated circuit layer at each node of the circuit. These layout areas and perimeters can be used to accurately calculate the parasitic capacitances and resistances that load the active devices. Prior to the layout and extraction step, most circuit parasitics can only be estimated by the designer. With accurate capacitance and resistance values from circuit extraction, a design can be accurately simulated to ensure correct operation. Thus, circuit extraction is an essential design verification tool for accurate characterization of modern integrated circuits.

Circuit Extractor Output

As a minimum, the output from a circuit extraction program should contain a complete list of transistors showing the type of transistor (p-channel, n-channel, depletion, etc.) and the nodes to which the transistor is connected. The circuit of Fig. 10.5-2 was extracted to show typical output. A sample of such output, called a net list, is shown in Fig. 10.5-3.

The extracted output of Fig. 10.5-3 lists an arbitrary transistor number; the drain (DS1), source (DS2), and gate (G) connections; the type of transistor (enhancement or depletion); the shape (*ok* means rectangular); the length and width of the transistor; and the x and y coordinates of the upper left corner of the transistor. All dimensions are based on the parameter λ. The resolution of Fig. 10.5-2 and its extracted output listings is 0.5 λ. With this information, transistor size can be verified, individual transistors can be located, and the V_{DD} connection for the depletion transistors (the normal case) can be verified. The net list provides sufficient information to allow reconstruction of a transistor-level circuit diagram

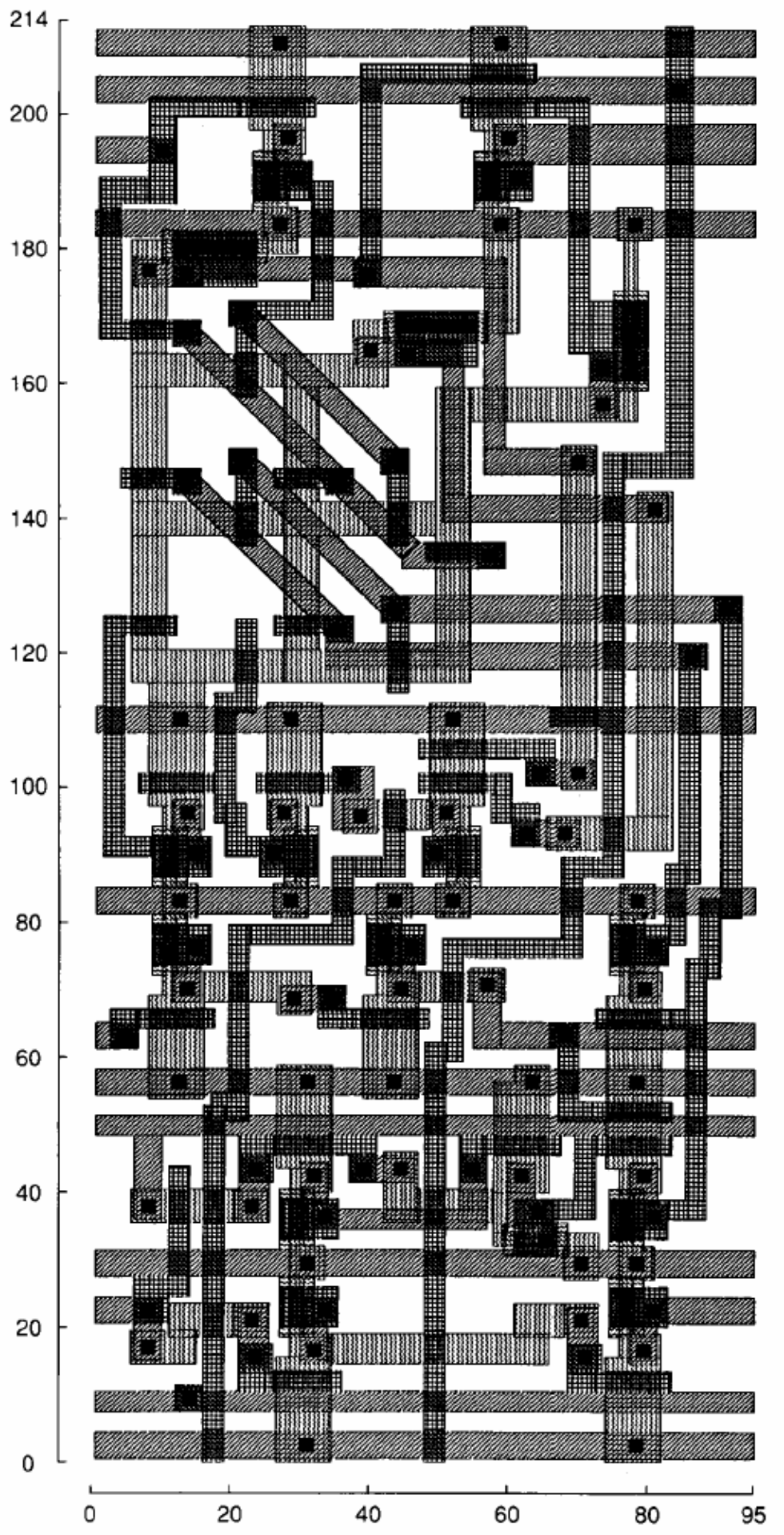


FIGURE 10.5-2
Sample layout for circuit extraction.

LEXTRACT version 3.337

Date 4-MAR-89 Time 19:54:46

X min = 0.0 X max = 95.0
Y min = 0.0 Y max = 214.0

Macro name is BGMLT
Macro number is 99

Final merge node list

Num	DS1	DS2	G	Type	Shape	Length	Width	X-loc	Y-loc
1	GND	42	3	enhN	ok	3.0	8.0	54.0	208.5
2	GND	6	4	enhN	ok	3.0	8.0	22.0	203.5
3	42	7	5	enhN	ok	3.0	8.0	54.0	203.5
4	6	VDD	6	depN	ok	6.0	2.0	24.0	194.0
5	7	VDD	7	depN	ok	6.0	2.0	56.0	194.0
6	3	VDD	3	depN	ok	12.0	2.0	11.0	182.0
7	VDD	5	5	depN	ok	12.0	2.0	76.0	173.0
8	3	51	4	enhN	ok	3.0	5.0	5.0	170.0
9	9	VDD	9	depN	ok	12.0	2.0	43.0	170.0
10	51	9	6	enhN	ok	3.0	5.0	20.0	165.0
11	51	55	11	enhN	ok	3.0	5.0	5.0	148.0
12	9	12	4	enhN	ok	3.0	5.0	27.0	148.0
13	55	12	10	enhN	ok	3.0	5.0	20.0	143.0
14	12	5	6	enhN	ok	3.0	5.0	42.0	143.0
15	5	13	4	enhN	ok	3.0	5.0	49.0	137.0
16	55	GND	14	enhN	ok	3.0	5.0	5.0	126.0
17	12	17	11	enhN	ok	3.0	5.0	27.0	126.0
18	GND	17	15	enhN	ok	3.0	5.0	20.0	121.0
19	17	13	10	enhN	ok	3.0	5.0	42.0	121.0
20	3	18	2	enhN	ok	3.0	5.0	67.0	112.5

FIGURE 10.5-3
Partial net list generated from Fig. 10.5-2 by circuit extractor (VDD and GND labels entered by user).

(not shown) for the integrated circuit. The extracted circuit diagram can be compared with the intended circuit diagram for omissions or errors.

Additional information based on the circuit extraction should be provided. For example, for each integrated circuit layout level, a complete list of nodes with their corresponding areas and perimeters can be provided. If the capacitance per unit area is known for each level, the circuit extraction program can provide an accurate estimate of the capacitance at each node. Figure 10.5-4 provides a partial circuit extractor output for the layout of Fig. 10.5-2 showing the details of the integrated circuit layers that form the nodes of a circuit. For each extracted geometry, this output lists the area, top edge length, left edge length, x and y coordinates of the upper left corner of the geometry, the new merged node number, the old node number assigned to the geometry during extraction, the layout level, and the node name (if any).

LOGIC AND SWITCH SIMULATION

Digital integrated circuits are designed to operate with binary representations for data. The basic presumption is that only two logic states are important for each signal line. Thus, knowledge of a precise voltage versus time characteristic for each node in the circuit is not necessary to design or analyze digital circuits. For many purposes, this simplifies both the circuits and their analysis compared to analog circuits. Nevertheless, computer simulation and verification of a circuit's functionality are necessary. Even though a digital circuit is designed based on logic gates, the logic gates are fabricated from the basic transistors and conductors allowed by the integrated circuit process. Therefore, it is often the case that the electrical operation of a simple logic circuit must be characterized by using a circuit simulator such as SPICE.

Though circuit simulation of digital circuits is frequently used, such circuit simulation has several drawbacks. As described in the previous section, the large number of logic gates in most digital integrated circuits precludes circuit simulation of the entire system because of the extended computer time required. Also, standard circuit simulators provide more detail about circuit voltages than is required to analyze a logic circuit. In an effort to reduce computer simulation time and to provide appropriate data to characterize the operation of digital circuits, *logic simulators* were developed.

Logic-level Simulation

Logic simulators allow specification of the operation of a circuit block in terms of its behavior. For example, a simple logic gate is described by its behavior, such as AND, OR, or NOT. More complex digital blocks such as full adders and multiplexers are each described by their corresponding behavior rather than their circuit structure. The circuit inputs are specified as binary values that change at discrete time intervals. Logic simulator outputs are provided as binary values as well. Pure logic simulation does not model time delays through logic blocks. Only the logical behavior of the simulated system is considered, although the concept of sequence wherein one action precedes another is important. Timed logic simulation considers the delays of logic gates and blocks in determining when outputs will change. Because a logic simulator models the circuit in terms of an abstracted (less detailed) representation, larger circuits can be simulated in a much shorter length of time than with circuit simulation. Consider the following example.

Commercial logic simulators model digital logic in terms of four or more states. As a minimum, these states include 1, 0, *X*, and *Z*. The logic values 1 and 0 model the high and low logic states. The value *X* is used to model an unknown condition. For example, the value of an internal logic node may be unknown when simulation is started. The value *Z* is used to model high-impedance (undriven) nodes. A tri-state bus with all driving circuits turned off is an example of this condition. Additional states may be defined to model the relative driving strength of logic outputs. Of course, as the number of allowable states increases, the simulator complexity and run time increase correspondingly.

Many logic simulators provide a variety of digital blocks for use in modeling a digital system. Besides the simple logic gates and more complex logic blocks mentioned previously, models for large digital blocks such as ROMs, RAMs, PLAs, ALUs, and even FSMs are often provided. Simulation capability is normally provided for both synchronous and asynchronous sequential circuits in addition to simple combinational logic.

Most logic simulators today are *event driven*. That is, calculations are required only in response to external or internal events. External events include changes in the state of inputs to the circuit. An internal event occurs when the output of a logic function changes in response to changes in its inputs. For example, when the input to an inverter changes, the corresponding change in the inverter output is considered an event. The use of event-driven rather than fixed time-step simulation algorithms reduces the time required for simulation of a circuit.

The capability of logic simulation is often measured in terms of *events per second* or *evaluations per second*. Whenever the inputs to a logic block change, an *evaluation* must occur to determine the correct output for the logic block. Thus, an evaluation is the application of a circuit's inputs to its behavior in order to determine its outputs. An average factor of 2.5 evaluations per event is typical for digital circuits. The performance of logic simulators depends on many factors including the number of logic states, the cleverness of the algorithms chosen for simulation, and the execution speed of the computer on which the simulator is run. An execution rate of several thousand events per second is common for today's logic simulators on typical computer workstations.

Switch-level Simulation

MOS integrated circuits present special problems for standard logic simulators because of bidirectional pass transistors, transmission gates and charge storage. Pass transistors are used frequently because of their desirable power dissipation and interconnection characteristics. Pass transistors are difficult to simulate as simple logic gates with a standard logic simulator. It might seem that the pass transistor of Fig. 10.7-3*a* could be simulated by using the AND gate of Fig. 10.7-3*b*. The following discussion shows why this is impractical.

A simple analysis of the operation of the circuits of Fig. 10.7-3 shows that the two circuits are not equivalent. Assume initially that both inputs and the output are low for both circuits. If a logic 1 is placed on a single input, the output remains low for both circuits. If a logic 1 is placed on both inputs, the output goes high for both circuits. If a logic 0 is placed at the *i* input of the two circuits, the output goes to a 0 for both circuits. Thus far, the operation of the two circuits seems identical. However, assume that all inputs and outputs are initially high. Further, consider that the source diffusion of the output of the pass transistor provides parasitic capacitance to ground. If the *c* input to both circuits is moved to a logic 0, the AND gate output goes to a logic 0 while the pass transistor output remains high because of the charge storage at its output. Clearly, the operation of the pass transistor cannot be accurately modeled in this fashion. Either a more complex logic circuit is required, or the logic simulator must be modified to account for drive strengths and charge storage. Examples of drive strength are *driven*, *resistive pullup*, and *undriven*. The output of the pass transistor just considered is undriven when its gate terminal is at 0 V.

Because selector circuits constructed from pass transistors and transmission gates are widely used within MOS circuitry, a logic simulator for MOS must allow specification of individual transistors and their connections in addition to simple logic gates. When a logic simulator can describe transistors in addition to standard Boolean logical primitives, it is called a *switch-level simulator*.

A typical switch-level simulator operates on circuits described by nodes, transistors, and logic gate primitives. Nodes are equipotential points to which one or more terminals of one or more transistors or logic primitives are connected. Each node has an associated name, logic state, capacitance (to ground), list of events, and perhaps other information. Each transistor has a type (n-channel, p-channel, or depletion), effective resistance (width and length are required), and node connection for its terminals. Macros are often allowed to describe circuits composed of several transistors; for example, logic gates may be constructed from nodes and transistors. These logic gates are then used as primitives.

A byte-wide MOS binary adder circuit will be used as an example to show the operation of a switch-level simulator.[17] The circuit for a full-adder stage is

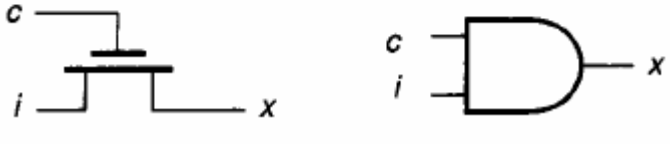


FIGURE 10.7-3
(*a*) Pass transistor logic, (*b*) AND-gate logic.

given in Fig. 10.7-4, and the corresponding input net list for the switch-level simulator is provided in Fig. 10.7-5. This net list describes the circuit of Fig. 10.7-4 in terms of four primitive elements: invert, trans, nor, and pulldown. The net list starts with a definition of a single-bit adder macro and its five inputs {*a b cif phi1 phi2*} and two outputs {*sum cof*}. Additionally, seven local signals {*af bf ci p pf k phi2f*} that are internal to the full-adder macro are specified. Each primitive element is then instantiated with its connections to other circuit nodes defined by arguments. The formats for these four procedure calls are: (invert out in), (trans gate source drain), (nor out in0 in1 in2), and (pulldown drain gate).

Next, eight single-bit full adders are combined to define a byte-wide binary adder, as shown in Fig. 10.7-6. The external nodes of the byte-wide full adder are first defined. The **a**, **b**, **cof**, and **sum** nodes represent 8-bit vectors that are expanded by the repeat statement. Signals *phi1* and *phi2* are the nonoverlapping two-phase clock inputs. The connect statement joins the *cifi* carry-input scalar to the first carry-in bit, *cof.0*. The repeat statement next creates eight copies of the full-adder circuit.

The results of a sample switch-level simulation run for the byte-wide adder are explained next. The input vector **a** was set to 11111111, while the input vector **b** was set to 00000000. This condition provides the longest carry propagation path for the full adder. The initial carry-in bit *cifi* is set to the low-true condition. A nonoverlapping two-phase clock is defined with each phase high for 90 ns and a 10 ns separation between phases. The results from a simulation for a complete

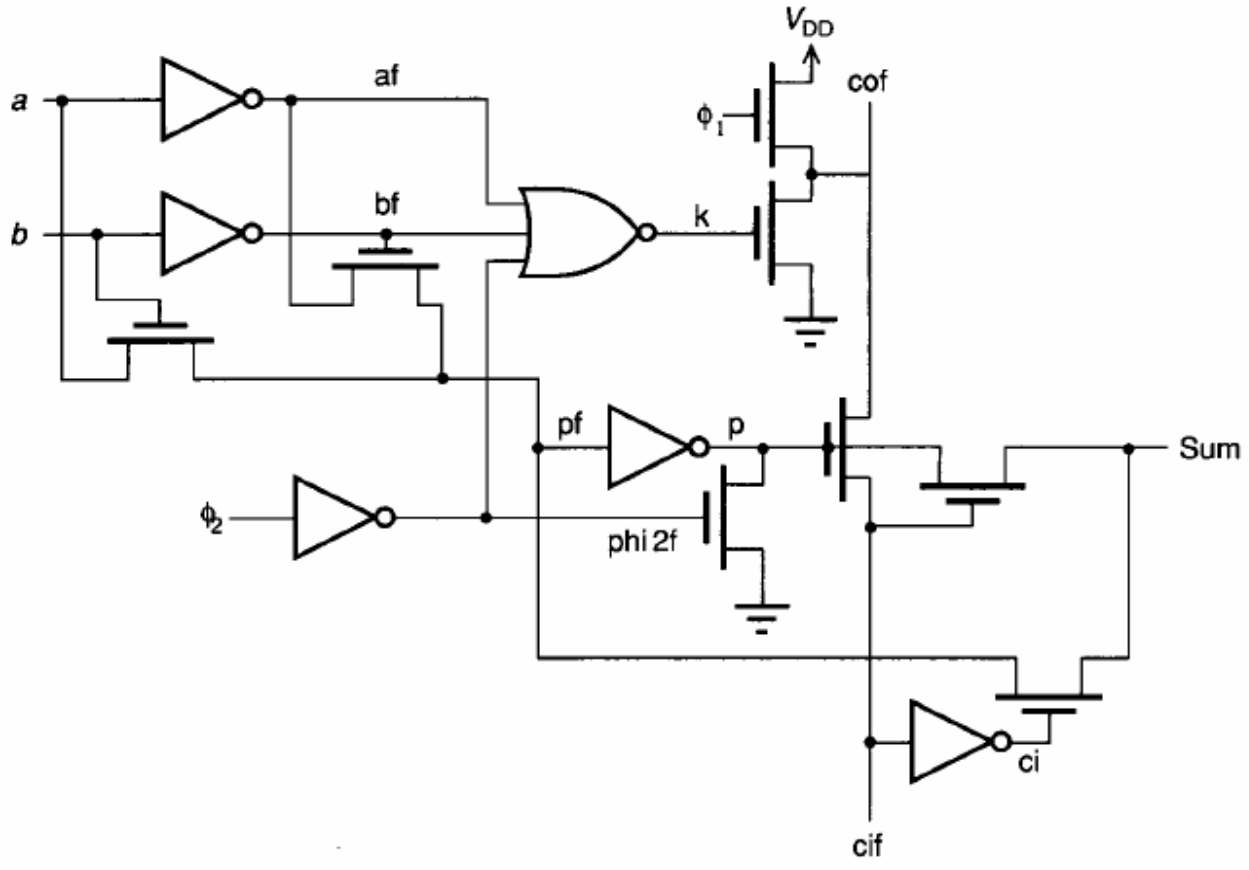


FIGURE 10.7-4
Single-bit slice of clocked full-adder circuit.

```
;Begin Full-Adder Macro
(macro adder (a b cif phi1 phi2 sum cof)
  (local af bf ci p pf k phi2f)
  (invert bf b)
  (invert af a)
  (trans b pf a)
  (trans bf pf af)
  (invert (p 2 16) pf)
  (invert (ci 2 16) cif)
  (trans cif sum p)
  (trans ci sum pf)
  (invert phi2f phi2)
  (nor k af bf phi2f)
  (pulldown cof k)
  (pulldown p phi2f)
  (trans phi1 cof vdd)
  (trans p cof cif)
)
;End Full-Adder Macro
```

FIGURE 10.7-5
Input net list for logic simulator describing circuit of Fig. 10.7-4.

```
;Instantiate Byte-Wide Adder
(node a b cifi phi1 phi2 sum cof)
(connect cifi cof.0)
(repeat i 1 8
  (adder a.i b.i cof.(1 - i) phi1 phi2 sum.i cof.i)
)
;End of Byte-Wide Adder
```

FIGURE 10.7-6
Input net list for a byte-wide binary adder.

cycle (200 ns) of the two-phase clocks are given for the first and last sum (*sum.1*, *sum.8*) and carry-out (*cof.1*, *cof.8*) bits only. Only changes in logic value of these bits are provided; that is, only simulator events for these bits are included. A typical event produces a statement with the format: name = value @ time.

ϕ_1 cycle: (t = 0 ns to 90 ns), precharge
cof.8 = 1 @ 2.4
cof.1 = 1 @ 2.6
sum.1 = 0 @ 2.8
sum.8 = 0 @ 3.2

ϕ_2 cycle: (t = 100 ns to 190 ns), evaluate
cof.1 = 0 @ 103.2
cof.1 = 1 @ 104.4
sum.8 = 1 @ 104.9
cof.1 = 0 @ 109
sum.8 = 0 @ 129.8
cof.8 = 0 @ 130

Note that all carry bits are precharged to a 1 during each ϕ_1 cycle. According to the simulation results shown, the *cof.8* bit changed to 1 at 2.4 ns and the *cof.1* bit changed to 1 at 2.6 ns after ϕ_1 was set high. As can be determined from the circuit connections of Fig. 10.7-4, the sum bits should be set to 0 during each ϕ_1 precharge cycle. The *sum.1* bit went to 0 at 2.8 ns and the *sum.8* bit went to 0 at 3.2 ns after ϕ_1 was set high.

During the ϕ_2 evaluate cycle, the carry and sum bits are set according to the sum of the two addends **a** = 11111111 and **b** = 00000000 and the carry in *cifi* = 0 (indicates a carry in). During the evaluate cycle *cof.1* changed to 0 at 3.2 ns, to 1 at 4.4 ns, and then back to 0 at 9.0 ns after ϕ_2 was set high. The most significant carry bit, *cof.8*, was set to 0 some 30 ns after ϕ_2 was set high. Also, *sum.8* was set to 1 at 4.9 ns and then to 0 at 29.8 ns after ϕ_2 was set high. For the input vectors given, each full-adder stage should have set its sum bit to 0 to indicate a sum of 0 and its carry bit to 0 to indicate a carry out of 1 (the carry bits use negative logic). The final results from simulating the first clock cycle are as expected. Note that the final event (*cof.8* set to 0) occurred 30 ns after the ϕ_2 clock was set high.

Prior to the second clock cycle, the carry-in bit is set to a false condition (*cifi* = 1). The following simulation results are for the second clock cycle (200 ns $\leq t <$ 400 ns).

ϕ_1 cycle: (t = 200 ns to 290 ns), precharge
cof.8 = 1 @ 200.2
cof.1 = 1 @ 200.4

ϕ_2 cycle: (t = 300 ns to 390 ns), evaluate
sum.8 = 1 @ 304.9
sum.1 = 1 @ 304.9

During the second ϕ_1 cycle, the carry bits change as they are each precharged to 1. The sum bits do not change during ϕ_1 since they were already each left set to 0 after the previous ϕ_2 cycle. During the second ϕ_2 cycle, the sum and carry bits should be changed to indicate the sum of the two addends **a** = 11111111 and **b** = 00000000 and the carry in *cifi* = 1. Thus, all sum bits should be set to 1 and all carry out bits should be set to 1 indicating no carry out. The simulation results show that the sum bits are each correctly set to 1 during the second ϕ_2 cycle. The carry bits do not change since they were each set to 1 during the precharge cycle.

For a third clock cycle (400 ns $\leq t <$ 600 ns), the carry in bit is set to 0 again (*cifi* = 0) and the results of the first clock cycle are repeated. These results are as follows.

ϕ_1 cycle: (t = 400 ns to 490 ns), precharge
sum.1 = 0 @ 402.8
sum.8 = 0 @ 403.2

ϕ_2 cycle: (t = 500 ns to 590 ns), evaluate
cof.1 = 0 @ 503.2
cof.1 = 1 @ 504.4
sum.8 = 1 @ 504.9
cof.1 = 0 @ 509
sum.8 = 0 @ 529.8
cof.8 = 0 @ 530

The previous results for three clock cycles demonstrate the operation of a switch-level simulator. Both the timing of the byte-wide adder and the correct logical operation of the adder are observed for the input conditions provided. Other switch-level simulators have different input and output formats and different capabilities, but all operate assuming discretized values for the circuit variables, and all produce results much faster than complete circuit simulation.

Hardware Logic Simulation

Even with the increased speed of logic simulators as compared with circuit simulators, full simulation of large digital circuits via general-purpose computers is not practical. An alternate approach is in use by several companies. Special-purpose hardware that executes many simulation steps in parallel has been developed to speed the simulation process. One of the early, large parallel simulators was the YSE (Yorktown Simulation Engine)[18] developed by IBM. This hardware consists of hundreds of identical processing units that each simulate part of the target circuit. By spreading the calculations over a large number of processors, even large-mainframe computers can be simulated in detail. Of course, such special-purpose hardware is expensive to build and to operate. Even so, several companies now offer hardware accelerators to enhance the speed of logic simulation.

In the future, methods of machine verification other than total logic simulation must be found. Logic simulation time increases exponentially with the number of logic components to be simulated. Thus, faster computers are necessary to simulate next-generation computers that contain more logic components. But how can the next-generation computers be built if the simulation capability of present-generation computers is inadequate?

Two current approaches to this problem are verification proofs and hierarchical simulation. For relatively simple hardware, it has been possible to verify correct logical operation by mathematical proofs. Unfortunately, the utility of this method diminishes quickly as the size and complexity of the hardware increase. The second method, hierarchical simulation, attempts to model the target machine at various levels of abstraction. Small blocks of hardware are verified by logic simulation. These blocks are then interconnected and simulated together without the internal detail of each block. Neither of these methods has been entirely successful, and both are now active areas of research and development.

TIMING ANALYSIS

For most digital circuits, a very important parameter is the maximum rate at which the circuit can correctly process data. For microprocessors, the processing speed is usually given in MIPS (millions of instructions per second); for scientific calculations, the rate of execution is given in FLOPS (floating-point operations per second); and for logical inference machines, the characteristic measure is LIPS (logical inferences per second). The execution rate of each of these machines is limited by parasitics and governed by its input clock. A primary goal in the design of a digital computing machine is to operate with the fastest possible input clock.

Each digital integrated circuit has a maximum rate of operation. This rate of operation is limited by the output drive capability of its logic elements and by the capacitance and resistance of the loads they must drive. In a FSM (finite-state machine), the clocking rate is limited primarily by the longest path through its combinational logic section. For integrated circuits composed of large blocks of circuitry, the maximum clocking rate may be limited by signal lines that must carry information between the blocks. The designer's task, then, is to find those paths in an integrated circuit design that cause the maximum delay and then to modify the circuitry to minimize that delay.

Finding the longest delay paths, called *critical paths*, for an integrated circuit is not a simple task. Until recently, the most common technique for finding critical delays was for the designer to perform detailed circuit simulation on the paths that were suspected of contributing long delay times. Of course, using circuit simulation for this task was not foolproof. Many times an unsuspected path that was not considered for simulation would limit the maximum clock speed. More recently, computer programs have been designed specifically to seek out delay paths directly from the circuit definition without requiring simulation. This type of computer analysis is called *timing analysis*.

Timing Analysis Methodology

Timing analysis differs from circuit and logic simulation in that all possible signal paths are considered. Circuit simulation and logic simulation both require the specification of input signals to control the simulation. Thus, only delay paths that are exercised by the particular set of inputs are tested. For many digital circuits, it is computationally impossible to provide sufficient input conditions to test the circuit fully. Timing analysis tools work by tracing *signal paths* instead of simulating the circuit for specific inputs. Specifically, timing analysis uses *state-independent* path tracing. Each time a logic gate is encountered, the gate is assumed to pass the signal regardless of the state of the other inputs to the gate. A signal path is terminated only when an output is reached or a clocked storage element is found. With this method, all possible delay paths are tested.

An example of timing analysis signal propagation through two logic gates is shown in Fig. 10.8-1. The signal path starts at input x and reaches the NAND gate. Inputs a and b for the NAND gate are assumed high to allow continuation of the signal path. When the signal reaches the NOR gate, input c is assumed low

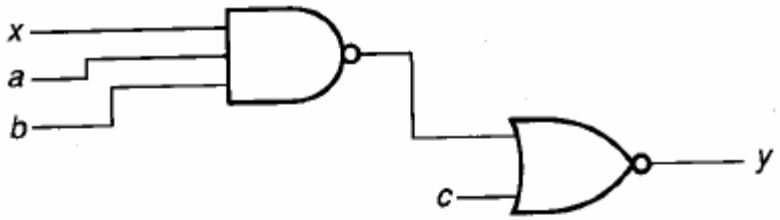


FIGURE 10.8-1
State-independent path trace.

to allow continued propagation of the signal. Finally, the signal reaches an output y, where it terminates. The delay for this signal path includes the contributions of the NAND gate, the NOR gate, and the series interconnections. The delay paths from x to y, b to y, a to y and c to y would all be found by a timing analysis of this circuit.

A second example shows a deficiency of timing analysis. From Fig. 10.8-2, signal paths from a to b and from a to c are expected. However, state-independent path tracing will also find a signal path from b to c and vice versa. Although the path from b to c is a real path, it will not normally be exercised within this circuit because node n is actively driven by the inverter. Analysis of additional paths that will not be exercised during operation of a circuit can degrade the performance of a timing analysis program. Circuit-level timing analyzers allow direction setting for pass transistors and transmission gates to circumvent this problem. Unfortunately, unless this is carefully done, some critical signal paths may be eliminated from consideration.

Timing Analysis Tools

To provide further insight into the capabilities of circuit-level timing analysis programs, two such programs will be described here. The first of these, called TV,[19] attempts to set directions for circuit elements by using rules. These rules, by setting some transistor directions, minimize the number of false paths that are found. The second tool, Crystal,[20] provides a wide range of capability, including improved delay models and coverage for circuits built from CMOS technology.

TV timing analyzer for NMOS designs, operates from extracted circuit parasitics and considers only stable, rising, and falling signal values. Program execution time is minimized by a *static analysis* that sets signal flow direction and clock qualification where possible. Otherwise, signal flow direction is determined from a set of direction-finding rules. Some of the rules are independent of design style. For example, the *constant-propagation rule* says that any transistor source

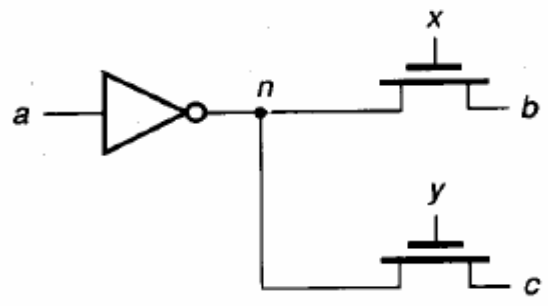


FIGURE 10.8-2
Problem paths for timing analysis.

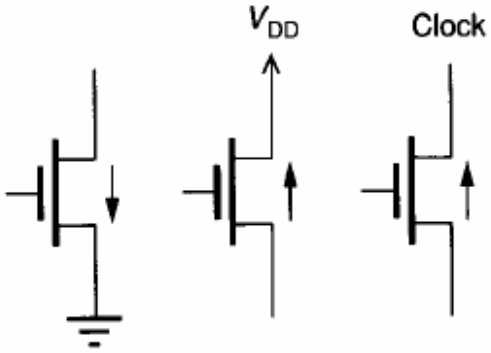


FIGURE 10.8-3
Constant propagation rule to set directions.

or drain connected to power, ground, or a clock must be a sink of signal flow, while the other terminal must be a source. Figure 10.8-3 demonstrates this rule, which by itself sets the directions for more than half the transistors in a typical circuit. Another rule, demonstrated in Fig. 10.8-4, is based on Kirchoff's current law. This rule, the *node current rule*, states that if all but one of the transistors to a node have a known direction, and the known transistors all sink or all source signal flow, then the unknown transistor must transmit flow in the opposite direction relative to the node.

Other signal-flow rules depend on technology or design style. For example, in an NMOS technology design, the *k-ratio rule* for inverters can be used to set direction. This rule is based on a standard device sizing ratio k as discussed in Chapter 7 for ratio logic. By finding the minimum resistance to ground through each unset (direction not specified) transistor connected to a pullup, a transistor can be considered as a pulldown (signal flow toward the pullup) or a pass transistor (signal flow away from the pullup), depending on the resistance ratio. The reasoning is that resistances to ground that satisfy the device sizing ratio k with respect to the pullup path must be part of the pulldown circuit for a logic gate. Transistors that cannot satisfy ratio rules can be safely classified as pass transistors and their direction set accordingly. Other rules cover pass transistors connected to a common node and having a common gate signal, and analogous structures where the direction of a boundary transistor can be determined, thereby allowing arrayed versions of the structure to have their directions set accordingly.

Signal path analysis is started from the clock or other input nodes. Paths are investigated in a breadth-first manner in accordance with the transistor directions that were set by the static analysis. Delays for paths are calculated based on the capacitance of the interconnections and the resistance of driving and series pass

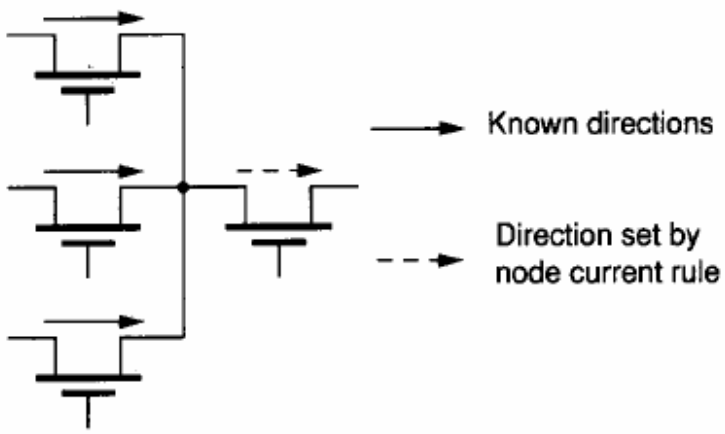


FIGURE 10.8-4
Node current rule.

transistors. Transistors are assigned a rising and a falling resistance from tables based on their use in the circuit. Signal direction changes are propagated so that a rising input signal to an inverter produces a falling output signal and vice versa. Pass transistors continue the direction of the input signal. Because path delays are calculated from a linearized model, the delays may differ from actual circuit values by 30% or more.

Output of the TV program includes a user-selectable number of the worst-case paths. Equivalent paths, such as parallel paths in a data bus, are condensed in the output list so that only the last path in the list is reported. Other useful information such as slack time for paths, excessive power used to drive a noncritical path, and nodes with unusually high capacitance are reported. The TV timing analyzer was successfully used in the analysis of the MIPS series of microprocessor chips[21] developed at Stanford University.

Another timing analysis tool, Crystal, was developed to analyze the RISC computer chips[22] developed at the University of California at Berkeley. This tool has found widespread use throughout the VLSI design community, particularly within universities. The timing analysis is based on a circuit description that is extracted directly from a geometrical specification file. This description includes transistor sizes and types, interconnection capacitance, and a rough calculation of interconnection resistance. A simple delay model is used for each stage to provide quick calculation of signal propagation delays along a path.

The Crystal timing analyzer was developed for MOS circuits with multiple nonoverlapping clocks. The program attempts to determine how long each clock phase must be to allow all signals to propagate to their destinations. The analysis is state-independent, so all possible paths are checked. The user must specify a minimum of information to begin the analysis. For two-phase clocking schemes, only two signals must be specified. One of the clock phases is specified as a rising edge or a falling edge to trigger the analysis. The other clock phase is specified as a stable low value. The reason for this can be seen from the shift register circuit of Fig. 10.8-5. Here a signal path trace is started from the ϕ_1 clock. Without a specified value for the ϕ_2 clock, the signal path would continue through all the stages shown. If the ϕ_2 clock is set to a stable low condition, then the signal path will terminate correctly after the first stage. The path delay

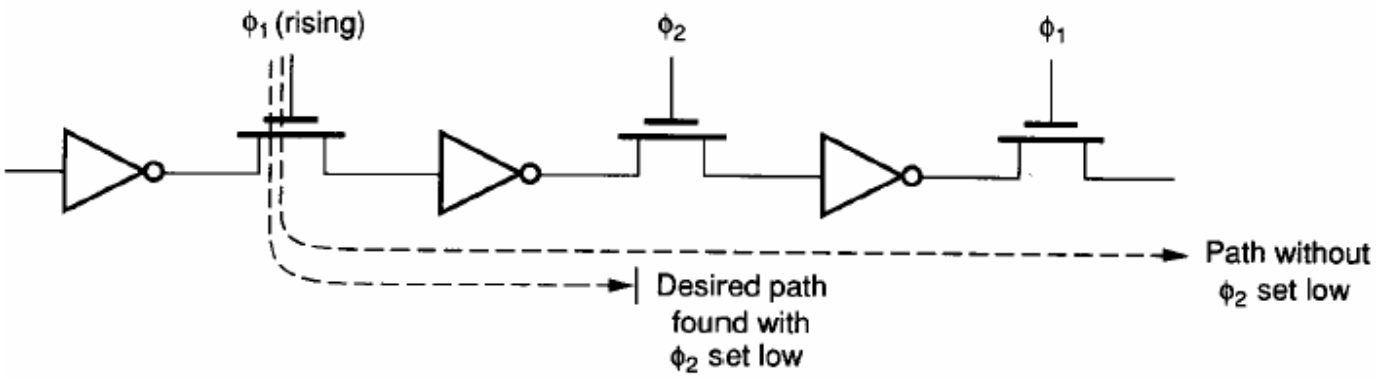


FIGURE 10.8-5
Clocked path analysis.

will consist of the time for the first inverter to discharge (charge) the input of the second inverter through the pass transistor gated by ϕ_1 plus the time for the second inverter to charge (discharge) the interconnection capacitance up to the input of the pass transistor gated by ϕ_2.

Each path trace for a signal is started from a rising or falling input specified by the user. As the signal path proceeds through inverters or logic gates, the appropriate rising or falling direction is determined to correctly model asymmetric stage delays. The path-trace analysis is done with a depth-first search algorithm. Thus, a signal path is followed until it reaches a circuit output or is stopped by a static signal specified by the user (like the ϕ_2 condition examined in the previous paragraph). Delay information from previous paths is maintained at each node so that the signal path can be aborted on later path traces through the same node if the cumulative delay is less than the stored value.

As with all state-independent timing analysis methods, the possibility of reporting false paths exists. A simple example is given in Fig. 10.8-6, where a signal path is gated by a signal and its complement. From a logical viewpoint, there is not a signal path from node a to node c because one of the AND gates will be disabled by x or $\bar{x}$. Since timing analysis is state-independent, this logical constraint is not recognized, and the path from node a through node b to node c will be considered and its delay calculated. A 1-of-n selector circuit is a classic example of this condition. In normal operation, only one path through the selector circuit will be enabled at any time, but state-independent timing analysis finds all n paths. In most timing analyzers the capability exists to set signals to a stable value to disable paths; however, this capability must be used carefully to avoid accidentally disabling critical delay paths.

To facilitate fast operation, Crystal uses a simple delay model consisting of an equivalent resistance for the drive transistor and a resistance and capacitance for the interconnections and load devices. The transistor drive model is table-driven with the equivalent resistance selected based on input signal slope and capacitive load value. This is not as accurate as a circuit-level simulation but is much faster. Once critical delay paths are found, they can be investigated with a circuit simulator if more accurate results are required.

In summary, timing analysis is an important tool for integrated circuit design. By using state-independent path tracing, it performs a function that is difficult, if not impossible, to perform with timed logic simulators. The execution time for timing analysis programs is determined by the size of the circuit

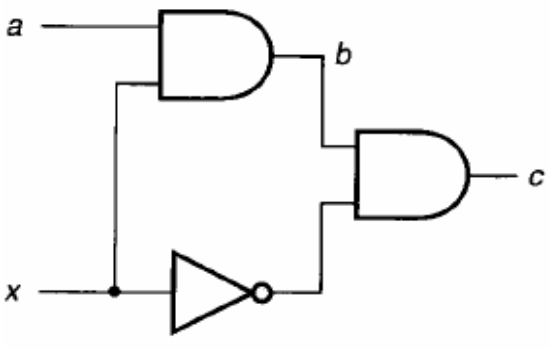


FIGURE 10.8-6
Logically impossible path.

being analyzed. While timing analysis is used to find and correct critical delay paths, correct functional operation can be verified with a logic simulator. Thus, logic simulation and timing analysis function as partners to ensure that a digital integrated circuit is functionally correct and that it operates at the proper speed.

REGISTER-TRANSFER-LEVEL SIMULATION

Specifications for the operation of digital integrated circuits are often given in terms of high-level operations on information. These high-level operations describe transformations on data as it moves from one storage device or register to another such device. For this reason, descriptions of this type are known as *register-transfer-level* descriptions.

Register-transfer-level descriptions provide a useful level of abstraction for the description and simulation of digital systems. The logic simulators described previously require too much detail about the exact logical structure of an integrated circuit for early design simulation. Also, because of the detailed specification of the logical structure of the circuit, complete logic simulation of an entire circuit such as a microprocessor requires impractically large computer resources. Alternatively, at the highest level, a natural language description of the function of a digital system is often ambiguous and vague. A concise natural language description of a next-generation computer might be, "build a new computer that is like computer XYZ, but is twice as fast and uses less power." To fill this gap between natural language descriptions and logical definitions, high-level description and simulation languages have been developed. Of these, register-transfer-level simulation languages allow specification and simulation of operations on data words, in addition to single-bit operations.

The operation of a digital system can be defined precisely through the use of a register-transfer-level description. In fact, one such language, ISPS (Instruction Set Processor Specification), was developed to allow unambiguous description and specification of computer operation.[23] The ISPS language allows data bits to be grouped into words. Logic and arithmetic operations are allowed on both bit-level and word-level entities as they are moved between storage registers. Operations common to most programming languages, such as conditional statements, if-then-else constructs, case statements, and procedures, are allowed. Thus, a register-transfer language is a special programming language tailored to describing the operation of digital systems.

Simple RTL

For demonstration purposes, a primitive register-transfer language (RTL) will be defined and used to describe the execution of one instruction from an early 8-bit microprocessor. This primitive RTL is defined in Table 10.9-1. The first operation required is the *transfer operation*—the contents of one group of bits (register) are placed into another storage device. Second, the common *arithmetic operations* of add and subtract are provided. Third, a simple *conditional capability* to alter control flow is added.

TABLE 10.9-1
Primitve RTL Definition

Operation	Description
A ← B	transfer
C ← A + B	addition
D ← A − B	subtraction
PC ← B if A = 0	conditional

The operation to be described using this RTL is the extended load of the A accumulator of the Motorola 6802 microprocessor. This microprocessor has a 16-bit address bus and a separate 8-bit data bus. The A accumulator is an 8-bit register. Execution of this instruction requires four memory cycles: fetch the 8-bit instruction, obtain the high byte of the operand address, obtain the low byte of the operand address, and obtain the data from the operand address. The approximate register-transfer-level steps are given in Table 10.9-2 and are explained next.

The first step moves contents of the program counter (PC) to the address bus (AB) in preparation for fetching the instruction byte. While the processor is waiting for the memory to respond, the program counter is incremented. After a delay time, the DBI (data bus input) is moved to the instruction register (IR). This ends the first memory cycle. As the instruction is being decoded, the incremented PC is moved to the address bus in preparation to fetch the next byte. The PC is incremented again, and the contents of the DBI are moved to the internal data bus (DB) and on to a temporary register (TMP) to complete the second cycle. To begin the third memory cycle, the previously incremented PC is moved to the AB and the PC value is incremented. The DBI contents are moved to DB where they are held in preparation for the cycle that outputs the data address (this is slightly oversimplified). The fourth and final cycle moves the data from DB to the low-order bits of the address bus (ABL) and the contents of TMP to the high-order bits of the address bus (ABH). At this point, the extended 16-bit address of

TABLE 10.9-2
Microprocessor Instruction Execution

Cycle	Operation	Explanation
1	AB ← PC	pc to address bus
	PC ← PC + 1	incr pc
	IR ← DBI	data to ir
2	AB ← PC	pc to address bus
	PC ← PC + 1	incr pc
	TMP ← DB ← DBI	data to tmp
3	AB ← PC	pc to address bus
	PC ← PC + 1	incr pc
	DB ← DBI	data to dynamic store
4	ABL ← DB	data adr to address bus
	ABH ← TMP	data adr to address bus
	ACCA ← DB ← DBI	data to accumulator

the data is present on the address bus. The memory responds with the requested data, and this data is moved from DBI to DB and into accumulator A (ACCA) to complete execution of the instruction. These RTL statements describe at a high level the execution of a simple microprocessor instruction.

ISPS Specification and Simulation

The Instruction Set Processor Specification (ISPS) language was developed for the certification, architectural evaluation, simulation, fault analysis, and design automation of instruction set processors. The language provides a behavioral rather than a structural description. There are no part numbers, pin assignments, layouts, or technologies defined. Of course, some structural information such as register lengths, data path widths, and connections of components are necessarily a part of the simulation. The operation of each part of a processor is specified algorithmically by its behavior.

The ISPS notation includes an interface and entities. First, the carriers (memory) elements are defined. This usually includes an array of memory locations with a specified bit width and number of words. Second, the procedures necessary for the execution of the processor statements are defined. This usually includes instruction decoding, effective address calculation, arithmetic and logical operation definitions, and memory load/store functions. ISPS provides a typical set of program operators, including assignment, if, case, and repeat. Additionally, provisions are made for concurrent or sequential processing. It is possible to specify the bit length of words. Aliases are available for variables, and bit fields of variables can be addressed directly by other variables. Normal number representations include binary, hex, decimal, and octal. An example will be presented to demonstrate briefly some of the capabilities of the ISPS language.

The Motorola 68000 microprocessor will be used as the example to describe typical ISPS capabilities. Figure 10.9-1 shows the definition of the memory and processor state. The memory is defined here as 1 K 16-bit words with the name M and the alias Memory. The processor state includes definition of the program counter (PC) and extended program counter (PCA), the register array (REG), the instruction register (IR), and other required processor state holders. In each case, the number of registers and the width in bits are defined. Multiple references to some resources are specified. For example, an array of sixteen 32-bit registers (REG) is defined. Then the data registers (D) are specified as the first eight registers, and the address registers (A) are specified as the second eight registers of the register array.

HARDWARE DESIGN LANGUAGES

Machine-readable descriptions of integrated circuit designs have become an important factor in designing VLSI circuits. These descriptions are often defined in terms of design languages that, like computer languages, have specific syntax and semantics. Such design languages have been used to describe circuits from the geometrical level up through the architectural level. As new designs become increasingly dependent on CAD tools, machine-readable descriptions become extremely important. Two hardware design languages have evolved as ANSI (American National Standards Institute) standards within the last few years. One of these, EDIF (Electronic Design Interchange Format), is intended to describe designs from the layout level through the logic level. Another such language, VHDL (VHSIC Hardware Description Language), is used to characterize both the function and structure of designs from logical primitives through architectural descriptions. The basics of these two languages will be introduced here along with simple examples of each.

EDIF Design Description

As integrated circuit designs increased in complexity and the use of computers became prominent within the semiconductor industry, the need for a common interchange format for integrated circuit design information arose. With such a standard, silicon foundries could accept design descriptions from many sources, CAD vendors could create widely applicable programs to process designs, and designers would benefit from wider availability of CAD tools and silicon processing. The EDIF (Electronic Design Interchange Format) standard was created by interested companies to fulfill this need.

Key elements in the design of the EDIF language were broad applicability and easy extensibility. To meet these goals, EDIF was designed with a syntax that is similar to LISP with all data represented as symbolic expressions. Primitive data such as strings, signals, ports, layers, numbers, and identifiers are the *atoms* of EDIF. These atoms are formed into more complex structures as *lists*; many times, the first element of a list is a keyword that gives a particular meaning to the subsequent elements of the list. This syntax is easily parsed, and the keywords—not the syntax—provide the semantics of the language. Thus, it is desirable to design EDIF parsers that respond to the particular set of keywords for their intended function. Unrecognized keywords may be ignored successfully, allowing upward compatibility with new extensions of the language.

EDIF is intended neither as a programming language nor a database language, but rather as an efficient interchange format for integrated circuit designs. The LISP-like structure is relatively compact and yet maintains a textlike property that allows it to be read and written directly by humans. An EDIF description may contain mask descriptions, technology information, net lists, test instructions, documentation, and other user-defined information. The structure is hierarchical in that larger design descriptions can be built from component descriptions and libraries of standard elements.

The basic organizational entity for describing designs within EDIF is the *cell*. A cell may contain different representations or *views* of a design. For example, one view might contain mask layout information while another view may contain behavioral-level modeling information. A view may be one of several types such as *physical*, *document*, *behavior*, *topology*, or *stranger*. Each view will contain a specific type of information about the cell. For example, the physical view may contain geometric figures for circuit schematics or mask artwork, but it will not contain behavioral information. The topology view might contain net list descriptions, schematic diagrams, or symbolic layout. The document view could contain a textual description of a design, figures describing the design, or

specifications for the behavior of the design. The stranger view is provided for data that does not meet the conventions of the other view types.

Each view of a cell may specify its *interface* to the external world. This interface includes a list of external ports and their characteristics. The interface description does not specify how the cell performs its function internally but rather defines how the cell will relate to its environment. A second part of the cell definition is its *contents*. The contents provide the detailed implementation for each view. This could include instances of other cells or could be the actual definition of mask geometry for the cell layout. A net list view and a mask layout view for a full adder are described here as examples of EDIF contents.

VHDL Design Description

VHDL was developed for the design, description, and simulation of VHSIC components. VHSIC is the acronym for the Very High Speed Integrated Circuits program of the U.S. Department of Defense. Thus, the language was originally developed to describe hardware designs for military purposes. Because the need for a standard hardware description language is industrywide, the VHDL language was adopted by the IEEE and formalized as an industry standard.

VHDL is concerned primarily with description of the functional operation and/or the logical organization of designs.[24] This description is accomplished by first specifying the inputs and outputs of a system or device. Then either its *behavior* (outputs as functions of inputs) or its *structure* (in terms of interconnected subcomponents) is specified. The primary abstraction in VHDL is called a *design entity*. A design entity has two parts: the *interface description* and one or more *body descriptions*.

An interface description must perform several functions. It must define the logical interface to the outside world. It must specify the input and output ports and their characteristics. Additionally, operating conditions and characteristics may be included. To accomplish this, the interface description provides a *port declaration* for each input and output of the design entity. Each port declaration includes a port *name* and an associated *mode* and *type*. The mode specifies direction as *in, out, inout, buffer*, or *linkage*. The type qualifies the data that flows through a port. Standard types include *BIT, INTEGER, REAL, CHARACTER*, and *BIT_VECTOR*. Additionally, user-defined types are acceptable.

ALGORITHMIC LAYOUT GENERATION

Algorithmic generation of integrated circuit layout is often perceived as a solution to the VLSI complexity problem. The basis of this well-known problem is that integrated circuit design cost is increasing for complex chips while the product life cycle is decreasing for these same chips. Design cost increases because of the design time and computer resources that must be expended to complete a state-of-the-art chip or system. Product life cycle is decreasing for these same designs because of rapid advances in technology and fierce competition to get the next-generation product to the market first.

Three approaches have been suggested to address this problem.[25] The first approach is to enhance the productivity of the human designer with faster computer workstations and improved design analysis tools. To date, this approach has been the most evident, and its description comprises the bulk of the topics in this chapter. A second approach is to capture the knowledge of a human designer with an expert system. This involves a knowledge base of concepts, rules, and strategies. These are processed by an inference engine that produces design fragments and design refinements to aid the design process. This approach is a subject of active research. A third approach is to algorithmically generate or synthesize designs from high-level descriptions or from parameterized definitions. Each variant of this approach tends to concentrate on a particular target architecture. For example, the PLA generators discussed earlier accept Boolean equations and generate layout in a well-defined form. More complex algorithmic generators are often termed *silicon compilers*. This section describes two pioneering efforts in this area and follows with a description of a state-of-the-art microprocessor chip set that was designed with heavy dependence on a commercial silicon compiler.

Bristle Blocks Silicon Compiler

The Bristle Blocks silicon compiler was first described in 1979.[26] The goal of the Bristle Blocks system was to produce a layout mask set from a single-page, high-level description of an integrated circuit. Many designs have their high-level structure and function frozen early in the design cycle, before the effects of such decisions are well known. If, on the other hand, a designer could use a few building blocks, organize them, and then obtain complete mask layouts and simulations early in the design cycle, then experimental configurations could be tried with a minimum of effort.

The fundamental unit in the Bristle Blocks system is the cell. Each cell can contain geometrical primitives and references to other cells. A cell can be compared to an HLL (high-level language) subroutine that contains some primitive operations and contains some references to other subroutines. A cell has the capability of containing each of the seven representations just presented. Each cell contains only local information. External connections are specified by their location and type. The location indicates where along the cell boundary the connection should occur, and the type specifies the kind of connection—for example, external output pad. The Bristle Blocks methodology gets its name from the connection points, which are like bristles along the edges of the cells. A primary directive of this method is that local information is kept local to the cell, while global information such as the location and routing to an external pad is kept separately.

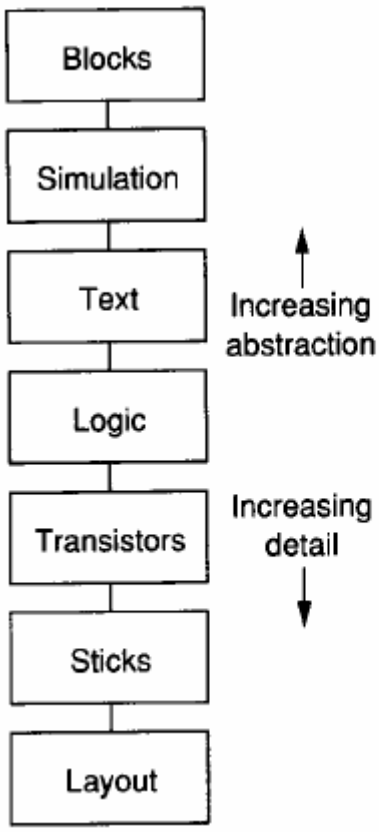


FIGURE 10.11-1
Bristle Blocks design abstraction hierarchy.

Information specifying the various representations of cells is kept in cell libraries and is accessed as needed. Each low-level cell must have been designed before it can be used in the Bristle Blocks system. Each such cell is defined by specifying the actual layout of the cell. It is felt that design of low-level cells does not take much time because of their small size. Also, the design is relatively error-free, and designer ingenuity is most beneficial at this design level.

MacPitts Silicon Compiler

A flexible register-transfer-type language called MacPitts was described in 1982 to address the generation of microprogram-sequenced data path designs.[27] Designs described in this high-level language are compiled into a technology-independent intermediate form. The intermediate form is then compiled into a CIF geometrical layout description, which can be submitted to a silicon foundry for fabrication. The latter compilation is accomplished by limiting the possible degrees of freedom in mask layout and restricting the layout to a fixed target architecture. The target architecture consists of two distinct sections: a data path and a control unit. This architecture is shown in Fig. 10.11-3.

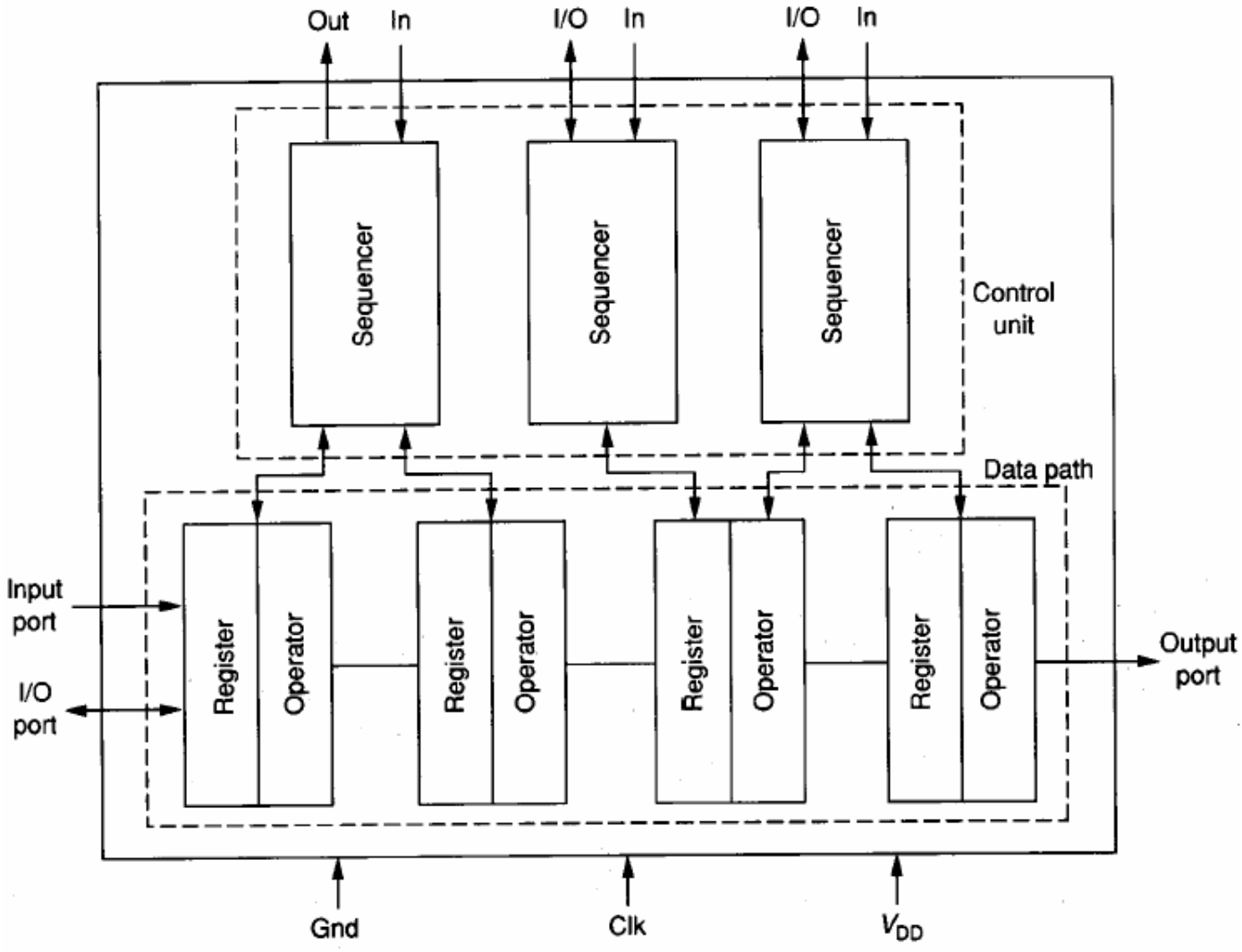


FIGURE 10.11-3
MacPitts data path/control architecture.

The data path consists of registers of width specified by the MacPitts source program. Operators for testing and modifying the data stored in the registers are also created. Data is communicated to the external world through parallel buses of wires called ports. A particular port can be an input port, a tri-state port, or an I/O port. The operations performed by the data path are specified by the control unit. In general, the control unit generates signals that cause the data path to perform certain operations. The data path returns signals that can be used to alter the control sequence. In addition, the control unit communicates to the external world through single-wire signals that may be input, output, tri-state, or I/O lines.

The data path is unconventional because it contains more than just a register array and an ALU, as is common in many microprocessors. Rather, the data path may contain many functional units interspersed among the registers. As many functional units as are needed to compute a set of parallel operations may be included between global buses. The functional units are interconnected by dedicated local buses as required by the function they perform. A given unit may take its input from several possible sources, so a multiplexer is often included to select the particular input for an operation. The output of the units is either a full word used by the data path or possibly a test result that is used directly by the control unit. A unit like an adder can generate both a word (sum) for the data path and a test signal (overflow) for the control unit. The number and type of register/operator units provided in the data path differ from system to system as specified by the MacPitts source language.

The control unit is implemented as a simplified variation of a finite-state machine. A typical FSM consists of combinatorial logic and a state register; the combinatorial logic computes the output signals and the next-state information. If the program flow is sequential, this general form of FSM is less efficient than simply using a counter to present the next state. The MacPitts compiler generates a FSM consisting of a counter and a state stack to allow subroutine calls. The logic portion of the control unit is implemented by a Weinberger array layout style consisting of interconnected NOR gates. This regular form for logic allows multilevel realizations of logic within the control unit for increased efficiency compared with a PLA-style implementation.