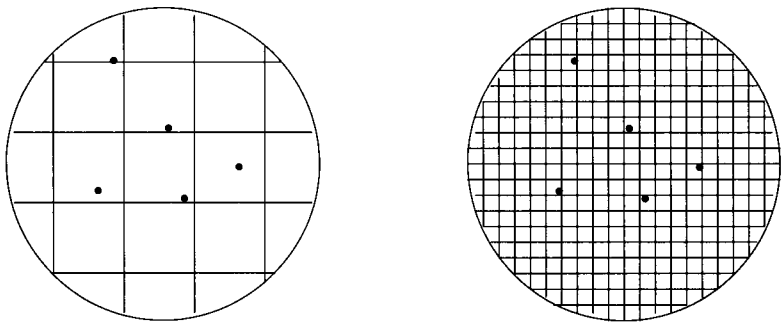
Chapter

15

VLSI Layout Examples

In the past chapters we have concentrated on basic logic-gate design and layout. In this chapter we discuss the implementation of logic functions on a chip where the size and organization of the layouts are important. The number of MOSFETs on a chip, depending on the application, can range from tens (an op-amp) to more than hundreds of millions (a 256 Mbit DRAM). Designs where thousands of MOSFETs or more are integrated on a single die are termed *very-large-scale-integration* (VLSI) designs.

To help us understand why chip size is important, examine Fig. 15.1. The dark dots indicate defects and thus bad chips. Figure 15.1a shows a wafer with nine full die. The partial die around the edge of the wafer are wasted. Five of the nine die do not contain a defect and thus can be packaged and sold. Next consider a reduction in the die size (Fig. 15.1b). We are assuming each die, whether discussing the die of Fig. 15.1a or b, performs the same function. This reduction can be the result of having better layout (resulting in a smaller layout area) or fabricating the chips in a process with smaller



(a) (b)

Figure 15.1 Defect density effects on yield.

device dimensions (e.g., going from a 130 nm process to a 50 nm process). The total number of die lost (see Fig. 15.1b), due to defects is five; however, the number of good die is significantly larger than the five good die of Fig. 15.1a. The yield (number of good die/total number of die on the wafer) is increased with smaller die size. The result is more die/wafer available for sale. Another benefit of reducing die size comes from the realization that processing costs per wafer are relatively constant. Increasing the number of die on a wafer decreases the cost per die.

15.1 Chip Layout

VLSI designs can be implemented using many different techniques including gate-arrays, standard-cells, and full-custom design. Because designs based on gate-arrays are used, in general, where low volume and fast turnaround time are required and the chip designers need know little to nothing[1] about the actual implementation of the CMOS circuits, we will concentrate on full-custom design and design using standard cells

Regularity

An important consideration when implementing a VLSI chip design is regularity. The layout should be an orderly arrangement of cells. Toward this goal, the first step in designing a chip is drawing up a chip (or section of the chip) floor plan. Figure 15.2 shows a simple floor plan for an adder data-path. This floor plan can be added to the floor plan of an overall chip, which includes output buffers, control logic, and memory. At this point, we may ask the question, "How do we determine the size of the blocks in Fig. 15.2?" The answer to this question leads us into the design and layout of the cells used to implement each of the logic blocks in Fig. 15.2.

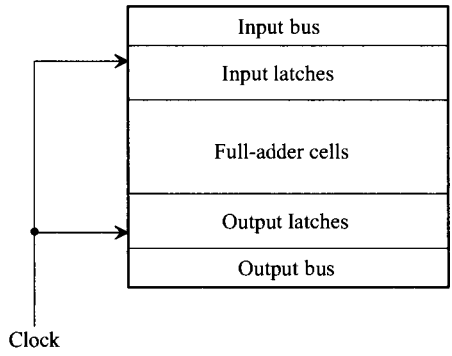


Figure 15.2 Floor plan for an adder.

[1] At many universities, design using a hardware description language (HDL) with field-programmable-gate arrays (FPGAs) is discussed in the first (or perhaps second) course on digital systems design.

Standard-Cell Examples

Standard cells are layouts of logic elements including gates, flip-flops, and ALU functions that are available in a cell library for use in the design of a chip. *Custom design* refers to the design of cells or standard cells using MOSFETs at the lowest level. *Standard-cell design* refers to design using standard cells; that is, the designer connects wires between standard cells to create a circuit or system. The difference between the two types of design can be illustrated using a printed circuit board-level analogy. A standard-cell design is analogous to designing with packaged parts. The design is accomplished by connecting wires between the pins of the packaged parts. Custom design is analogous to designing the "insides" of the packaged parts themselves

Figure 15.3 shows an example of an inverter. In addition to keeping the layout size as small as possible, an important consideration when laying out a standard cell is the routing of signals. Keeping this in mind, we can state the following general guidelines for standard-cell design:

1. Cell inputs and outputs should be available, at the same relative horizontal distance, on the top and bottom of the cell.

2. Horizontal runs of metal are used to supply power and ground to the cell, a.k.a., power and ground buses. Also, well and substrate tie downs should be under these buses.

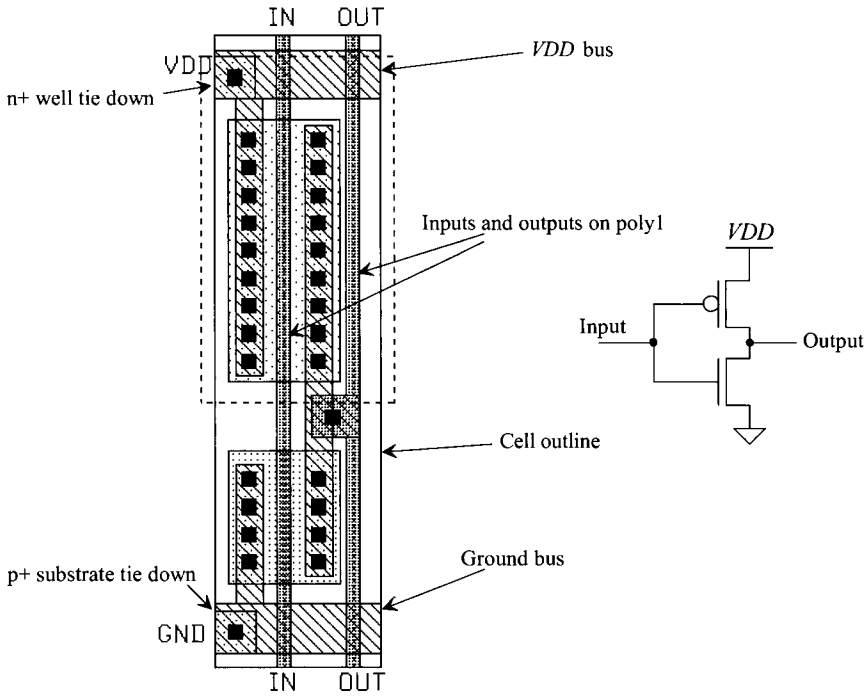


Figure 15.3 Standard cell layout of an inverter.

3. The height of the cells should be a constant, so that when the standard cells are placed end to end the power and ground buses line up. The width of the cell should be as narrow as the layout will allow. However, the absolute width is not important and can be increased as needed.

4. The layout should be labeled to indicate power, ground, and input and output connections. Also, an outline of the cell, useful in alignment, should be added to the cell layout.

Figure 15.4 illustrates the connection of standard cells to a bus. Note that poly, which runs vertically, can cross the metal1 lines, which run horizontally without making contact. This fact is used to route signals and interconnect standard cells in a VLSI design. Also, in this figure, note how the two inverter standard cells are placed end to end. The result is that power and ground are automatically routed to each cell.

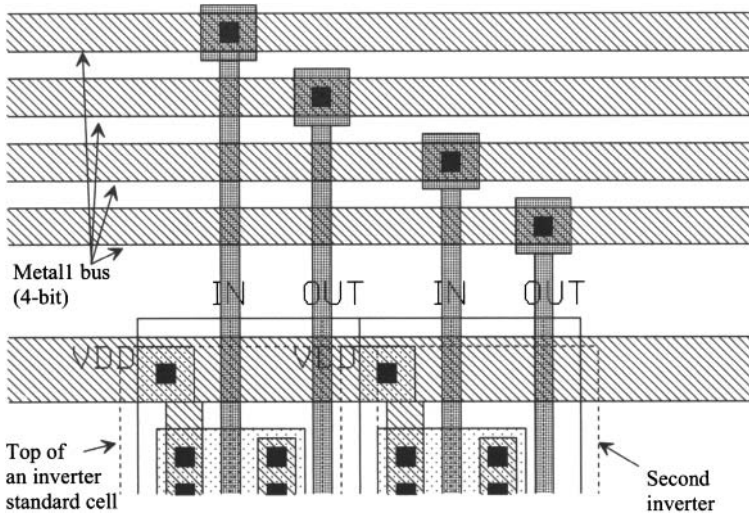


Figure 15.4 Connection of two inverter standard cells to a bus.

Other examples of static standard cells are shown in Fig. 15.5. A double inverter standard cell is shown in Fig. 15.5a, while NAND, NOR, and transmission gate standard cells are shown in Figs. 15.5b, c, and d.

Figure 15.6 shows the layout of a NAND-based SR latch. This layout differs from the others we have discussed. In all layouts discussed so far metal1 and contacts are adjacent to the gate poly. Also, the gate poly has been laid down without bends. The expanded view of a PMOS device used in the SR latch is shown in Fig. 15.7. Keeping in mind that whenever poly crosses active (n+ or p+), a MOSFET is formed, we see that the source of the MOSFET is connected to metal through two contacts, while the p+ implant forms a resistive connection to metal1 along the remainder of the device. The layout size,

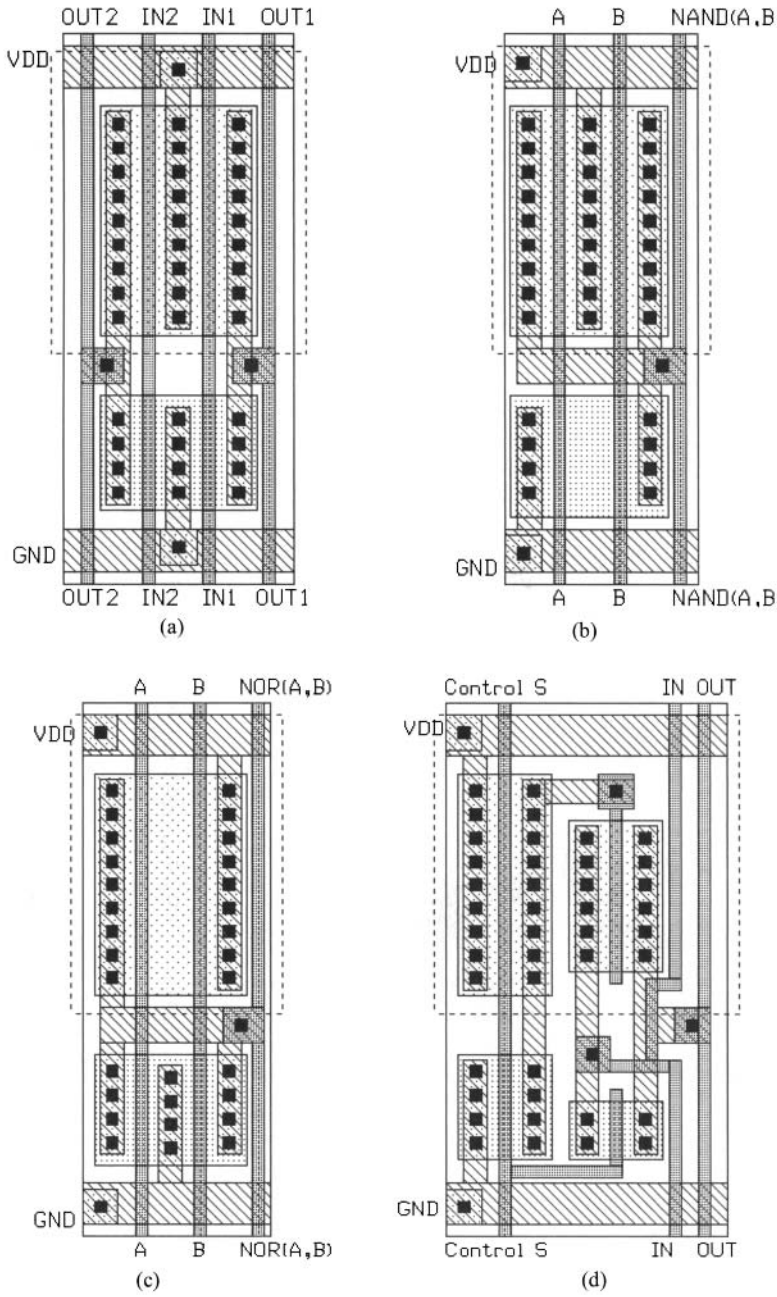


Figure 15.5 (a) Double inverter, (b) two-input NAND, (c) two-input NOR, and (d) transmission gate.

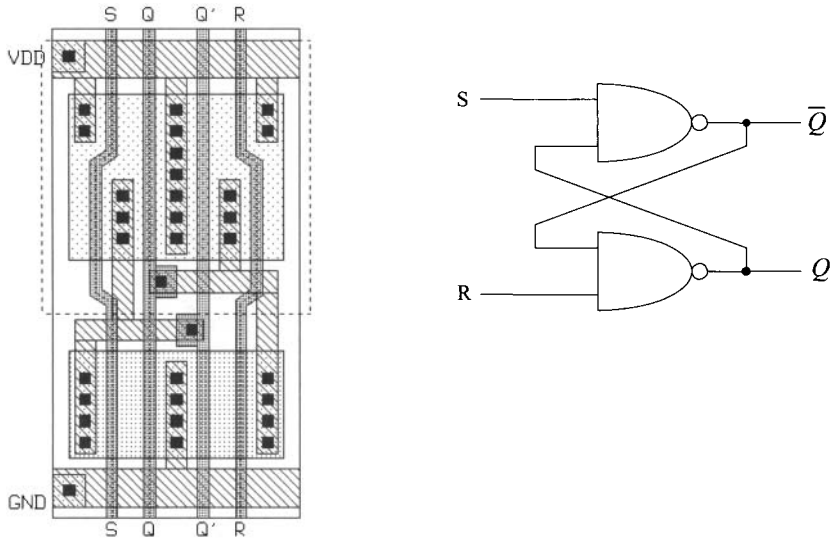


Figure 15.6 SR latch using NAND gates.

in this case the width of the standard cell, can be reduced using this technique. Because of the bend in the gate, the width of this MOSFET is longer than the adjacent MOSFET. This additional width is of little importance and has little effect on the DC and transient properties of the gate. Figure 15.8 shows the NOR implementation of an SR latch.

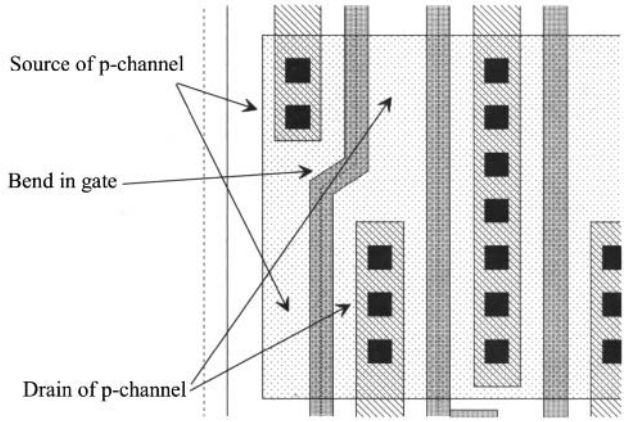


Figure 15.7 Section of the layout shown in Fig. 15.6.

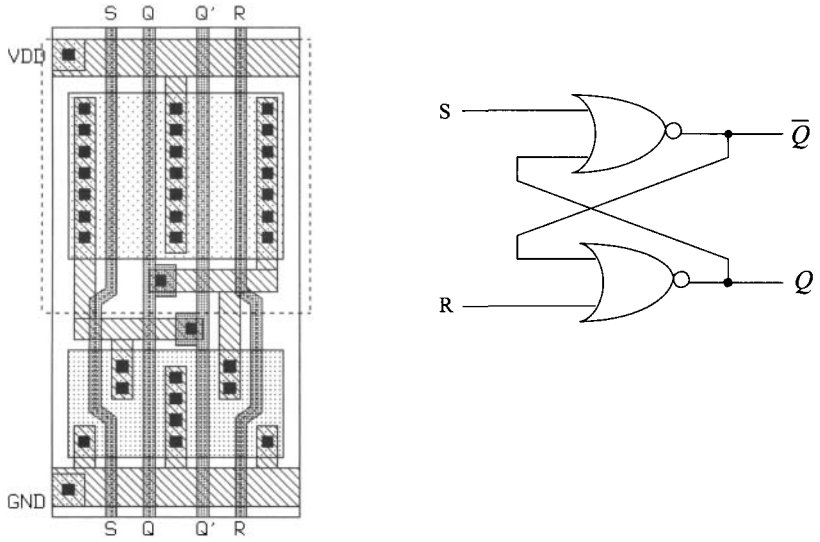


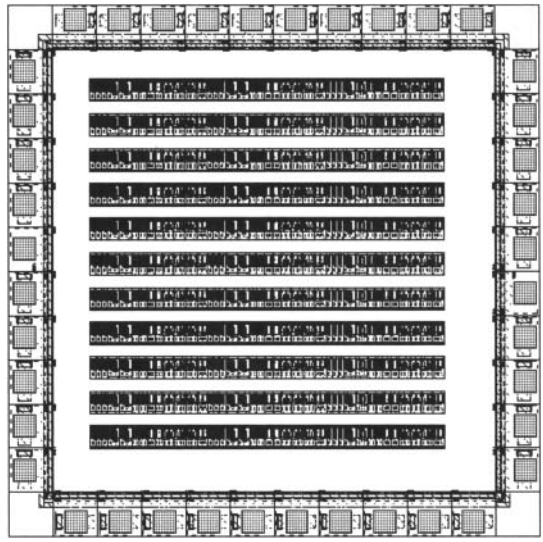
Figure 15.8 SR latch using NOR gates.

Power and Ground Connections

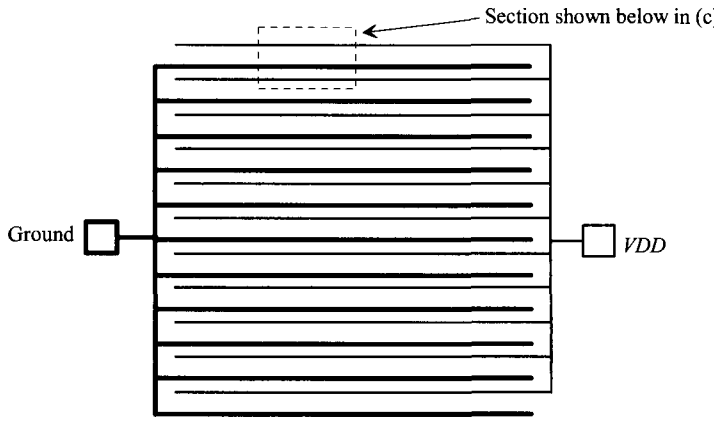
Many of the problems encountered when designing a chip can be related to the distribution of power and ground. When power and ground are not distributed properly, noise can be coupled from one circuit onto the power and ground conductors and injected into some other circuit.

Consider the placement of standard cells in a padframe shown in Fig. 15.9a, without connections to power and ground. Approximately 600 standard cells are shown in this figure. The space between the rows of standard cells is used for the routing of signals. A line drawing of a possible power and ground busing architecture is shown in Fig. 15.9b. Consider the section of bus shown in Fig. 15.9c. Wire A connects the standard cells in the top row to *VDD*, while wire B is used for the connection to ground. Ideally, the current supplied on A (*VDD*) is returned on B (ground). In practice, there is coupling between conductors B and C, which gives rise to an unwanted signal (noise) on either conductor. This coupling can be reduced by increasing the space between B and C, which reduces the inductive and capacitive coupling between the conductors. Another solution is to increase the capacitance between A and B. A standard cell decoupling capacitor (Fig. 15.10) can be used toward this goal. The capacitor is placed in the middle of a standard-cell row. Also, the AC resistive drop effects discussed in Ch. 3 (see Fig. 3.17 and the associated discussion) are greatly reduced by including this capacitor.

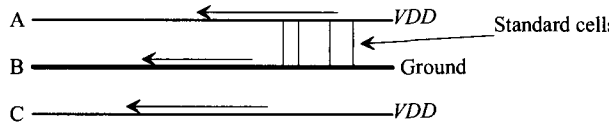
Coupling is a problem on signal buses as well. Figure 15.11 shows a simple scheme to reduce coupling. The length of a section, where two wires are adjacent, is reduced by routing the wire to other locations at varying distances along the bus. The inductive or capacitive coupling between two conductors is directly related to the length of the wire runs.



(a)



(b)



(c)

Figure 15.9 Connection of power and ground to standard cells.

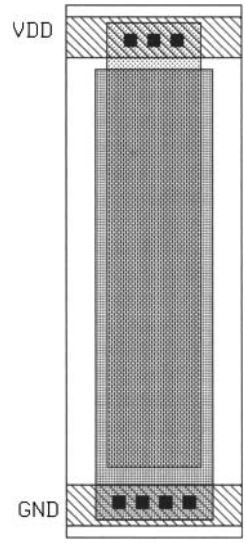


Figure 15.10 Decoupling capacitor.

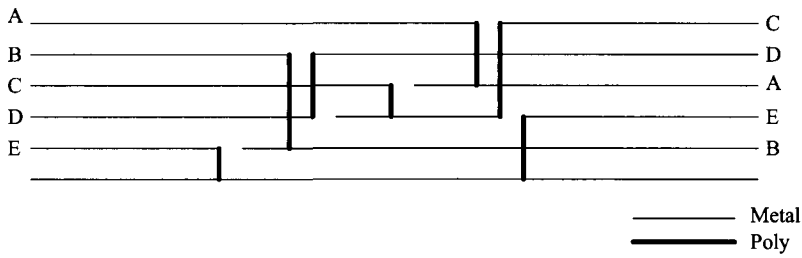


Figure 15.11 Busing structure used to decrease signal coupling.

An Adder Example

As another example, let's consider the implementation of a 4-bit adder. (The floorplan for this adder was shown in Fig. 15.2.) The first components that must be designed are the input and output latches. Figure 15.12a shows the schematic of the latches. This latch is the level-sensitive type discussed in Ch. 13. When CLK is high, the output, Q, changes states with the input, D. The inverter, I4, provides positive feedback and is sized with a small W/L ratio so that I1 does not need to supply a large amount of DC current to force the latch to change states. The layout of the latch is shown in Fig. 15.12b. The layout size and the size of the MOSFETs in these examples may be larger than normal to make it easier to understand and view the layouts.

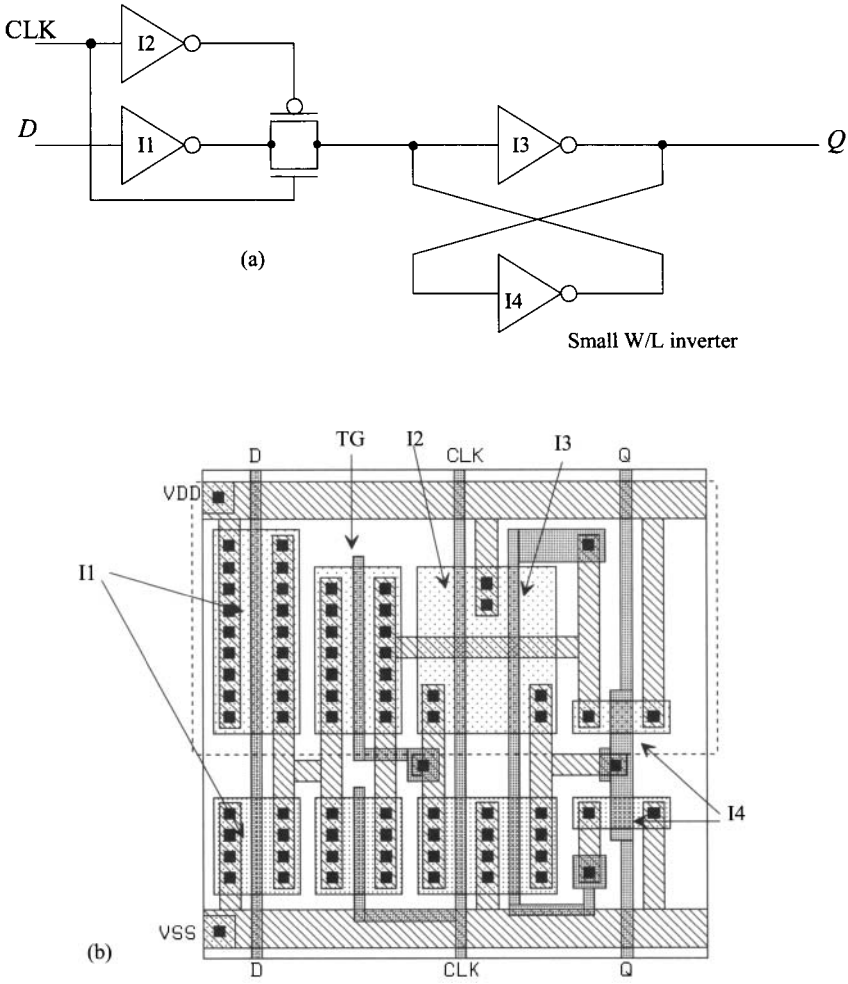


Figure 15.12 Schematic and layout of a latch.

The layout of the static adder is shown in Fig. 15.13. This is the implementation, using near minimum-size MOSFETs, of an AOI (and-or-inverter) static adder. Both the carry-out and sum-out logic functions are implemented in this cell.

The complete layout of the adder is shown in Fig. 15.14. The two 4-bit words, Word-A and Word-B, are input to the adder on the input bus. These data are clocked into the input latch when CLK is high, while the results of the addition are clocked into the output latch when CLK is low. The inverter standard cell of Fig. 15.3 is placed at the end of the output latches and generates $\overline{CLK}$ for use in the output latches. The inputs and outputs of the adder cells are run on poly because of the short distances involved. The carry-in of the adders is connected to ground, as shown in the figure.

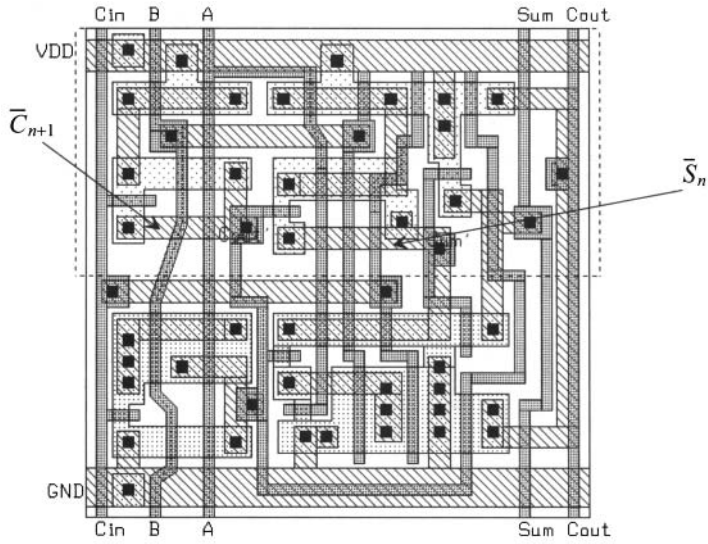


Figure 15.13 Layout of an AOI static adder.

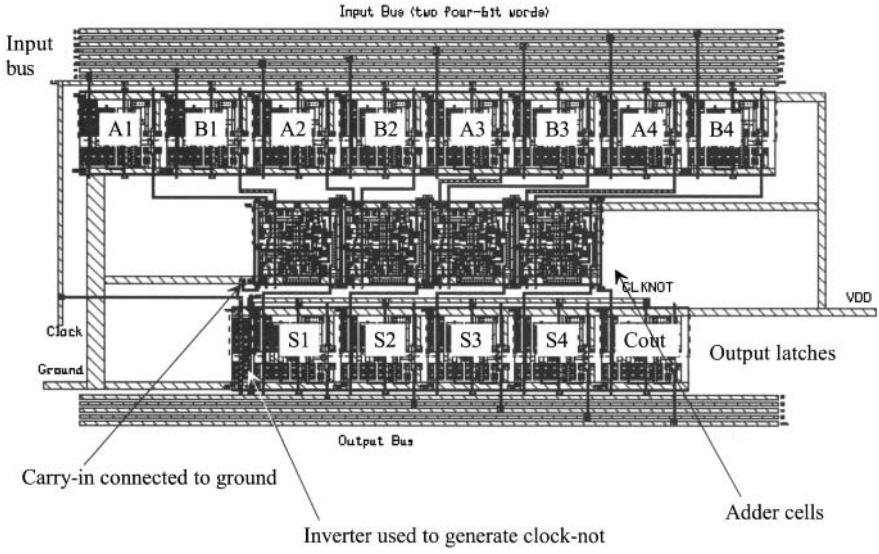


Figure 15.14 Layout of the complete adder.

A 4-to-1 MUX/DEMUX

The layout of a 4-to-1 MUX/DEMUX is shown in Fig. 15.15 (based on the use of NMOS pass gates). Notice that the (required) p+ substrate connections are not shown. This layout is different from the layouts discussed so far as the circuit does not require power and ground connections and the input/output signals are connected on n+. The select signals are supplied to the circuit on metal1 at the top of the layout. For A to be connected to the output, the signals S1 and S2 should be high. For a large MUX, the propagation delay through the n+ should be considered.

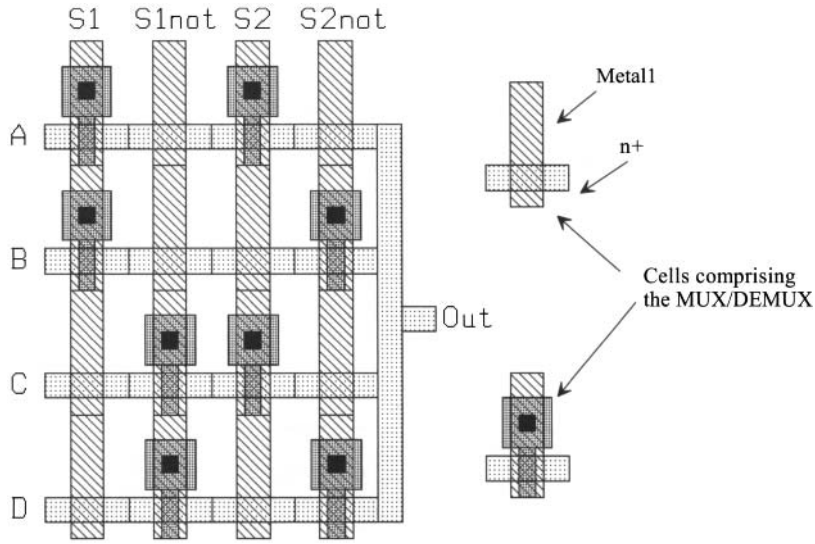


Figure 15.15 Layout of a 4-to-1 MUX/DEMUX.

15.2 Layout Steps *by Dean Moriarty*

The steps involved in rendering a schematic diagram into its physical layout are plan, place, connect, polish, and verify. Let's illustrate each of these steps in some detail.

Planning and Stick Diagrams

The planning steps start with paper and pencil. Colored pencils are useful for distinguishing one object from another. You can use graph paper to help achieve a sense of proportion in the cell plan but don't get too bogged down in the details of design rules or line widths at this point; we just want to come up with a general plan. A "stick diagram" is a paper and pencil tool that you can use to plan the layout of a cell. The stick diagram resembles the actual layout but uses "sticks" or lines to represent the devices and conductors. When used thoughtfully, it can reveal any special hook-up problems early in the layout, and you can then resolve them without wasting any time.

Figure 15.16a shows the schematic of an inverter. To realize the layout of this circuit, it is first necessary to define the direction and metallization of the power supply, ground, input, and output. Since the standard-cell template was discussed earlier (see also Fig. 4.15 and the associated discussion), we'll use it. Power and ground run horizontally and provide the substrate and well connections. The input and output are accessible from the top or bottom of the cell and will be in metal2 running vertically. Figure 15.16b shows the completed stick diagram. Note the use of "X" and "O" to denote contacts and vias, respectively. The stick diagram should be compared to the resulting layout of Fig. 15.17.

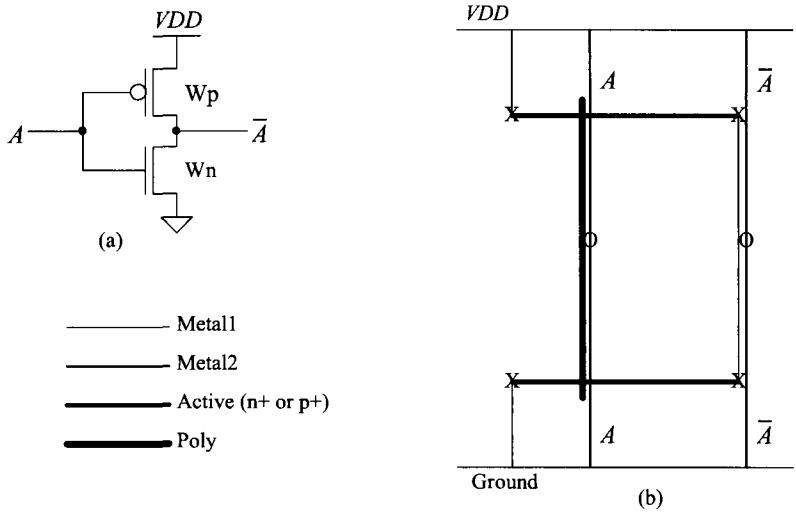


Figure 15.16 (a) Inverter and (b) stick diagram used for layout.

Suppose the device widths of the inverter circuit in Figure 15.16a were quadrupled. Furthermore, let's assume that the maximum recommended poly1 gate width is 20 (due to the sheet resistance of the poly) and that exceeding that maximum could introduce significant unwanted RC delays. Let's also suppose that we are to optimize the layout for size and speed (as most digital circuits are). To meet these criteria, it will be necessary to split transistors M1 and M2 in half and lay them out as two parallel "stripes." Figures 15.18a–d show the schematics, stick diagram, and layout for this scenario. The output node (drain of M1 and M2) is shared between the stripes so as to minimize the output capacitance. Taking the output in metal2 also helps in this regard. Notice that the stick diagram for this circuit looks like the previous inverter plus its mirror image along the output node. Also observe that the layout of this inverter is mirrored, as shown in the stick diagram. This is a common layout technique.

Figure 15.19 shows stick diagrams and layouts for two more common circuits: the two-input NAND and the two-input NOR. Compare the stick diagrams of Figs. 15.19a and c to the layouts of Figs. 15.19b and d. Observe that the output nodes share the active area just as in the previous example. Also note that the spacing between the gates of the series-connected devices is minimal.

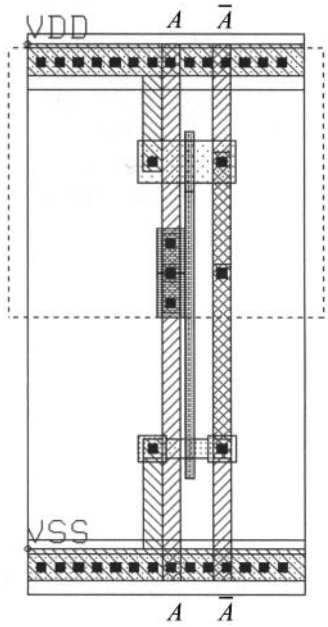
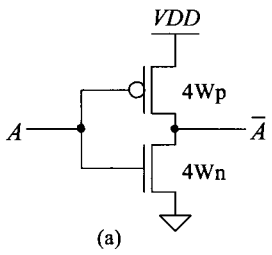


Figure 15.17 Layout of the inverter shown in Fig. 15.16.

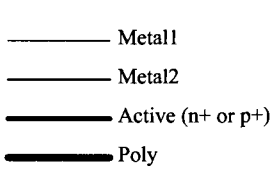
Take another look at the two circuits from a geometrical rather than an electrical viewpoint. Compare the NAND gate layout to the NOR gate layout. Each can be created from the other by simply "flipping" the metal and poly connections about the x-axis.

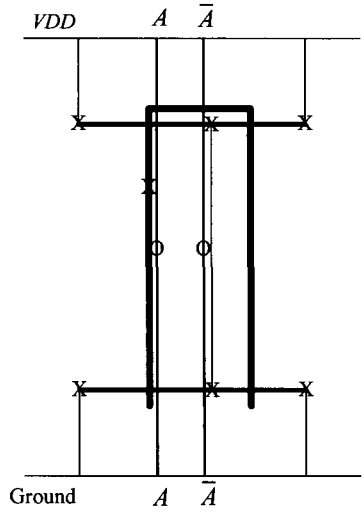
Device Placement

Figure 15.20 shows the schematic of a dynamic register cell, while Figs. 15.21 a–c show the stick diagrams and layout for a dynamic register. Compare the schematic of Fig. 15.20 to the stick diagram of Fig. 15.21a. We have labeled this stick diagram "preliminary" for reasons that will soon become apparent. Notice that there is a break or gap in the active area, which will form our NMOS devices. Also note that the clock signals CLK and $\overline{\text{CLK}}$ must be "cross-connected" from one side of the layout to the other. We don't have to think this through very far to notice that, with this placement of devices, hooking up the clock signals is going to be very difficult. Now look at the stick diagram shown in Fig 15.21b. Notice that we have rearranged the devices so that the active area is a continuous unbroken line. Normally, this "unbroken line" approach to device placement is preferred. It usually results in the most workable device placement. We say "usually" because at times your layout has to fit in an area defined by other blocks around it and you have no control over it. Also observe from Fig. 15.21b that the clock signal hook-up is more straightforward. Compare this stick diagram to the layout of Fig. 15.22c. Obviously, the device sizes used for this circuit are not practical; its purpose is merely to illustrate a layout concept. We can also see that the stick diagram is a useful tool throughout the layout process.

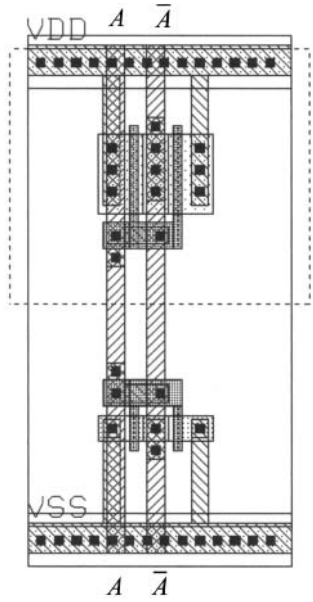


(a)

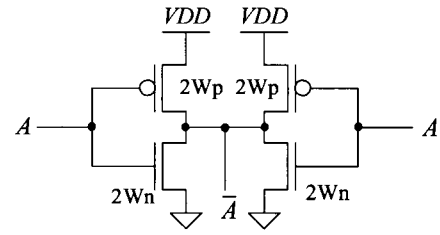




(b)



(c)



(d) Doubling the widths of the MOSFETs used in an inverter.

Figure 15.18 (a) Inverter, (b) stick diagram used for layout, (c) layout, and (d) equivalent schematic.

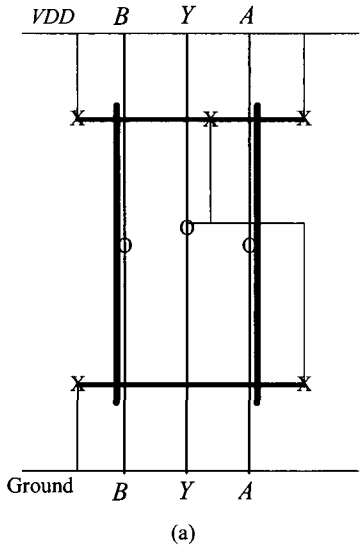


Figure 15.19 (a) NAND stick diagram, (b) layout, (c) NOR stick diagram, and (d) layout.

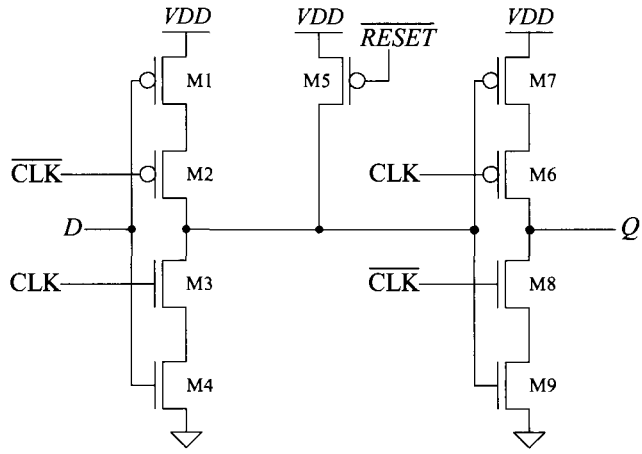


Figure 15.20 Schematic of a dynamic register cell.

Polish

After your layout is basically finished, it is time to step back and take a look at it from a purely aesthetic point of view. Is it pleasing to the eye? Is the hook-up as straightforward as possible, or is it "busy" and hard to follow? Are the spaces between poly gates and contacts minimum? What about the space between diffusions? Are there enough contacts? Did you share all of the source and drain implants that can be shared? Are there sufficient well and substrate ties? If you have planned well and followed the plan described here, you shouldn't run into too many problems.

Standard Cells Versus Full-Custom Layout

The standard cell approach to physical design usually dictates a fixed cell height and variable width when implementing the circuit. Furthermore, standard cells are designed to abut on two sides, usually left and right, and that abutment scheme must be quite regular so that any cell can reside next to any other cell without creating a design rule violation. The standard cell approach to layout is very useful and is always an excellent place to start. However, in the real world, area on a wafer translates directly into profit and loss (money). Wafer costs are relatively fixed whether they're blank or as tightly packed with circuitry as possible. Therefore, it follows that we want a layout that is as small as possible so that there can be as many die per wafer as possible. These are the economics of the situation. There are also technical advantages to be gained from having as small a layout as possible: interconnecting wires can be as short as possible, thereby reducing parasitic loading and crosstalk effects.

Figure 15.22 shows a typical standard-cell block that has been placed and routed by an automatic tool. Most of the individual cells have been omitted for clarity. Notice the interconnect channels between the rows of standard cells. Power, ground, and clock signal trunks run vertically to both sides of the block by means of a special cell called an "end cap." Cell rows are connected to power and ground through horizontal buses that are

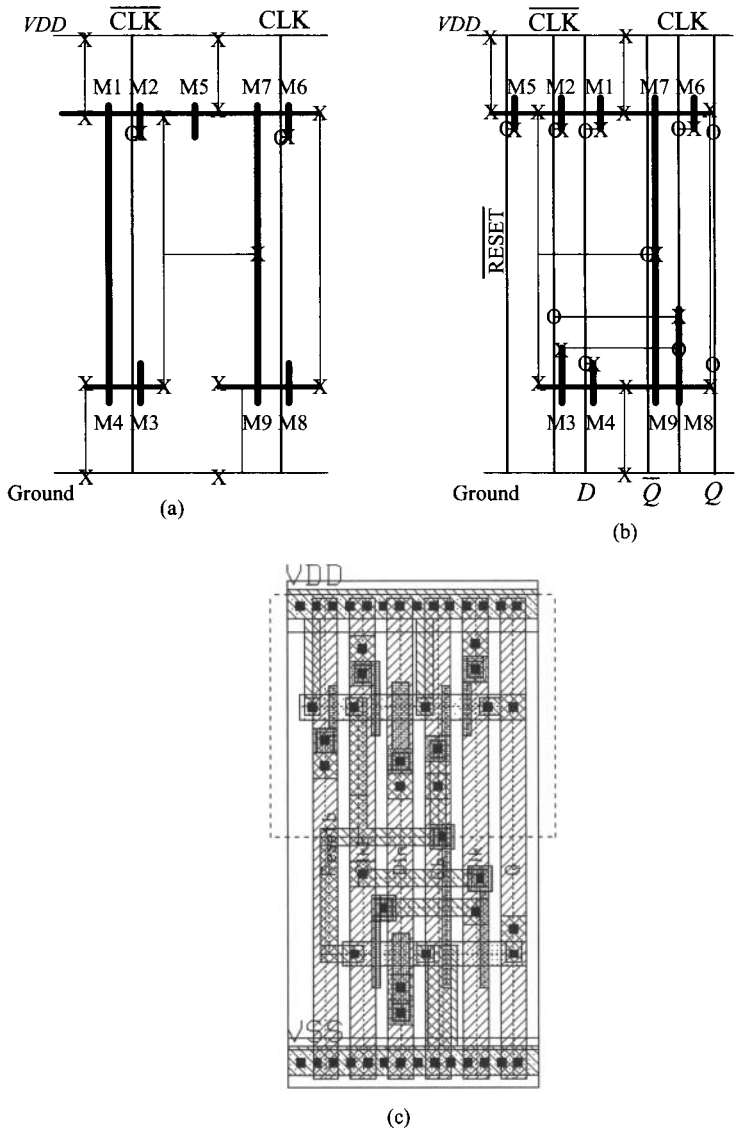


Figure 15.21 Layout of a dynamic register cell.

part of the standard cells themselves. All remaining connections are made via the routing channels. The standard-cell layouts are designed to accommodate metal2 feedthroughs that run vertically through each cell. The autorouter makes use of this space and adds the feedthroughs as needed to connect or pass signals from one routing channel to another. The routing channels and their associated interconnecting wires are the limiting factors for both the density and circuit performance of this type of layout.

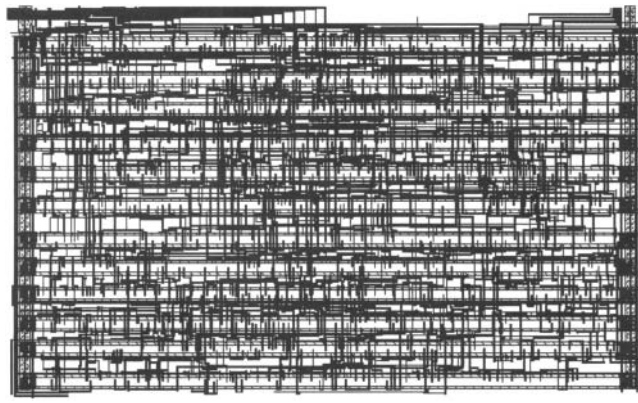


Figure 15.22 Layout based on standard cells.

Before we continue our discussion of relative layout density, we need to define a metric with which to quantify the matter. It is customary to use the number of transistors per square millimeter of area for this purpose. Because it is a raw number and the common denominator of all circuit layouts, we can use it even when comparing different types of circuitry or even unlike processes.

The density of the standard-cell route shown in Fig. 15.22 is approximately 5,000 transistors per square millimeter. This is fairly representative of the possible density for the channel-based routing approach and the process used (0.8 μm). Figure 15.23 shows a full-custom layout for a digital filter. The circuit area is approximately 2.1 square millimeters. The density is approximately 17,500 transistors per square millimeter, representing a 3.5-fold increase. This circuit, too, is fairly representative of the attainable density of full-custom layout using this particular 0.8 μm process. Both of these circuits were laid out using the same process, and in fact they are from the same die. The device sizes within each block would probably average out to minimum or close to minimum. The main difference affecting density is the interconnect wiring. This overhead associated with interconnect wiring is commonly referred to as the "interconnect burden." The designer must bear this burden in terms of both physical (wasted area) and electrical parameters (parasitic loading). Let us examine one method of creating a high-density custom layout that will minimize interconnect burden and circuit area.

Figures 15.24a–c show a small section of the interpolation filter from Fig. 15.23. In Fig. 15.24a we see an exploded view of four cells that form part of a data-path: an input data register, a t-gate, a full adder, and an output data register. These are instantiated (placed as a cell) twice, creating a view of eight cells. The two adder cells are slightly different: the carry inputs and outputs are on opposite sides, so that the carry-out can cascade to the carry-in of the next adder by abutting (placing next to one another) the cells. Unlike standard cells, the height and width constraints placed on custom layouts are contextual. In other words, a cell's aspect ratio depends on that of its neighbors. In this

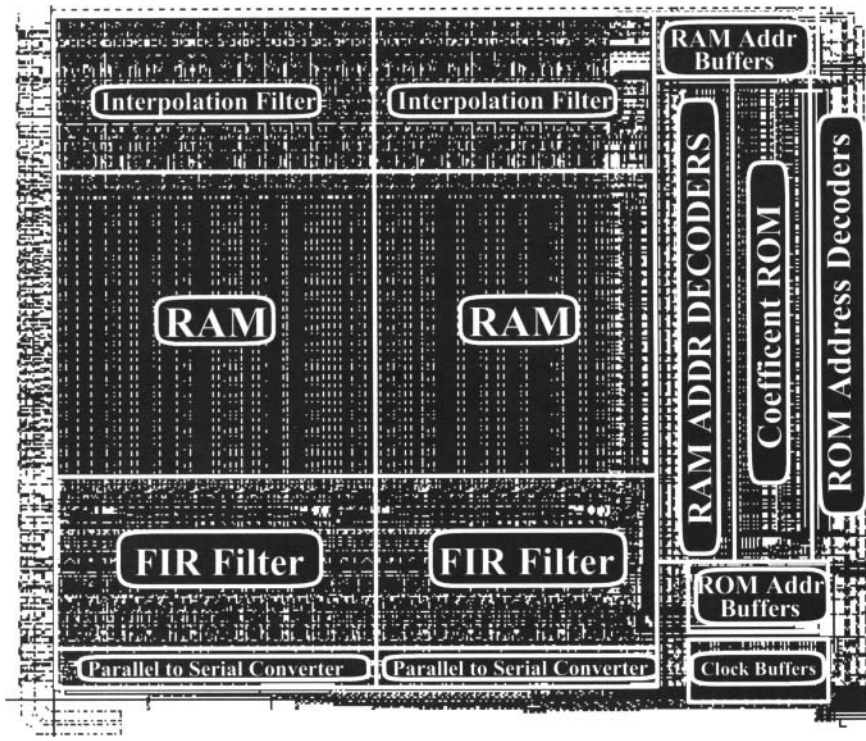


Figure 15.23 Full custom layout of a digital filter.

case, the width of each cell depended on the maximum allowable width of the widest cell in the group: the data register. Notice the top, bottom, left, and right boundaries of each cell in Fig. 15.24a. Data enter the register cell from the top and are output at the bottom. Clocks, power, ground, and control signals route across all the cells. The adder receives its A and B inputs from the top and outputs their SUM at the bottom. As already mentioned, carry-out and carry-in are available on the left and right edges of the adder, respectively. Figure 15.24b shows a 2-bit slice of this data-path with all of the connections made by cell abutment. Figure 15.24c illustrates how all four edges of each cell join together to complete the hook-up.

We have seen how circuits can be implemented by means of standard cells or custom layout. The time needed to produce a standard-cell route is far less than that of a full-custom implementation. The trade-offs are area and performance. Automatic place and route tools based on routing area rather than routing channels are now coming into use. These promise a compromise solution between the two extremes. The density of their results rivals that of full custom layout. Perhaps the hand-rendered, full custom layout will someday become a thing of the past. Nevertheless, process technology continues to advance, circuit designers continue to design circuits that test the outermost

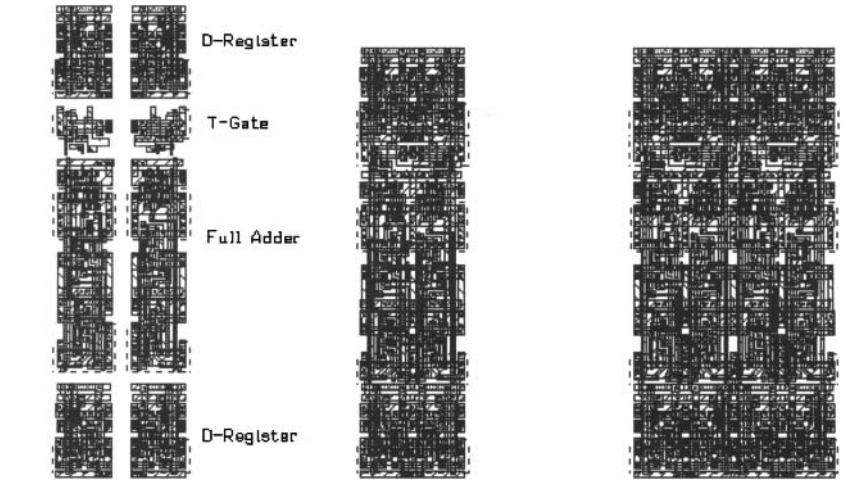


Figure 15.24 Sections of the digital interpolation filter.

limits of this technology, and the marketplace will still be there demanding ever cheaper, more powerful, and faster products. It is likely then that we will all still have the opportunity to "push a polygon" or two for the foreseeable future. There remains no doubt that the future will bring us ever more powerful software tools that will take over the tedious aspects of placing and connecting layouts, leaving to us the more creative aspects of planning and polishing them.

ADDITIONAL READING

[1] N. H. E. Weste and D. Harris, *Principles of CMOS VLSI Design*, 4th ed., Addison-Wesley, 2010. ISBN 978-0321547743.

[2] J. P. Uyemura, *Introduction to VLSI Circuits and Systems*, John Wiley and Sons Publishers, 2002. ISBN 0-471-12704-3.

[3] C. Saint and J. Saint, *IC Mask Design: Essential Layout Techniques*, McGraw-Hill, 2002. ISBN 0-071-38996-2. Companion book for reference [4].

[4] C. Saint and J. Saint, *IC Layout Basics: A Practical Guide*, McGraw-Hill, 2001. ISBN 0-071-38625-4. Very good introductory book on IC Layout.

[5] D. Clein, *CMOS IC Layout: Concepts, Methodologies, and Tools*, Newnes Publishers, 2000. ISBN 0-750-67194-7. Excellent introductory book for CMOS IC layout. Covers the entire layout process.

[6] J. Uyemura, *Physical Design of CMOS Integrated Circuits Using L-EDIT*, PWS Publishing Co., 1995. ISBN 0-534-94326-8. Covers the use of the L-EDIT layout software.

[7] Kerth, Donald A. *"Floorplanning-Lecture Notes,"* Crystal Semiconductor, Inc.

[8] Kerth, Donald A. *"Analog Tricks of the Trade-Lecture Notes."* Crystal Semiconductor, Inc.

[9] D. V. Heinbuch, *CMOS3 Cell Library*, Addison-Wesley, 1988. ISBN 0-201-11257-4. Excellent source of layout examples.