PROGRAMMABLE ASIC INTERCONNECT

All FPGAs contain some type of programmable interconnect . The structure and complexity of the interconnect is largely determined by the programming technology and the architecture of the basic logic cell. The raw material that we have to work with in building the interconnect is aluminum-based metallization, which has a sheet resistance of approximately 50 m W /square and a line capacitance of 0.2 pFcm $^{-1}$. The first programmable ASICs were constructed using two layers of metal; newer programmable ASICs use three or more layers of metal interconnect.

7.1 Actel ACT

7.2 Xilinx LCA

7.3 Xilinx EPLD

7.4 Altera MAX 5000 and 7000

7.5 Altera MAX 9000

7.6 Altera FLEX

7.7 Summary

7.8 Problems

7.9 Bibliography

7.10 References

7.1 Actel ACT

The Actel ACT family interconnect scheme shown in Figure 7.1 is similar to a channeled gate array. The channel routing uses dedicated rectangular areas of fixed size within the chip called wiring channels (or just channels). The horizontal channels run across the chip in the horizontal direction. In the vertical direction there are similar vertical channels that run over the top of the basic logic cells, the Logic Modules. Within the horizontal or vertical channels wires run horizontally or vertically, respectively,

within tracks . Each track holds one wire. The capacity of a fixed wiring channel is equal to the number of tracks it contains. Figure 7.2 shows a detailed view of the channel and the connections to each Logic Module-the input stubs and output stubs .

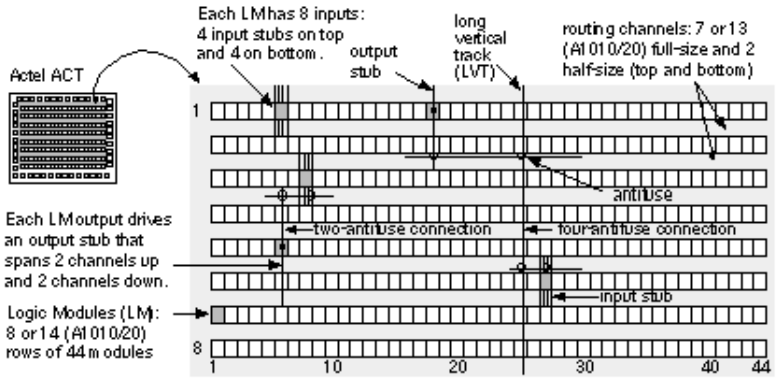


FIGURE 7.1 The interconnect architecture used in an Actel ACT family FPGA. (Source: Actel.)

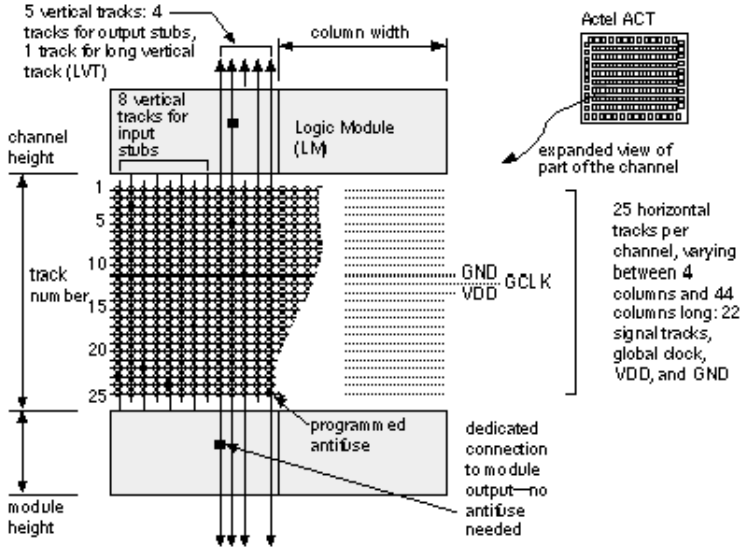


FIGURE 7.2 ACT 1 horizontal and vertical channel architecture. (Source: Actel.)

In a channeled gate array the designer decides the location and length of the interconnect within a channel. In an FPGA the interconnect is fixed at the time of manufacture. To allow programming of the interconnect, Actel divides the fixed interconnect wires within each channel into various lengths or wire segments. We call this segmented channel routing, a variation on channel routing. Antifuses join the wire segments. The designer then programs the interconnections by blowing antifuses and making connections between wire segments; unwanted connections are left unprogrammed. A statistical analysis of many different layouts determines the optimum number and the lengths of the wire segments.

7.1.1 Routing Resources

The ACT 1 interconnection architecture uses 22 horizontal tracks per channel for signal routing with three tracks dedicated to VDD, GND, and the global clock (GCLK), making a total of 25 tracks per channel. Horizontal segments vary in length from four columns of Logic Modules to the entire row of modules (Actel calls these long segments long lines).

Four Logic Module inputs are available to the channel below the Logic Module and four inputs to the channel above the Logic Module. Thus eight vertical tracks per Logic Module are available for inputs (four from the Logic Module above the channel and four from the Logic Module below). These connections are the input stubs.

The single Logic Module output connects to a vertical track that extends across the two channels above the module and across the two channels below the module. This is the output stub. Thus module outputs use four vertical tracks per module (counting two tracks from the modules below, and two tracks from the modules above each channel). One vertical track per column is a long vertical track (LVT) that spans the entire height of the chip (the 1020 contains some segmented LVTs). There are thus a total of 13 vertical tracks per column in the ACT 1 architecture (eight for inputs, four for outputs, and one for an LVT).

Table 7.1 shows the routing resources for both the ACT 1 and ACT 2 families. The last two columns show the total number of antifuses (including antifuses in the I/O cells) on each chip and the total number of antifuses assuming the wiring channels are fully populated with antifuses (an antifuse at every horizontal and vertical interconnect intersection). The ACT 1 devices are very nearly fully populated.

TABLE 7.1 Actel FPGA routing resources.

	Horizontal tracks per channel, H	Vertical tracks per column, V	Rows, R	Columns, C	Total antifuses on each chip	H ¥ V ¥ R ¥ C
A1010	22	13	8	44	112,000	100,672
A1020	22	13	14	44	186,000	176,176
A1225A	36	15	13	46	250,000	322,920
A1240A	36	15	14	62	400,000	468,720
A1280A	36	15	18	82	750,000	797,040

If the Logic Module at the end of a net is less than two rows away from the driver module, a connection requires two antifuses, a vertical track, and two horizontal segments. If the modules are more than two rows apart, a connection between them will require a long vertical track together with another vertical track (the output stub) and two horizontal tracks. To connect these tracks will require a total of four antifuses in series and this will add delay due to the resistance of the antifuses. To examine the extent of this delay problem we need some help from the analysis of RC networks.

7.1.2 Elmore's Constant

Figure 7.3 shows an RC tree -representing a net with a fanout of two. We shall assume that all nodes are initially charged to $V_{DD} = 1$ V, and that we short node 0 to ground, so $V_0 = 0$ V, at time t = 0 sec. We need to find the node voltages, V_1 to V_4, as a function of time. A similar problem arose in the design of wideband vacuum tube distributed amplifiers in the 1940s. Elmore found a measure of delay that we can use today [Rubenstein, Penfield, and Horowitz, 1983].

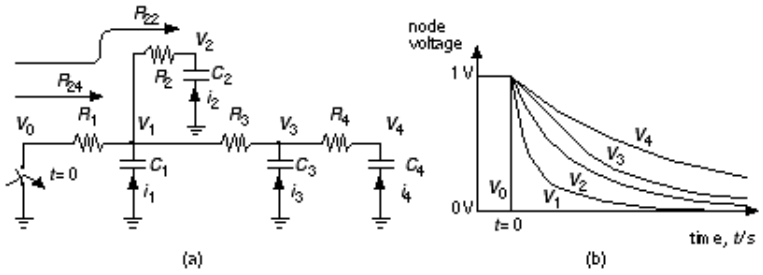


FIGURE 7.3 Measuring the delay of a net. (a) An RC tree. (b) The waveforms as a result of closing the switch at t = 0.

The current in branch k of the network is

$$i_k = -C_k \frac{dV_k}{dt} \quad . (7.1)$$

The linear superposition of the branch currents gives the voltage at node i as

$$V_i = -\sum_{k=1}^{n} R_{ki} C_k \frac{dV_k}{dt} \quad , (7.2)$$

where R_{ki} is the resistance of the path to V_0 (ground in this case) shared by node k and node i . So, for example, $R_{24} = R_1$, $R_{22} = R_1 + R_2$, and $R_{31} = R_1$.

Unfortunately, Eq. 7.2 is a complicated set of coupled equations that we cannot easily solve. We know the node voltages have different values at each point in time, but, since the waveforms are similar, let us assume the slopes (the time derivatives) of the waveforms are related to each other. Suppose we express the slope of node voltage V_k as a constant, a_k , times the slope of V_i ,

$$\frac{dV_k}{dt} = a_k \frac{dV_i}{dt} \quad . (7.3)$$

Consider the following measure of the error, E , of our approximation:

$$E = -\sum_{k=1}^{n} R_{ki} C_k \quad . (7.4)$$

The error, E , is a minimum when $a_k = 1$ since initially $V_i (t = 0) = V_k (t = 0) = 1$ V (we normalized the voltages) and $V_i (t = *) = V_k (t = *) = 0$.

Now we can rewrite Eq. 7.2 , setting $a_k = 1$, as follows:

$$V_i = -\sum_{k=1}^{n} R_{ki} C_k \frac{dV_i}{dt} \quad , (7.5)$$

This is a linear first-order differential equation with the following solution:

$$V_i (t) = \exp (- t / t_{Di}) \; ; \; t_{Di} = \sum_{k=1}^{n} R_{ki} C_k \quad . (7.6)$$

The time constant t D i is often called the Elmore delay and is different for each node. We shall refer to t D i as the Elmore time constant to remind us that, if we approximate V_i by an exponential waveform, the delay of the RC tree using 0.35/0.65 trip points is approximately t Di seconds.

7.1.3 RC Delay in Antifuse Connections

Suppose a single antifuse, with resistance R_1 , connects to a wire segment with parasitic capacitance C_1 . Then a connection employing a single antifuse will delay the signal passing along that connection by approximately one time constant, or $R_1 C_1$ seconds. If we have more than one antifuse, we need to use the Elmore time constant to estimate the interconnect delay.

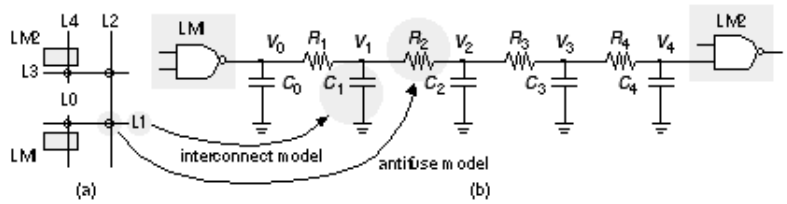


FIGURE 7.4 Actel routing model. (a) A four-antifuse connection. L0 is an output stub, L1 and L3 are horizontal tracks, L2 is a long vertical track (LVT), and L4 is an input stub. (b) An RC-tree model. Each antifuse is modeled by a resistance and each interconnect segment is modeled by a capacitance.

For example, suppose we have the four-antifuse connection shown in Figure 7.4 . Then, from Eq. 7.6 ,

$$t_{D4} = R_{14} C_1 + R_{24} C_2 + R_{14} C_1 + R_{44} C_4$$
$$= (R_1 + R_2 + R_3 + R_4) C_4 + (R_1 + R_2 + R_3) C_3 + (R_1 + R_2) C_2 + R_1 C_1$$

If all the antifuse resistances are approximately equal (a reasonably good assumption) and the antifuse resistance is much larger than the resistance of any of the metal lines, L1-L5, shown in Figure 7.4 (a very good assumption) then $R_1 = R_2 = R_3 = R_4 = R$, and the Elmore time constant is

$$t_{D4} = 4RC_4 + 3RC_3 + 2RC_2 + RC_1 \quad (7.7)$$

Suppose now that the capacitance of each interconnect segment (including all the antifuses and programming transistors that may be attached) is approximately constant, and equal to C . A connection with two antifuses will generate a 3 RC time constant, a connection with three antifuses a 6 RC time constant, and a connection with four antifuses gives a 10 RC time constant. This analysis is disturbing-it says that the interconnect delay grows quadratically ($? n^2$) as we increase the interconnect length and the number of antifuses, n . The situation is worse when the intermediate wire segments have larger capacitance than that of the short input stubs and output stubs. Unfortunately, this is the situation in an Actel FPGA where the horizontal and vertical segments in a connection may be quite long.

7.1.4 Antifuse Parasitic Capacitance

We can determine the number of antifuses connected to the horizontal and vertical lines for the Actel architecture. Each column contains 13 vertical signal tracks and each channel contains 25 horizontal tracks (22 of these are used for signals). Thus, assuming the channels are fully populated with antifuses,

- An input stub (1 channel) connects to 25 antifuses.
- An output stub (4 channels) connects to 100 (25 ¥ 4) antifuses.
- An LVT (1010, 8 channels) connects to 200 (25 ¥ 8) antifuses.
- An LVT (1020, 14 channels) connects to 350 (25 ¥ 14) antifuses.
- A four-column horizontal track connects to 52 (13 ¥ 4) antifuses.
- A 44-column horizontal track connects to 572 (13 ¥ 44) antifuses.

A connection to the diffusion of an Actel antifuse has a parasitic capacitance due to the diffusion

junction. The polysilicon of the antifuse has a parasitic capacitance due to the thin oxide. These capacitances are approximately equal. For a 2 m m CMOS process the capacitance to ground of the diffusion is 200 to 300 aF m m $^{-2}$ (area component) and 400 to 550 aF m m $^{-1}$ (perimeter component). Thus, including both area and perimeter effects, a 16 m m ^{2} diffusion contact (consisting of a 2 m m by 2 m m opening plus the required overlap) has a parasitic capacitance of 10-14 f F. If we assume an antifuse has a parasitic capacitance of approximately 10 fF in a 1.0 or 1.2 m m process, we can calculate the parasitic capacitances shown in Table 7.2 .

TABLE 7.2 Actel interconnect parameters.

Parameter	A1010/A1020	A1010B/A1020B
Technology	2.0 m m, l = 1.0 m m	1.2 m m, l = 0.6 m m
Die height (A1010)	240 mil	144 mil
Die width (A1010)	360 mil	216 mil
Die area (A1010)	86,400 mil ^{2} = 56 M l ^{2}	31,104 mil ^{2} = 56 M l ^{2}
Logic Module (LM) height (Y1)	180 m m = 180 l	108 m m = 180 l
LM width (X)	150 m m = 150 l	90 m m = 150 l
LM area (X ¥ Y1)	27,000 m m ^{2} = 27 k l ^{2}	9,720 m m ^{2} = 27 k l ^{2}
Channel height (Y2)	25 tracks = 287 m m	25 tracks = 170 m m
Channel area per LM (X ¥ Y2)	43,050 m m ^{2} = 43 k l ^{2}	15,300 m m ^{2} = 43 k l ^{2}
LM and routing area (X ¥ Y1 + X ¥ Y2)	70,000 m m ^{2} = 70 k l ^{2}	25,000 m m ^{2} = 70 k l ^{2}
Antifuse capacitance	-	10 fF
Metal capacitance	0.2 pFmm $^{-1}$	0.2 pFmm $^{-1}$
Output stub length (spans 3 LMs + 4 channels)	4 channels = 1688 m m	4 channels = 1012 m m
Output stub metal capacitance	0.34 pF	0.20 pF
Output stub antifuse connections	100	100
Output stub antifuse capacitance	-	1.0 pF
Horiz. track length	4-44 cols. = 600-6600 m m	4-44 cols. = 360-3960 m m
Horiz. track metal capacitance	0.1-1.3 pF	0.07-0.8 pF
Horiz. track antifuse connections	52-572 antifuses	52-572 antifuses
Horiz. track antifuse capacitance	-	0.52-5.72 pF
Long vertical track (LVT)	8-14 channels = 3760-6580 m m	8-14 channels = 2240-3920 m m
LVT metal capacitance	0.08-0.13 pF	0.45-0.8 pF
LVT track antifuse connections	200-350 antifuses	200-350 antifuses
LVT track antifuse capacitance		2-3.5 pF
Antifuse resistance (ACT 1)		0.5 k W (typ.), 0.7 k W (max.)

We can use the figures from Table 7.2 to estimate the interconnect delays. First we calculate the following resistance and capacitance values:

1. The antifuse resistance is assumed to be R = 0.5 k W .
2. C_0 = 1.2 pF is the sum of the gate output capacitance (which we shall neglect) and the output stub capacitance (1.0 pF due to antifuses, 0.2 pF due to metal). The contribution from this term is zero in our calculation because we have neglected the pull resistance of the driving gate.
3. $C_1 = C_3$ = 0.59 pF (0.52 pF due to antifuses, 0.07 pF due to metal) corresponding to a minimum-length horizontal track.
4. C_2 = 4.3 pF (3.5 pF due to antifuses, 0.8 pF due to metal) corresponding to a LVT in a 1020B.
5. The estimated input capacitance of a gate is C_4 = 0.02 pF (the exact value will depend on which input of a Logic Module we connect to).

From Eq. 7.7 , the Elmore time constant for a four-antifuse connection is

$$t_{D4} = 4(0.5)(0.02) + 3(0.5)(0.59) + 2(0.5)(4.3) + (0.5)(0.59) \quad (7.8)$$
$$= 5.52 \text{ ns} .$$

This matches delays obtained from the Actel delay calculator. For example, an LVT adds between 5-10 ns delay in an ACT 1 FPGA (6-12 ns for ACT 2, and 4-14 ns for ACT 3). The LVT connection is about the slowest connection that we can make in an ACT array. Normally less than 10 percent of all connections need to use an LVT and we see why Actel takes great care to make sure that this is the case.

7.1.5 ACT 2 and ACT 3 Interconnect

The ACT 1 architecture uses two antifuses for routing nearby modules, three antifuses to join horizontal segments, and four antifuses to use a horizontal or vertical long track. The ACT 2 and ACT 3 architectures use increased interconnect resources over the ACT 1 device that we have described. This reduces further the number of connections that need more than two antifuses. Delay is also reduced by decreasing the population of antifuses in the channels, and by decreasing the antifuse resistance of certain critical antifuses (by increasing the programming current).

The channel density is the absolute minimum number of tracks needed in a channel to make a given set of connections (see Section 17.2.2, "Measurement of Channel Density"). Software to route connections using channeled routing is so efficient that, given complete freedom in location of wires, a channel router can usually complete the connections with the number of tracks equal or close to the theoretical minimum, the channel density. Actel's studies on segmented channel routing have shown that increasing the number of horizontal tracks slightly (by approximately 10 percent) above density can lead to very high routing completion rates.

The ACT 2 devices have 36 horizontal tracks per channel rather than the 22 available in the ACT 1 architecture. Horizontal track segments in an ACT 3 device range from a module pair to the full channel length. Vertical tracks are: input (with a two channel span: one up, one down); output (with a four-channel span: two up, two down); and long (LVT). Four LVTs are shared by each column pair. The ACT 2/3 Logic Modules can accept five inputs, rather than four inputs for the ACT 1 modules, and thus

the ACT 2/3 Logic Modules need an extra two vertical tracks per channel. The number of tracks per column thus increases from 13 to 15 in the ACT 2/3 architecture.

The greatest challenge facing the Actel FPGA architects is the resistance of the polysilicon-diffusion antifuse. The nominal antifuse resistance in the ACT 1-2 1-2 m m processes (with a 5 mA programming current) is approximately 500 W and, in the worst case, may be as high as 700 W . The high resistance severely limits the number of antifuses in a connection. The ACT 2/3 devices assign a special antifuse to each output allowing a direct connection to an LVT. This reduces the number of antifuses in a connection using an LVT to three. This type of antifuse (a fast fuse) is blown at a higher current than the other antifuses to give them about half the nominal resistance (about 0.25 k W for AC T 2) of a normal antifuse. The nominal antifuse resistance is reduced further in the ACT 3 (using a 0.8 m m process) to 200 W (Actel does not state whether this value is for a normal or fast fuse). However, it is the worst-case antifuse resistance that will determine the worst-case performance.

7.2 Xilinx LCA

Figure 7.5 shows the hierarchical Xilinx LCA interconnect architecture.

- The vertical lines and horizontal lines run between CLBs.
- The general-purpose interconnect joins switch boxes (also known as magic boxes or switching matrices).
- The long lines run across the entire chip. It is possible to form internal buses using long lines and the three-state buffers that are next to each CLB.
- The direct connections (not used on the XC4000) bypass the switch matrices and directly connect adjacent CLBs.
- The Programmable Interconnection Points (PIP s) are programmable pass transistors that connect the CLB inputs and outputs to the routing network.
- The bidirectional (BIDI) interconnect buffers restore the logic level and logic strength on long interconnect paths.

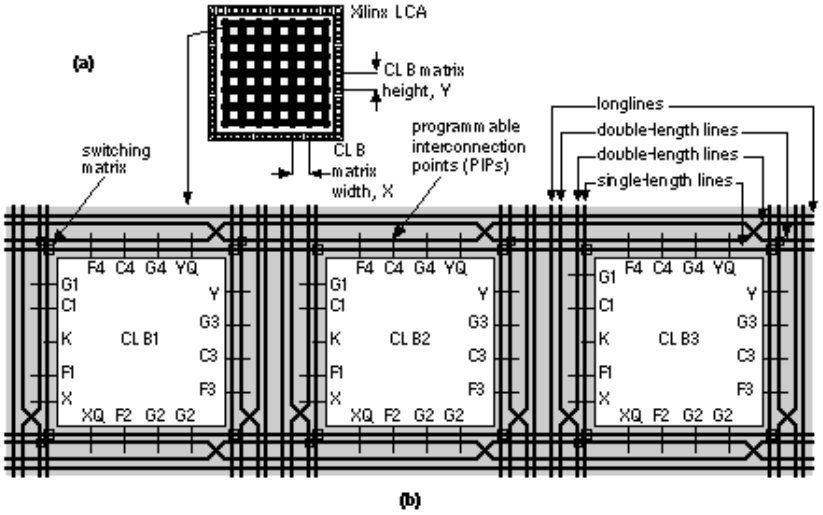


FIGURE 7.5 Xilinx LCA interconnect. (a) The LCA architecture (notice the matrix element size is larger than a CLB). (b) A simplified representation of the interconnect resources. Each of the lines is a bus.

Table 7.3 shows the interconnect data for an XC3020, a typical Xilinx LCA FPGA, that uses two-level metal interconnect. Figure 7.6 shows the switching matrix. Programming a switch matrix allows a number of different connections between the general-purpose interconnect.

TABLE 7.3 XC3000 interconnect parameters.

Parameter	XC3020
Technology	1.0 m m, l = 0.5 m m
Die height	220 mil
Die width	180 mil
Die area	39,600 mil ^{2} = 102 M l ^{2}
CLB matrix height (Y)	480 m m = 960 l
CLB matrix width (X)	370 m m = 740 l
CLB matrix area (X ¥ Y)	17,600 m m ^{2} = 710 k l ^{2}
Matrix transistor resistance, R_{P1}	0.5-1k W
Matrix transistor parasitic capacitance, C_{P1}	0.01-0.02 pF
PIP transistor resistance, R_{P2}	0.5-1k W
PIP transistor parasitic capacitance, C_{P2}	0.01-0.02 pF
Single-length line (X, Y)	370 m m, 480 m m
Single-length line capacitance: C_{LX} , C_{LY}	0.075 pF, 0.1 pF
Horizontal Longline (8X)	8 cols. = 2960 m m
Horizontal Longline metal capacitance, C_{LL}	0.6 pF

In Figure 7.6 (d), (g), and (h):

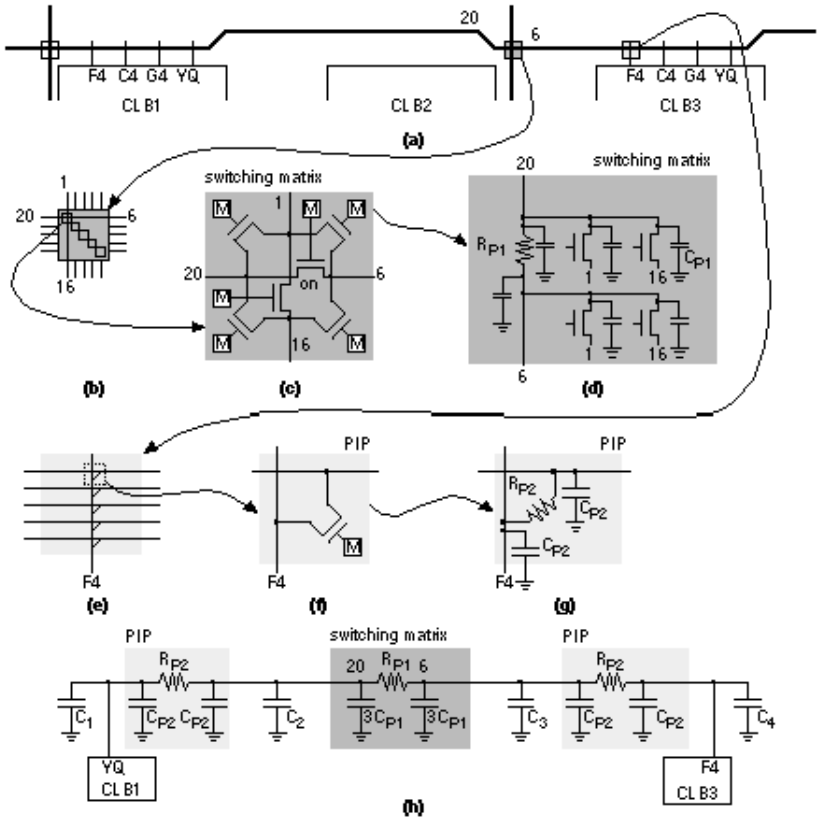


FIGURE 7.6 Components of interconnect delay in a Xilinx LCA array. (a) A portion of the interconnect around the CLBs. (b) A switching matrix. (c) A detailed view inside the switching matrix showing the pass-transistor arrangement. (d) The equivalent circuit for the connection between nets 6 and 20 using the matrix. (e) A view of the interconnect at a Programmable Interconnection Point (PIP). (f) and (g) The equivalent schematic of a PIP connection. (h) The complete RC delay path.

- $C1 = 3CP1 + 3CP2 + 0.5C_{LX}$ is the parasitic capacitance due to the switch matrix and PIPs (F4, C4, G4) for CLB1, and half of the line capacitance for the double-length line adjacent to CLB1.
- C_{P1} and R_{P1} are the switching-matrix parasitic capacitance and resistance.
- C_{P2} and R_{P2} are the parasitic capacitance and resistance for the PIP connecting YQ of CLB1 and F4 of CLB3.
- $C2 = 0.5CLX + CLX$ accounts for half of the line adjacent to CLB1 and the line adjacent to CLB2.
- $C_3 = 0.5C_{LX}$ accounts for half of the line adjacent to CLB3.
- $C_4 = 0.5C_{LX} + 3C_{P2} + C_{LX} + 3C_{P1}$ accounts for half of the line adjacent to CLB3, the PIPs of CLB3 (C4, G4, YQ), and the rest of the line and switch matrix capacitance following CLB3.

We can determine Elmore's time constant for the connection shown in Figure 7.6 as

$$t_D = R_{P2}(C_{P2} + C_2 + 3C_{P1}) + (R_{P2} + R_{P1})(3C_{P1} + C_3 + C_{P2}) + (2R_{P2} + R_{P1})(C_{P2} + C_4). \quad (7.9)$$

If RP1 = RP2 , and CP1 = CP2 , then

$$t_D = (15 + 21)R_P C_P + (1.5 + 1 + 4.5)R_P C_{LX} \cdot \quad (7.10)$$

We need to know the pass-transistor resistance RP . For example, suppose RP = 1k W . If $k'_n = 50$ m AV $^{-2}$, then (with Vt n = 0.65 V and $V_{DD} = 3.3$ V)

$$W/L = \frac{1}{k'_n R_p (V_{DD} - V_{tn})} = \frac{1}{(50 \times 10^{-6})(1 \times 10^{3})(3.3 - 0.65)} = 7.5 . \quad (7.11)$$

If L = 1 m m, both source and drain areas are 7.5 m m long and approximately 3 m m wide (determined by diffusion overlap of contact, contact width, and contact-to-gate spacing, rules 6.1a + 6.2a + 6.4a = 5.5 l in Table 2.7). Both drain and source areas are thus 23 m m ^{2} and the sidewall perimeters are 14 m m (excluding the sidewall facing the channel). If we have a diffusion capacitance of 140 aF m m $^{-2}$ (area) and 500 aF m m $^{-1}$ (perimeter), typical values for a 1.0 m m process, the parasitic source and drain capacitance is

$$C_P = (140 \times 10^{-18})(23) + (500 \times 10^{-18})(14) \quad (7.12)$$
$$= 1.022 \times 10^{-14} \text{ F} .$$

If we assume CP = 0.01 pF and CLX = 0.075 pF (Table 7.3),

$$t_D = (36)(1)(0.01) + (7)(1)(0.075) \quad (7.13)$$
$$= 0.885 \text{ ns} .$$

A delay of approximately 1 ns agrees with the typical values from the XACT delay calculator and is about the fastest connection we can make between two CLBs.

7.3 Xilinx EPLD

The Xilinx EPLD family uses an interconnect bus known as Universal Interconnection Module (UIM) to distribute signals within the FPGA. The UIM, shown in Figure 7.7 , is a programmable AND array with constant delay from any input to any output. In Figure 7.7 :

- C_G is the fixed gate capacitance of the EPROM device.
- C_D is the fixed drain parasitic capacitance of the EPROM device.
- C_B is the variable horizontal bus ("bit" line) capacitance.
- C_W is the variable vertical bus ("word" line) capacitance.

Figure 7.7 shows the UIM has 21 output connections to each FB. 1 Thus the XC7272 UIM (with a 4 ¥ 2 array of eight FBs as shown in Figure 7.7) has 168 (8 ¥ 21) output connections. Most (but not all) of the nine I/O cells attached to each FB have two input connections to the UIM, one from a chip input and one feedback from the macrocell output. For example, the XC7272 has 18 I/O cells that are outputs only and thus have only one connection to the UIM, so n = (18 ¥ 8) - 18 = 126 input connections. Now we can calculate the number of tracks in the UIM: the XC7272, for example, has H = 126 tracks and V = 168/2 = 84 tracks. The actual physical height, V , of the UIM is determined by the size of the FBs, and is close to the die height.

The UIM ranges in size with the number of FBs. For the smallest XC7236 (with a 2 ¥ 2 array of four FBs), the UIM has n = 68 inputs and 84 outputs. For the XC73108 (with a 6 ¥ 2 array of 12 FBs), the UIM has n = 198 inputs. The UIM is a large array with large parasitic capacitance; it employs a highly optimized structure that uses EPROM devices and a sense amplifier at each output. The signal swing on the UIM uses less than the full V_{DD} = 5 V to reduce the interconnect delay.

---

1. 1994 data book p. 3-62 and p. 3-78.

7.4 Altera MAX 5000 and 7000

Altera MAX 5000 devices (except the EPM5032, which has only one LAB) and all MAX 7000 devices use a Programmable Interconnect Array (PIA), shown in Figure 7.8 . The PIA is a cross-point switch for logic signals traveling between LABs. The advantages of this architecture (which uses a fixed number of connections) over programmable interconnection schemes (which use a variable number of connections) is the fixed routing delay. An additional benefit of the simpler nature of a large regular interconnect structure is the simplification and improved speed of the placement and routing software.

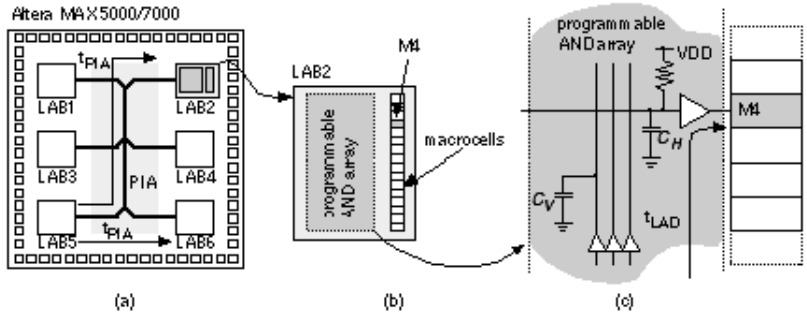


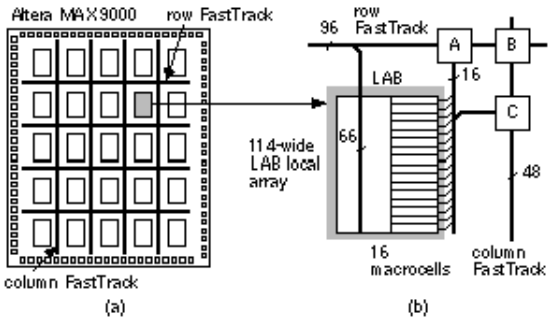
FIGURE 7.8 A simplified block diagram of the Altera MAX interconnect scheme. (a) The PIA (Programmable Interconnect Array) is deterministic-delay is independent of the path length. (b) Each LAB (Logic Array Block) contains a programmable AND array. (c) Interconnect timing within a LAB is also fixed.

Figure 7.8 (a) illustrates that the delay between any two LABs, t_{PIA}

7.5 Altera MAX 9000

Figure 7.9 shows the Altera MAX 9000 interconnect architecture. The size of the MAX 9000 LAB arrays varies between 4 ¥ 5 (rows ¥ columns) for the EPM9320 and 7 ¥ 5 for the EPM9560. The MAX 9000 is an extremely coarse-grained architecture, typical of complex PLDs, but the LABs themselves have a finer structure. Sometimes we say that complex PLDs with arrays (LABs in the Altera MAX family) that are themselves arrays (of macrocells) have a dual-grain architecture .

FIGURE 7.9 The Altera MAX 9000 interconnect scheme. (a) A 4 ¥ 5 array of Logic Array Blocks (LABs), the same size as the EMP9400 chip. (b) A simplified block diagram of the interconnect architecture showing the connection of the FastTrack buses to a LAB.



In Figure 7.9 (b), boxes A, B, and C represent the interconnection between the FastTrack buses and the 16 macrocells in each LAB:

- Box A connects a macrocell to one row channel.
- Box B connects three column channels to two row channels.
- Box C connects a macrocell to three column channels.

7.6 Altera FLEX

Figure 7.10 shows the interconnect used in the Altera FLEX family of complex PLDs. Altera refers to the FLEX interconnect and MAX 9000 interconnect by the same name, FastTrack, but the two are different because the granularity of the logic cell arrays is different. The FLEX architecture is of finer grain than the MAX arrays-because of the difference in programming technology. The FLEX horizontal interconnect is much denser (at 168 channels per row) than the vertical interconnect (16 channels per column), creating an aspect ratio for the interconnect of over 10:1 (168:16). This imbalance is partly due to the aspect ratio of the die, the array, and the aspect ratio of the basic logic cell, the LAB.

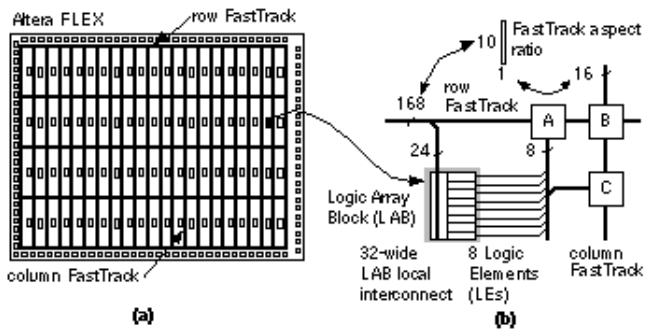


FIGURE 7.10 The Altera FLEX interconnect scheme. (a) The row and column FastTrack interconnect. The chip shown, with 4 rows ¥ 21 columns, is the same size as the EPF8820. (b) A simplified diagram of the interconnect architecture showing the connections between the FastTrack buses and a LAB. Boxes A, B, and C represent the bus-to-bus connections.

As an example, the EPF8820 has 4 rows and 21 columns of LABs (Figure 7.10 a). Ignoring, for simplicity's sake, what happens at the edge of the die we can total the routing channels as follows:

- Horizontal channels = 4 rows ¥ 168 channels/row = 672 channels.
- Vertical channels = 21 rows ¥ 16 channels/row = 336 channels.

It appears that there is still approximately twice (672:336) as much interconnect capacity in the horizontal direction as the vertical. If we look inside the boxes A, B, and C in Figure 7.10 (b) we see that for individual lines on each bus:

- Box A connects an LE to two row channels.
- Box B connects two column channels to a row channel.
- Box C connects an LE to two column channels.

7.7 Summary

The RC product of the parasitic elements of an antifuse and a pass transistor are not too different. However, an SRAM cell is much larger than an antifuse which leads to coarser interconnect architectures for SRAM-based programmable ASICs. The EPROM device lends itself to large wired-logic structures. These differences in programming technology lead to different architectures:

- The antifuse FPGA architectures are dense and regular.
- The SRAM architectures contain nested structures of interconnect resources.
- The complex PLD architectures use long interconnect lines but achieve deterministic routing.

Table 7.4 is a look-up table for Tables 7.5 and 7.6 , which summarize the features of the logic cells used by the various FPGA vendors.

TABLE 7.4 I/O Cell Tables.

Table	Programmable ASIC family
Table 7.5	Actel (ACT 1)
	Xilinx (XC3000)
	Actel (ACT 2)
	Xilinx (XC4000)
	Altera MAX (EPM 5000)
	Xilinx EPLD (XC7200/7300)
	Actel (ACT 3)
	QuickLogic (pASIC 1)
	Crosspoint (CP20K)
	Altera MAX (EPM 7000)
	Atmel (AT6000)
	Xilinx LCA (XC5200)

Table	Programmable ASIC family
Table 7.6	Xilinx (XC8100)
	Lucent ORCA (2C)
	Altera FLEX (8000/10k)
	AMD MACH 5
	Actel 3200DX
	Altera MAX (EPM 9000)

TABLE 7.5 Programmable ASIC interconnect.

	Actel (ACT 1)	Xilinx (XC3000)	Actel (ACT 2)	Xilinx (XC4000)
Interconnect between logic cells (tracks = trks)	Channeled array with segmented routing, long lines: 25 trks/ch. (horiz.); 13 trks/ch. (vert.); < 4 antifuses/path	Switch box, PIPs (Programmable Interconnect Points), 3-state internal bus, and long lines	Channeled array with segmented routing, long lines: 36 trks/ch. (horiz.); 15 trks/ch. (vert.); < 4 antifuses/path	Switch box, PIPs (Programmable Interconnect Points), 3-state internal bus, and long lines
Interconnect delay	Variable	Variable	Variable	Variable
Interconnect inside logic cells	Poly-diffusion antifuse	32-bit SRAM LUT	Poly-diffusion antifuse	32-bit SRAM LUT

	Altera (MAX 5000)	Xilinx EPLD	QuickLogic (pASIC 1)	Actel (ACT 3)
Interconnect between logic cells	Cross-bar PIA (Programmable Interconnect Architecture) using EPROM programmable-AND array	UIM (Universal Interconnect Matrix) using EPROM programmable-AND array	Programmable fully populated antifuse matrix	Channeled array with segmented routing, long lines: <4 antifuses/path
Interconnect delay	Fixed	Fixed	Variable	Variable
Interconnect inside logic cells	EPROM	EPROM	Metal-metal antifuse	Poly-diffusion antifuse

	Crosspoint (CP20K)	Altera MAX (MAX 7000)	Atmel (AT6000)	Xilinx LCA (XC5200)
Interconnect between logic cells	Programmable highly interconnected matrix	Fixed cross-bar PIA (Programmable Interconnect Architecture)	Programmable regular, local, and express bus scheme with line repeaters	Switch box, PIPs (Programmable Interconnect Points), 3-state internal bus, and long lines
Interconnect delay	Variable	Fixed	Variable	Variable
Interconnect inside logic cells	Metal-metal antifuse	EEPROM	SRAM	16-bit SRAM LUT

TABLE 7.6 Programmable ASIC interconnect (continued).

	Xilinx (XC8100)	Lucent ORCA 2C	Altera FLEX 8000/10k
Interconnect between logic cells	Channeled array with segmented routing, long lines. Programmable fully populated antifuse matrix.	Switch box, SRAM programmable interconnect, 3-state internal bus, and long lines	Row and column FastTrack between LABs
Interconnect			Fixed with small

Interconnect delay	Variable	Variable	variation in delay in row FastTrack
Interconnect inside logic cells	Antifuse	SRAM LUTs and MUXs	LAB local interconnect between LEs. 16-bit SRAM LUT in LE.
	AMD MACH 5	Actel 3200DX	Altera MAX 9000
Interconnect between logic cells	EPROM programmable array	Channeled gate array with segmented routing, long lines	Row and column FastTrack between LABs
Interconnect delay	Fixed	Variable	Fixed
Interconnect inside logic cells	EPROM	Poly-diffusion antifuse	Programmable AND array inside LAB, EEPROM MUXes

The key points covered in this chapter are:

- The difference between deterministic and nondeterministic interconnect
- Estimating interconnect delay
- Elmore's constant

7.8 Problems

* = Difficult, ** = Very difficult, *** = Extremely difficult

7.1 (*Xilinx interconnect, 120 min.)

- a. Write a minitutorial (one or two pages) explaining what you need to know to run and use the XACT delay calculator. Explain how to choose the part, set the display preferences, make connections to CLBs and the interconnect, and obtain timing figures.
- b. Use the XACT editor to determine typical delays using the longlines, a switch matrix, the PIPs, and BIDI buffers (see the Xilinx data book for more detailed explanations of the interconnect structure). Draw six different typical paths using these elements and show the components of delay. Include screen shots showing the layout of the paths and cells with detailed explanations of the figures.
- c. Construct a path using the TBUFs, the three-state buffers, driving a longline (do not forget the pull-up). Show the XACT calculated delay for your path and explain the number from data book parameters (list them and the page number from the data book).
- d. Extend one simple path to the I/O and explain the input and output timing, again using the data book.
- e. Include screen shots from the layout editor showing you example paths.
- f. Bury all the ASCII (but not binary) files you used and the tools produced inside your report using "Hidden Text." Include explanations as to what these files are and which parts

of the report they go with. This includes any schematic files, netlist files, and all files produced by the Xilinx tools. Use a separate directory for this problem and make a list in your report of all files (binary and ASCII) with explanations of what each file is.

7.2 (*Actel interconnect, 120 min.) Use the Actel chip editor to explore the properties of the interconnect scheme in a similar fashion to Problem 7.1 with the following changes: in part b make at least six different paths using various antifuse connections and explain the numbers from the delay calculator. Omit part c.

7.3 (*Altera MAX interconnect, 120 min.) Use the Altera tools to determine the properties of the MAX or FLEX interconnect in a similar fashion to Problem 7.1 with the following changes: In parts b and c construct at least six example circuits that show the various paths through the FastTrack or PIA chip-level interconnect, the local LAB array, the LAB, and the macrocells.

7.4 (**Custom ASICs, 120 min.)

- a. Write a minitutorial (one or two pages) explaining how to run an ASIC tool (Compass/Mentor/Cadence/Tanner). Enter a simple circuit (using schematic entry or synthesis and cells from a cell library) and obtain a delay estimate.
- b. Construct at least six example circuits that show various logic paths using various logic cells (for example: an inverter, a full adder).
- c. Perform a timing simulation (either using a static timing verifier or using a logic simulator). Compare your results with those from a data book.
- d. Extract the circuit to include the parasitic capacitances from layout in your circuit netlist and run a simulation to predict the delays.
- e. Compare the results that include routing capacitance with the data book values for the logic cell delays and with the values predicted before routing.
- f. Extend one simple path to the outputs of the chip by including I/O pads in your circuit and explain the input and output timing predictions.
- g. Bury the ASCII files you used and the tools produced inside your report.

7.5 (**Actel stubs, 60 min.)

- a. Which metal layers do you think Actel assigns to the horizontal and vertical interconnect in the ACT 1-3 architectures and why?
- b. Why do the ACT 1-3 input stubs not extend over more than two channels above and below the Logic Modules, since this would reduce the need for LVTs?
- c. The ACT 2 data sheet describes the output stubs as "twisted" (or interwoven) so that they occupy only four tracks. Show that the stubs occupy four vertical tracks whether they are twisted or not.
- d. Suggest the real reason for the twisted stubs.

7.6 (A three-input NAND in ACT 1, 30 min.) The macros that require two ACT 1 modules include the three-input NAND (others include four-input NAND, AND, NOR).

- a. What is the problem with trying to implement a three-input NAND gate using the Actel ACT 1 Logic Module?
- b. Suggest a modification to the ACT 1 Logic Module that would allow the implementation

of a three-input NAND using one of your new Logic Modules.

- c. Can you think of a reason why Actel did not use your modification to its Logic Module design? Hi nt: The modification has to do with routing, and not the logic itself.

7.7 (*Actel architecture, 60 min.) This is a long but relatively straightforward problem that "reverse-engineers" the Actel architecture. If you measured the chip photo on the front of the April 1990 Actel data book, you would find the following:

1. Die height (scribe to scribe) = 170 mm.
2. Channel height = 8 mm (there are 7 full-height and 2 half-height channels).
3. Logic Module height = 5 mm (there are 8 rows of Logic Modules).
4. Column (Logic Module) width = 4.2 mm.

(The scribe line is an area at the edge of a die where a cut is made by a diamond saw when the dice are separated.) An Actel 1010 die in 2 m m technology is 240 mil high by 360 mil wide (p. 4-17 in the 1990 data book). Assuming these data book dimensions are scribe to scribe, calculate (a) the Logic Module height, (b) the channel height, and (c) the column (Logic Module) width.

Given that there are 25 tracks per horizontal channel, and 13 tracks per column in the vertical direction, calculate (d) the horizontal channel track spacing and (e) the vertical channel track spacing. (f) Using the fact that each output stub spans two channels above and below the Logic Module, calculate the height of the output stub.

We can now estimate the capacitance of the Logic Module stubs and interconnect. Assume the interconnect capacitance is 0.2 $pFmm^{-1}$. (g) Calculate the capacitance of an output stub and an input stub. (h) Calculate the width and thus the capacitance of the horizontal tracks that are from four columns to 44 columns long.

You should not have to make any other assumptions to calculate these figures, but if you do, state them clearly. The figures you have calculated are summarized in Table 7.2 .

7.8 (Xilinx bank shots, 20 min.) Figure 7.11 shows a magic box. Explain how to use a "bank shot" to enter one side of the box, bounce off another, and exit on a third side. What is the delay involved in this maneuver?

7.9 Bibliography

The paper by Greene et al. [1993] (reprinted in the 1994 Actel data book) is a good description of the Actel interconnect. The 1995 AT&T data book contains a very detailed account of the routing for the ORCA series of FPGAs, which is similar to the Xilinx LCA interconnect. You can learn a great deal about the details of the Lucent and Xilinx interconnect architecture from the AT&T data book. The Xilinx data book gives a good high-level overview of SRAM-based FPGA interconnect. The best way to learn about any FPGA interconnect is to use the software tools provided by the vendor. The Xilinx XACT editor that shows point-to-point routing delays on a graphical representation of the chip layout is an easy way to become familiar with the interconnect properties. The book by Brown et al. [