PROGRAMMABLE ASIC I/O CELLS

All programmable ASICs contain some type of input/output cell (I/O cell). These I/O cells handle driving logic signals off-chip, receiving and conditioning external inputs, as well as handling such things as electrostatic protection. This chapter explains the different types of I/O cells that are used in programmable ASICs and their functions.

The following are different types of I/O requirements.

- DC output. Driving a resistive load at DC or low frequency (less than 1 MHz). Example loads are light-emitting diodes (LEDs), relays, small motors, and such. Can we supply an output signal with enough voltage, current, power, or energy?
- AC output. Driving a capacitive load with a high-speed (greater than 1 MHz) logic signal off-chip. Example loads are other logic chips, a data or address bus, ribbon cable. Can we supply a valid signal fast enough?
- DC input. Example sources are a switch, sensor, or another logic chip. Can we correctly interpret the digital value of the input?
- AC input. Example sources are high-speed logic signals (higher than 1 MHz) from another chip. Can we correctly interpret the input quickly enough?
- Clock input. Examples are system clocks or signals on a synchronous bus. Can we transfer the timing information from the input to the appropriate places on the chip correctly and quickly enough?
- Power input. We need to supply power to the I/O cells and the logic in the core, without introducing voltage drops or noise. We may also need a separate power supply to program the chip.

These issues are common to all FPGAs (and all ICs) so that the design of FPGA I/O cells is driven by the I/O requirements as well as the programming technology.

6.1 DC Output

6.2 AC Output

6.3 DC Input

6.4 AC Input

6.5 Clock Input

6.6 Power Input

6.7 Xilinx I/O Block

6.8 Other I/O Cells

6.9 Summary

6.10 Problems

6.11 Bibliography

6.12 References

6.1 DC Output

Figure 6.1 shows a robot arm driven by three small motors together with switches to control the motors. The motor armature current varies between 50 mA and nearly 0.5 A when the motor is stalled. Can we replace the switches with an FPGA and drive the motors directly?

FIGURE 6.1 A robot arm. (a) Three small DC motors drive the arm. (b) Switches control each motor.

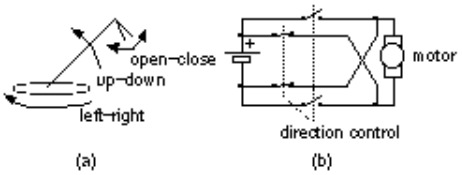


Figure 6.2 shows a CMOS complementary output buffer used in many FPGA I/O cells and its DC characteristics. Data books typically specify the output characteristics at two points, A (V_{OHmin}, I_{OHmax}) and B (V_{OLmax}, I_{OLmax}), as shown in Figure 6.2 (d). As an example, values for the Xilinx XC5200 are as follows [1]:

- V_{OLmax} = 0.4 V, low-level output voltage at I_{OLmax} = 8.0 mA.
- V_{OHmin} = 4.0 V, high-level output voltage at I_{OHmax} = -8.0 mA.

By convention the output current, I_O, is positive if it flows into the output. Input currents, if there are any, are positive if they flow into the inputs. The Xilinx XC5200 specifications show that the output buffer can force the output pad to 0.4 V or lower and sink no more than 8 mA if the load requires it. CMOS logic inputs that may be connected to the pad draw minute amounts of current, but bipolar TTL inputs can require several milliamperes. Similarly, when the output is 4 V, the buffer can source 8 mA. It is common to say that V_{OLmax} = 0.4 V and V_{OHmin} = 4.0 V for a technology-without referring to the current values at which these are measured-strictly this is incorrect.

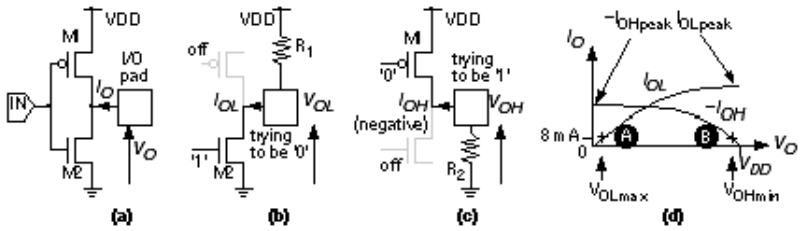


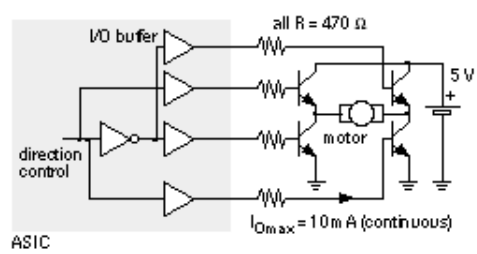
FIGURE 6.2 (a) A CMOS complementary output buffer. (b) Pull-down transistor M2 (M1 is off) sinks (to GND) a current I_{OL} through a pull-up resistor, R_1. (c) Pull-up transistor M1 (M2 is off) sources (from VDD) current $-I_{OH}$ (I_{OH} is negative) through a pull-down resistor, R_2. (d) Output characteristics.

If we force the output voltage , V_O , of an output buffer, using a voltage supply, and measure the output current, IO , that results, we find that a buffer is capable of sourcing and sinking far more than the specified I_{OHmax} and I_{OLmax} values. Most vendors do not specify output characteristics because they are difficult to measure in production. Thus we normally do not know the value of I_{OLpeak} or I_{OHpeak} ; typical values range from 50 to 200 mA.

Can we drive the motors by connecting several output buffers in parallel to reach a peak drive current of 0.5 A? Some FPGA vendors do specifically allow you to connect adjacent output cells in parallel to increase the output drive. If the output cells are not adjacent or are on different chips, there is a risk of contention. Contention will occur if, due to delays in the signal arriving at two output cells, one output buffer tries to drive an output high while the other output buffer is trying to drive the same output low. If this happens we essentially short VDD to GND for a brief period. Although contention for short periods may not be destructive, it increases power dissipation and should be avoided. [2]

It is thus possible to parallel outputs to increase the DC drive capability, but it is not a good idea to do so because we may damage or destroy the chip (by exceeding the maximum metal electromigration limits). Figure 6.3 shows an alternative-a simple circuit to boost the drive capability of the output buffers. If we need more power we could use two operational amplifiers (op-amps) connected as voltage followers in a bridge configuration. For even more power we could use discrete power MOSFETs or power op-amps.

FIGURE 6.3 A circuit to drive a small electric motor (0.5 A) using ASIC I/O buffers. Any npn transistors with a reasonable gain (b ª 100) that are capable of handling the peak current (0.5 A) will work with an output buffer that is capable of sourcing more than 5 mA. The 470 W resistors drop up to 5 V if an output buffer current approaches 10 mA, reducing the drive to the output transistors.



6.1.1 Totem-Pole Output

Figure 6.4 (a) and (b) shows a totem-pole output buffer and its DC characteristics. It is similar to the TTL totem-pole output from which it gets its name (the totem-pole circuit has two stacked transistors of the same type, whereas a complementary output uses transistors of opposite types). The high-level voltage, V_{OHmin} , for a totem pole is lower than VDD . Typically V_{OHmin} is in the range of 3.5 V to 4.0 V (with VDD = 5 V), which makes rising and falling delays more symmetrical and more closely matches TTL voltage levels. The disadvantage is that the totem pole will typically only drive the output as high as 3-4 V; so this would not be a good choice of FPGA output buffer to work with the circuit shown in Figure 6.3 .

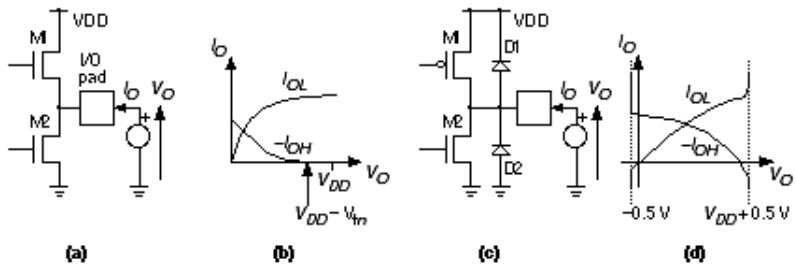


FIGURE 6.4 Output buffer characteristics. (a) A CMOS totem-pole output stage (both M1 and M2 are n -channel transistors). (b) Totem-pole output characteristics. (c) Clamp diodes, D1 and D2, in an output buffer (these diodes are present in all output buffers-totem-pole or complementary). (d) The clamp diodes start to conduct as the output voltage exceeds the supply voltage bounds.

6.1.2 Clamp Diodes

Figure 6.4 (c) show the connection of clamp diodes (D1 and D2) that prevent the I/O pad from voltage excursions greater than V_{DD} and less than V_{SS} . Figure 6.4 (d) shows the resulting characteristics.

---

1. XC5200 data sheet, October 1995 (v. 3.0).

2. Actel specifies a maximum I/O current of ± 20 mA for ACT3 family (1994 data book, p. 1-93) and its ES family. Altera specifies the maximum DC output current per pin, for example ± 25 mA for the FLEX 10k (July 1995, v. 1 data sheet, p. 42).

6.2 AC Output

Figure 6.5 shows an example of an off-chip three-state bus. Chips that have inputs and outputs connected to a bus are called bus transceivers . Can we use FPGAs to perform the role of bus transceivers? We will focus on one bit, B1, on bus BUSA, and we shall call it BUSA.B1. We need

unique names to refer to signals on each chip; thus CHIP1.OE means the signal OE inside CHIP1. Notice that CHIP1.OE is not connected to CHIP2.OE.

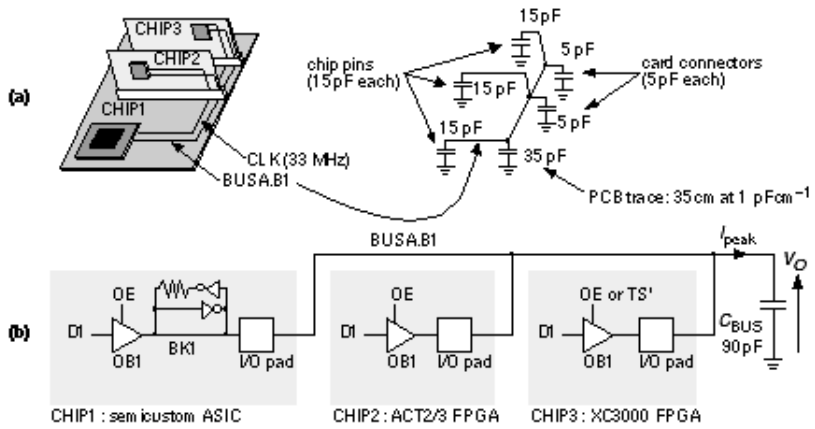


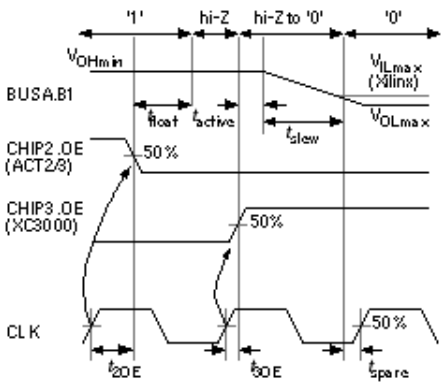
FIGURE 6.5 A three-state bus. (a) Bus parasitic capacitance. (b) The output buffers in each chip. The ASIC CHIP1 contains a bus keeper, BK1.

Figure 6.6 shows the timing of part of a bus transaction (a sequence of signals on a bus):

1. Initially CHIP2 drives BUSA.B1 high (CHIP2.D1 is ’1’ and CHIP2.OE is ’1’).
2. The buffer output enable on CHIP2 (CHIP2.OE) goes low, floating the bus. The bus will stay high because we have a bus keeper, BK1.
3. The buffer output enable on CHIP3 (CHIP3.OE) goes high and the buffer drives a low onto the bus (CHIP3.D1 is ’0’).

We wish to calculate the delays involved in driving the off-chip bus in Figure 6.6 . In order to find t_{float} , we need to understand how Actel specifies the delays for its I/O cells. Figure 6.7 (a) shows the circuit used for measuring I/O delays for the ACT FPGAs. These measurements do not use the same trip points that are used to characterize the internal logic (Actel uses input and output trip points of 0.5 for internal logic delays).

FIGURE 6.6 Three-state bus timing for Figure 6.5 . The on-chip delays, t_{2OE} and t_{3OE} , for the logic that generates signals CHIP2.E1 and CHIP3.E1 are derived from the timing models described in Chapter 5 (the minimum values for each chip would be the clock-to-Q delay times).



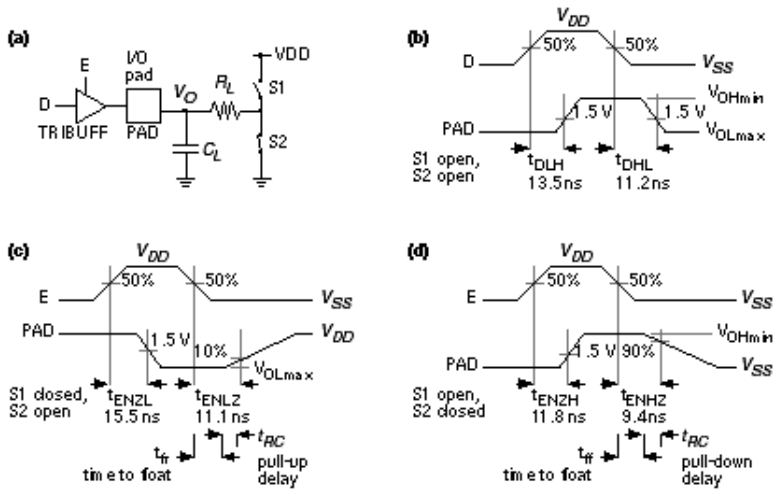


FIGURE 6.7 (a) The test circuit for characterizing the ACT 2 and ACT 3 I/O delay parameters. (b) Output buffer propagation delays from the data input to PAD (output enable, E, is high). (c) Three-state delay with D low. (d) Three-state delay with D high. Delays are shown for ACT 2 'Std' speed grade, worst-case commercial conditions (R_L = 1 k W , C_L = 50 pF, V_{OHmin} = 2.4 V, V_{OLmax} = 0.5 V). (The Actel three-state buffer is named TRIBUFF, an input buffer INBUF, and the output buffer, OUTBUF.)

Notice in Figure 6.7 (a) that when the output enable E is '0' the output is three-stated (high-impedance or hi-Z). Different companies use different polarity and naming conventions for the "output enable" signal on a three-state buffer. To measure the buffer delay (measured from the change in the enable signal, E) Actel uses a resistor load (R_L = 1 k W for ACT 2). The resistor pulls the buffer output high or low depending on whether we are measuring:

- t_{ENZL} , when the output switches from hi-Z to '0'.
- t_{ENLZ} , when the output switches from '0' to hi-Z.
- t_{ENZH} , when the output switches from hi-Z to '1'.
- t_{ENHZ} , when the output switches from '1' to hi-Z.

Other vendors specify the time to float a three-state output buffer directly (t_{fr} and t_{ff} in Figure 6.7 c and d). This delay time has different names (and definitions): disable time , time to begin hi-Z , or time to turn off .

Actel does not specify the time to float but, since $R_L C_L = 50$ ns, we know $t_{RC} = - R_L C_L \ln 0.9$ or approximately 5.3 ns. Now we can estimate that

$t_{fr} = t_{ENLZ} - t_{RC} = 11.1 - 5.3 = 5.8$ ns, and $t_{ff} = 9.4 - 5.3 = 4.1$ ns,

and thus the Actel buffer can float the bus in $t_{float} = 4.1$ ns (Figure 6.6).

The Xilinx FPGA is responsible for the second part of the bus transaction. The time to make the buffer CHIP2.B1 active is t_{active} . Once the buffer is active, the output transistors turn on, conducting a current I_{peak} . The output voltage V_O across the load capacitance, C_{BUS} , will slew or change at a steady rate, $d V_O / d t = I_{peak} / C_{BUS}$; thus $t_{slew} = C_{BUS} D V_O / I_{peak}$, where $D V_O$ is the change in output voltage.

Vendors do not always provide enough information to calculate t_{active} and t_{slew} separately, but we can usually estimate their sum. Xilinx specifies the time from the three-state input switching to the time the "pad is active and valid" for an XC3000-125 switching with a 50 pF load, to be $t_{active} = t_{TSON} = 11$ ns (fast option), and 27 ns (slew-rate limited option). [1] If we need to drive the bus in less than one clock cycle (30 ns), we will definitely need to use the fast option.

A supplement to the XC3000 timing data specifies the additional fall delay for switching large capacitive loads (above 50 pF) as $R_{fall} = 0.06$ nspF $^{-1}$ (falling) and $R_{rise} = 0.12$ nspF $^{-1}$ (rising) using the fast output option. [2] We can thus estimate that

I_{peak} ª (5 V)/(-0.06 ¥ 10^3 sF $^{-1}$) ª -84 mA (falling)

and I_{peak} ª (5 V)/(0.12 ¥ 10^3 sF $^{-1}$) ª 42 mA (rising).

Now we can calculate,

$t_{slew} = R_{fall} (C_{BUS} - 50 \text{ pF}) = (90 \text{ pF} - 50 \text{ pF}) (0.06 \text{ nspF}^{-1})$ or 2.4 ns ,

for a total falling delay of 11 + 2.4 = 13.4 ns. The rising delay is slower at 11 + (40 pF)(0.12 nspF $^{-1}$) or 15.8 ns. This leaves (30 - 15.8) ns, or about 14 ns worst-case, to generate the output enable signal CHIP2.OE (t_{3OE} in Figure 6.6) and still leave time t_{spare} before the bus data is latched on the next clock edge. We can thus probably use a XC3000 part for a 30 MHz bus transceiver, but only if we use the fast slew-rate option.

An aside: Our example looks a little like the PCI bus used on Pentium and PowerPC systems, but the

bus transactions are simplified. PCI buses use a sustained three-state system (s / t / s). On the PCI bus an s / t / s driver must drive the bus high for at least one clock cycle before letting it float. A new driver may not start driving the bus until a clock edge after the previous driver floats it. After such a turnaround cycle a new driver will always find the bus parked high.

6.2.1 Supply Bounce

Figure 6.8 (a) shows an n -channel transistor, M1, that is part of an output buffer driving an output pad, OUT1; M2 and M3 form an inverter connected to an input pad, IN1; and M4 and M5 are part of another output buffer connected to an output pad, OUT2. As M1 sinks current pulling OUT1 low (V_{o1} in Figure 6.8 b), a substantial current I_{OL} may flow in the resistance, R_S , and inductance, L_S , that are between the on-chip GND net and the off-chip, external ground connection.

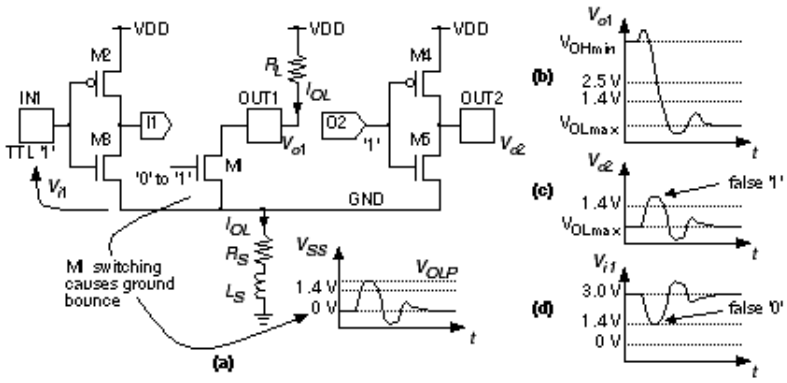


FIGURE 6.8 Supply bounce. (a) As the pull-down device, M1, switches, it causes the GND net (value V_{SS}) to bounce. (b) The supply bounce is dependent on the output slew rate. (c) Ground bounce can cause other output buffers to generate a logic glitch. (d) Bounce can also cause errors on other inputs.

The voltage drop across R_S and L_S causes a spike (or transient) on the GND net, changing the value of V_{SS} , leading to a problem known as supply bounce . The situation is illustrated in Figure 6.8 (a), with V_{SS} bouncing to a maximum of V_{OLP} . This ground bounce causes the voltage at the output, V_{o2} , to bounce also. If the threshold of the gate that OUT2 is driving is a TTL level at 1.4 V, for example, a ground bounce of more than 1.4 V will cause a logic high glitch (a momentary transition from one logic level to the opposite logic level and back again).

Ground bounce may also cause problems at chip inputs. Suppose the inverter M2/M3 is set to have a TTL threshold of 1.4 V and the input, IN1, is at a fixed voltage equal to 3 V (a respectable logic high for bipolar TTL). In this case a ground bounce of greater than 1.6 V will cause the input, IN1, to see a logic low instead of a high and a glitch will be generated on the inverter output, I1. Supply bounce can also occur on the VDD net, but this is usually less severe because the pull-up transistors in an output buffer are usually weaker than the pull-down transistors. The risk of generating a glitch is also greater at the low logic level for TTL-threshold inputs and TTL-level outputs because the low-level noise margins are

smaller than the high-level noise margins in TTL.

Sixteen SSOs, with each output driving 150 pF on a bus, can generate a ground bounce of 1.5 V or more. We cannot simulate this problem easily with FPGAs because we are not normally given the characteristics of the output devices. As a rule of thumb we wish to keep ground bounce below 1 V. To help do this we can limit the maximum number of SSOs, and we can limit the number of I/O buffers that share GND and VDD pads.

To further reduce the problem, FPGAs now provide options to limit the current flowing in the output buffers, reducing the slew rate and slowing them down. Some FPGAs also have quiet I/O circuits that sense when the input to an output buffer changes. The quiet I/O then starts to change the output using small transistors; shortly afterwards the large output transistors "drop-in." As the output approaches its final value, the large transistors "kick-out," reducing the supply bounce.

6.2.2 Transmission Lines

Most of the problems with driving large capacitive loads at high speed occur on a bus, and in this case we may have to consider the bus as a transmission line. Figure 6.9 (a) shows how a transmission line appears to a driver, D1, and receiver, R1, as a constant impedance, the characteristic impedance of the line, Z_0 . For a typical PCB trace, Z_0 is between 50 W and 100 W .

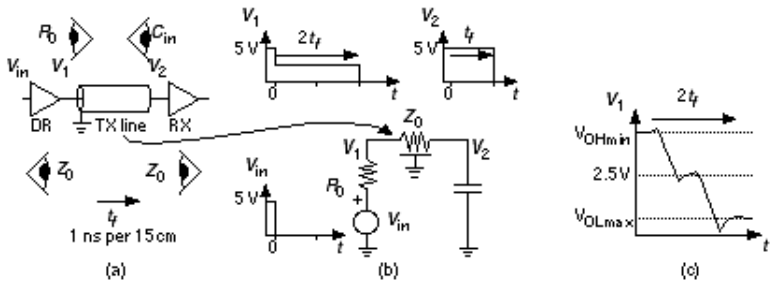


FIGURE 6.9 Transmission lines. (a) A printed-circuit board (PCB) trace is a transmission (TX) line. (b) A driver launches an incident wave, which is reflected at the end of the line. (c) A connection starts to look like a transmission line when the signal rise time is about equal to twice the line delay (2 t_f).

The voltages on a transmission line are determined by the value of the driver source resistance, R_0 , and the way that we terminate the end of the transmission line. In Figure 6.9 (a) the termination is just the capacitance of the receiver, C_{in} . As the driver switches between 5 V and 0 V, it launches a voltage wave down the line, as shown in Figure 6.9 (b). The wave will be $Z_0 / (R_0 + Z_0)$ times 5 V in magnitude, so that if R_0 is equal to Z_0 , the wave will be 2.5 V.

Notice that it does not matter what is at the far end of the line. The bus driver sees only Z_0 and not C_{in} . Imagine the transmission line as a tunnel; all the bus driver can see at the entrance is a little way into the tunnel-it could be 500 m or 5 km long. To find out, we have to go with the wave to the end, turn around,

come back, and tell the bus driver. The final result will be the same whether the transmission line is there or not, but with a transmission line it takes a little longer for the voltages and currents to settle down. This is rather like the difference between having a conversation by telephone or by post.

The propagation delay (or time of flight), t_f, for a typical PCB trace is approximately 1 ns for every 15 cm of trace (the signal velocity is about one-half the speed of light). A voltage wave launched on a transmission line takes a time t_f to get to the end of the line, where it finds the load capacitance, C_{in}. Since no current can flow at this point, there must be a reflection that exactly cancels the incident wave so that the voltage at the input to the receiver, at V_2, becomes exactly zero at time t_f. The reflected wave travels back down the line and finally causes the voltage at the output of the driver, at V_1, to be exactly zero at time $2\ t_f$. In practice the nonidealities of the driver and the line cause the waves to have finite rise times. We start to see transmission line behavior if the rise time of the driver is less than $2\ t_f$, as shown in Figure 6.9 (c).

There are several ways to terminate a transmission line. Figure 6.10 illustrates the following methods:

- Open-circuit or capacitive termination. The bus termination is the input capacitance of the receivers (usually less than 20 pF). The PCI bus uses this method.
- Parallel resistive termination. This requires substantial DC current (5 V / 100 W = 50 mA for a 100 W line). It is used by bipolar logic, for example emitter-coupled logic (ECL), where we typically do not care how much power we use.
- Thévenin termination. Connecting 300 W in parallel with 150 W across a 5 V supply is equivalent to a 100 W termination connected to a 1.6 V source. This reduces the DC current drain on the drivers but adds a resistance directly across the supply.
- Series termination at the source. Adding a resistor in series with the driver so that the sum of the driver source resistance (which is usually 50 W or even less) and the termination resistor matches the line impedance (usually around 100 W). The disadvantage is that it generates reflections that may be close to the switching threshold.
- Parallel termination with a voltage bias. This is awkward because it requires a third supply and is normally used only for a specialized high-speed bus.
- Parallel termination with a series capacitance. This removes the requirement for DC current but introduces other problems.

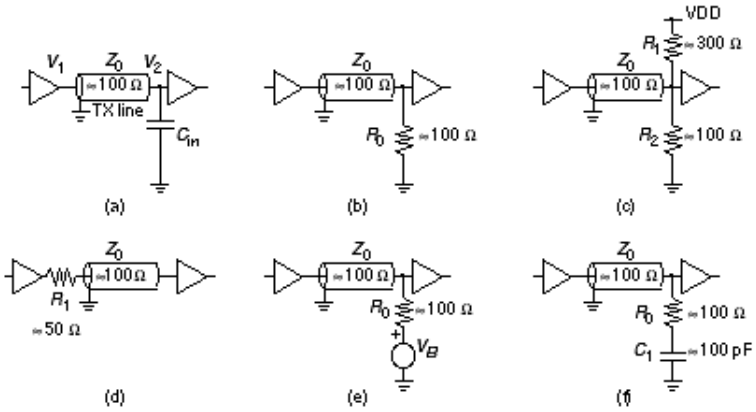


FIGURE 6.10 Transmission line termination. (a) Open-circuit or capacitive termination. (b) Parallel resistive termination. (c) Thévenin termination. (d) Series termination at the source. (e) Parallel termination using a voltage bias. (f) Parallel termination with a series capacitor.

Until recently most bus protocols required strong bipolar or BiCMOS output buffers capable of driving all the way between logic levels. The PCI standard uses weaker CMOS drivers that rely on reflection from the end of the bus to allow the intermediate receivers to see the full logic value. Many FPGA vendors now offer complete PCI functions that the ASIC designer can "drop in" to an FPGA [PCI, 1995].

An alternative to using a transmission line that operates across the full swing of the supply voltage is to use current-mode signaling or differential signals with low-voltage swings. These and other techniques are used in specialized bus structures and in high-speed DRAM. Examples are Rambus, and Gunning transistor logic (GTL). These are analog rather than digital circuits, but ASIC methods apply if the interface circuits are available as cells, hiding some of the complexity from the designer. For example, Rambus offers a Rambus access cell (RAC) for standard-cell design (but not yet for an FPGA). Directions to more information on these topics are in the bibliography at the end of this chapter.

---

1. 1994 data book, p. 2-159.

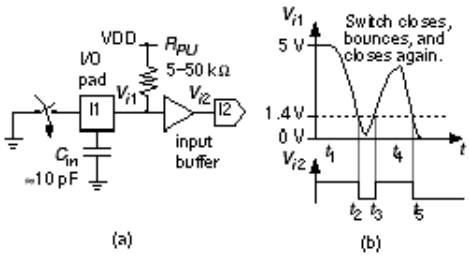
2. Application Note XAPP 024.000, Additional XC3000 Data, 1994 data book p. 8-15.

6.3 DC Input

Suppose we have a pushbutton switch connected to the input of an FPGA as shown in Figure 6.11 (a). Most FPGA input pads are directly connected to a buffer. We need to ensure that the input of this buffer never floats to a voltage between valid logic levels (which could cause both n -channel and p -channel transistors in the buffer to turn on, leading to oscillation or excessive power dissipation) and so we use the optional pull-up resistor (usually about 100 k W) that is available on many FPGAs (we could also connect a 1 k W pull-up or pull-down resistor externally).

Contacts may bounce as a switch is operated (Figure 6.11 b). In the case of a Xilinx XC4000 the effective pull-up resistance is 5-50 k W (since the specified pull-up current is between 0.2 and 2.0 mA) and forms an RC time constant with the parasitic capacitance of the input pad and the external circuit. This time constant (typically hundreds of nanoseconds) will normally be much less than the time over which the contacts bounce (typically many milliseconds). The buffer output may thus be a series of pulses extending for several milliseconds. It is up to you to deal with this in your logic. For example, you may want to debounce the waveform in Figure 6.11 (b) using an SR flip-flop.

FIGURE 6.11 A switch input. (a) A pushbutton switch connected to an input buffer with a pull-up resistor. (b) As the switch bounces several pulses may be generated.



A bouncing switch may create a noisy waveform in the time domain, we may also have noise in the voltage level of our input signal. The Schmitt-trigger inverter in Figure 6.12 (a) has a lower switching threshold of 2 V and an upper switching threshold of 3 V. The difference between these thresholds is the hysteresis , equal to 1 V in this case. If we apply the noisy waveform shown in Figure 6.12 (b) to an inverter with no hysteresis, there will be a glitch at the output, as shown in Figure 6.12 (c). As long as the noise on the waveform does not exceed the hysteresis, the Schmitt-trigger inverter will produce the glitch-free output of Figure 6.12 (d).

Most FPGA input buffers have a small hysteresis (the 200 mV that Xilinx uses is a typical figure) centered around 1.4 V (for compatibility with TTL), as shown in Figure 6.12 (e). Notice that the drawing inside the symbol for a Schmitt trigger looks like the transfer characteristic for a buffer, but is backward for an inverter. Hysteresis in the input buffer also helps prevent oscillation and noise problems with inputs that have slow rise times, though most FPGA manufacturers still have a restriction that input signals must have a rise time faster than several hundred nanoseconds.

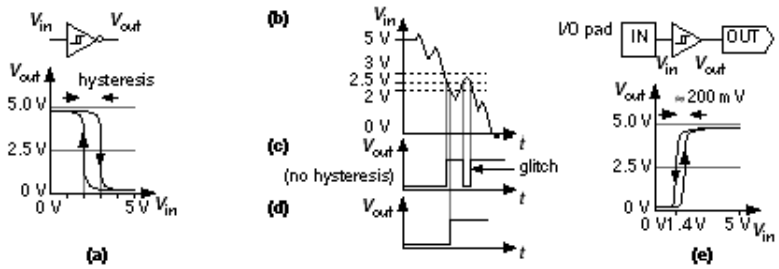


FIGURE 6.12 DC input. (a) A Schmitt-trigger inverter. (b) A noisy input signal. (c) Output from an inverter with no hysteresis. (d) Hysteresis helps prevent glitches. (e) A typical FPGA input buffer with a hysteresis of 200 mV centered around a threshold of 1.4 V.

6.3.1 Noise Margins

Figure 6.13 (a) and (b) show the worst-case DC transfer characteristics of a CMOS inverter. Figure 6.13 (a) shows a situation in which the process and device sizes create the lowest possible switching threshold. We define the maximum voltage that will be recognized as a ’0’ as the point at which the gain (V_{out} / V_{in}) of the inverter is -1. This point is V_{ILmax} = 1V in the example shown in Figure 6.13 (a). This means that any input voltage that is lower than 1V will definitely be recognized as a ’0’, even with the most unfavorable inverter characteristics. At the other worst-case extreme we define the minimum

voltage that will be recognized as a ’1’ as V_{IHmin} = 3.5V (for the example in Figure 6.13 b).

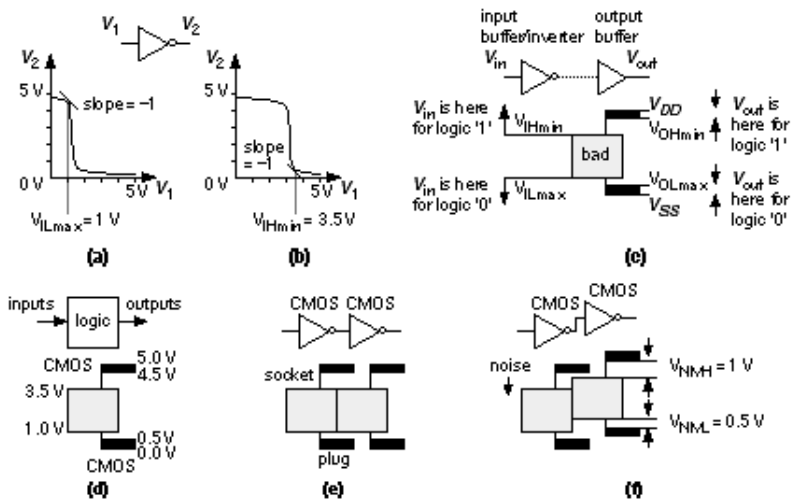


FIGURE 6.13 Noise margins. (a) Transfer characteristics of a CMOS inverter with the lowest switching threshold. (b) The highest switching threshold. (c) A graphical representation of CMOS logic thresholds. (d) Logic thresholds at the inputs and outputs of a logic gate or an ASIC. (e) The switching thresholds viewed as a plug and socket. (f) CMOS plugs fit CMOS sockets and the clearances are the noise margins.

Figure 6.13 (c) depicts the following relationships between the various voltage levels at the inputs and outputs of a logic gate:

- A logic ’1’ output must be between V_{OHmin} and V_{DD} .
- A logic ’0’ output must be between V_{SS} and V_{OLmax} .
- A logic ’1’ input must be above the high-level input voltage , V_{IHmin} .
- A logic ’0’ input must be below the low-level input voltage , V_{ILmax} .
- Clamp diodes prevent an input exceeding V_{DD} or going lower than V_{SS} .

The voltages, V_{OHmin} , V_{OLmax} , V_{IHmin} , and V_{ILmax} , are the logic thresholds for a technology. A logic signal outside the areas bounded by these logic thresholds is "bad"-an unrecognizable logic level in an electronic no-man’s land. Figure 6.13 (d) shows typical logic thresholds for a CMOS-compatible FPGA. The V_{IHmin} and V_{ILmax} logic thresholds come from measurements in Figure 6.13 (a) and (b) and V_{OHmin} and V_{OLmax} come from the measurements shown in Figure 6.2 (c).

Figure 6.13 (d) illustrates how logic thresholds form a plug and socket for any gate, group of gates, or even a chip. If a plug fits a socket, we can connect the two components together and they will have compatible logic levels. For example, Figure 6.13 (e) shows that we can connect two CMOS gates or chips together.

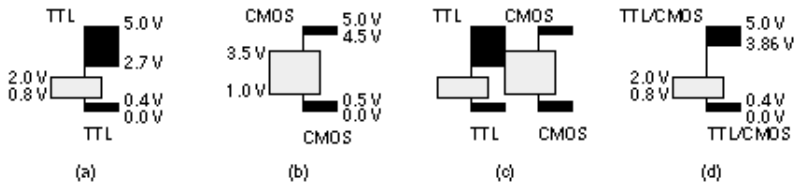


FIGURE 6.14 TTL and CMOS logic thresholds. (a) TTL logic thresholds. (b) Typical CMOS logic thresholds. (c) A TTL plug will not fit in a CMOS socket. (d) Raising V_{OHmin} solves the problem.

Figure 6.13 (f) shows that we can even add some noise that shifts the input levels and the plug will still fit into the socket. In fact, we can shift the plug down by exactly $V_{OHmin} - V_{IHmin}$ (4.5 - 3.5 = 1 V) and still maintain a valid '1'. We can shift the plug up by $V_{ILmax} - V_{OLmax}$ (1.0 - 0.5 = 0.5 V) and still maintain a valid '0'. These clearances between plug and socket are the noise margins :

$$V_{NMH} = V_{OHmin} - V_{IHmin} \text{ and } V_{NML} = V_{ILmax} - V_{OLmax} \cdot \quad (6.1)$$

For two logic systems to be compatible, the plug must fit the socket. This requires both the high-level noise margin (V_{NMH}) and the low-level noise margin (V_{NML}) to be positive. We also want both noise margins to be as large as possible to give us maximum immunity from noise and other problems at an interface.

Figure 6.14 (a) and (b) show the logic thresholds for TTL together with typical CMOS logic thresholds. Figure 6.14 (c) shows the problem with trying to plug a TTL chip into a CMOS input level-the lowest permissible TTL output level, V_{OHmin} = 2.7 V, is too low to be recognized as a logic '1' by the CMOS input. This is fixed by most FPGA manufacturers by raising V_{OHmin} to around 3.8-4.0 V (Figure 6.14 d). Table 6.1 lists the logic thresholds for several FPGAs.

6.3.2 Mixed-Voltage Systems

To reduce power consumption and allow CMOS logic to be scaled below 0.5 m m it is necessary to reduce the power supply voltage below 5 V. The JEDEC 8 [JEDEC I/O] series of standards sets the next lower supply voltage as 3.3 ± 0.3 V. Figure 6.15 (a) and (b) shows that the 3 V CMOS I/O logic-thresholds can be made compatible with 5 V systems. Some FPGAs can operate on both 3 V and 5 V supplies, typically using one voltage for internal (or core) logic, V_{DDint} and another for the I/O circuits, $V_{DDI/O}$ (Figure 6.15 c).

TABLE 6.1 FPGA logic thresholds.

I/O options		Input levels		Output levels (high current)				Output levels (low current)			
		V_{IH}	V_{IL}	V_{OH}	I_{OH}	V_{OL}	I_{OL}	V_{OH}	I_{OH}	VOL	I_{OL}
Input	Output	(min)	(max)	(min)	(max)	(max)	(max)	(min)	(max)	(max)	(max)

XC3000 [1]	TTL		2.0	0.8	3.86	-4.0	0.40	4.0				
	CMOS		3.85 [2]	0.9 [3]	3.86	-4.0	0.40	4.0				
XC3000L			2.0	0.8	2.40	-4.0	0.40	4.0	2.80 [4]	-0.1	0.2	0.1
XC4000 [5]			2.0	0.8	2.40	-4.0	0.40	12.0				
XC4000H 6	TTL	TTL	2.0	0.8	2.40	-4.0	0.50	24.0				
	CMOS	CMOS	3.85 [2]	0.9 [3]	4.00 [7]	-1.0	0.50	24.0				
XC8100 [8]	TTL	R	2.0	0.8	3.86	-4.0	0.50	24.0				
	CMOS	C	3.85 [2]	0.9 [3]	3.86	-4.0	0.40	4.0				
ACT 2/3			2.0	0.8	2.4	-8.0	0.50	12.0	3.84	-4.0	0.33	6.0
FLEX10k [9]		3V/5V	2.0	0.8	2.4	-4.0	0.45	12.0				

There is one problem when we mix 3 V and 5 V supplies that is shown in Figure 6.15 (d). If we apply a voltage to a chip input that exceeds the power supply of a chip, it is possible to power a chip inadvertently through the clamp diodes. In the worst case this may cause a voltage as high as 2.5 V (= 5.5 V - 3.0 V) to appear across the clamp diode, which will cause a very large current (several hundred milliamperes) to flow. One way to prevent damage is to include a series resistor between the chips, typically around 1 k W . This solution does not work for all chips in all systems. A difficult problem in ASIC I/O design is constructing 5 V-tolerant I/O . Most solutions may never surface (there is little point in patenting a solution to a problem that will go away before the patent is granted).

Similar problems can arise in several other situations:

- when you connect two ASICs with "different" 5 V supplies;
- when you power down one ASIC in a system but not another, or one ASIC powers down faster than another;
- on system power-up or system reset.

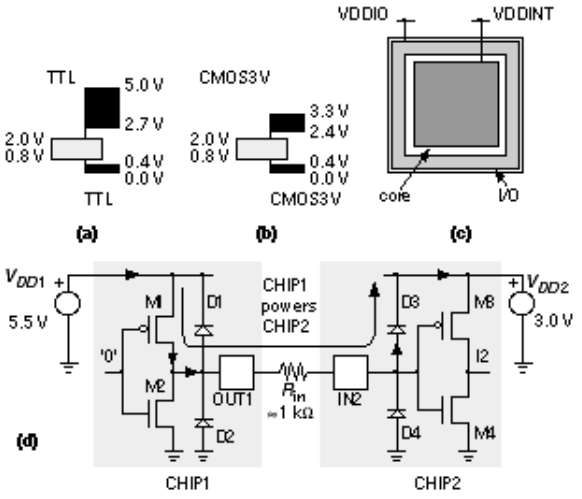


FIGURE 6.15 Mixed-voltage systems. (a) TTL levels. (b) Low-voltage CMOS levels. (c) A mixed-voltage ASIC. (d) A problem when connecting two chips with different supply voltages-caused by the input clamp diodes.

1. XC2000, XC3000/A have identical thresholds. XC3100/A thresholds are identical to XC3000 except for ±8 mA source-sink current. XC5200 thresholds are identical to XC3100A.

2. Defined as 0.7 V_{DD} , calculated with $V_{DD\ max}$ = 5.5 V.

3. Defined as 0.2 V_{DD} , calculated with $V_{DD\ min}$ = 4.5 V.

4. Defined as V_{DD} - 0.2 V, calculated with $V_{DD\ min}$ = 3.0 V.

5. XC4000, XC4000A have identical I/O thresholds except XC4000A has -24 mA sink current.

6. XC4000H/E have identical I/O thresholds except XC4000E has -12 mA sink current. Options are independent.

7. Defined as VDD - 0.5 V, calculated with VDD_{min} = 4.5 V.

8. Input and output options are independent.

9. MAX 9000 has identical thresholds to FLEX 10k.

Note: All voltages in volts, all currents in milliamperes.

6.4 AC Input

Suppose we wish to connect an input bus containing sampled data from an analog-to-digital converter (A/D) that is running at a clock frequency of 100 kHz to an FPGA that is running from a system clock on a bus at 10 MHz (a NuBus). We are to perform some filtering and calculations on the sampled data before placing it on the NuBus. We cannot just connect the A/D output bus to our FPGA, because we have no idea when the A/D data will change. Even though the A/D data rate (a sample every 10 m s or every 100 NuBus clock cycles) is much lower than the NuBus clock, if the data happens to arrive just before we are due to place an output on the NuBus, we have no time to perform any calculations. Instead we want to register the data at the input to give us a whole NuBus clock cycle (100 ns) to perform the calculations. We know that we should have the A/D data at the flip-flop input for at least the flip-flop setup time before the NuBus clock edge. Unfortunately there is no way to guarantee this; the A/D converter clock and the NuBus clock are completely independent. Thus it is entirely possible that every now and again the A/D data will change just before the NuBus clock edge.

6.4.1 Metastability

If we change the data input to a flip-flop (or a latch) too close to the clock edge (called a setup or hold-time violation), we run into a problem called metastability , illustrated in Figure 6.16. In this situation the flip-flop cannot decide whether its output should be a ’1’ or a ’0’ for a long time. If the flip-flop makes a decision, at a time t_r after the clock edge, as to whether its output is a ’1’ or a ’0’,

there is a small, but finite, probability that the flip-flop will decide the output is a ’1’ when it should have been a ’0’ or vice versa. This situation, called an upset , can happen when the data is coming from the outside world and the flip-flop can’t determine when it will arrive; this is an asynchronous signal , because it is not synchronized to the chip clock.

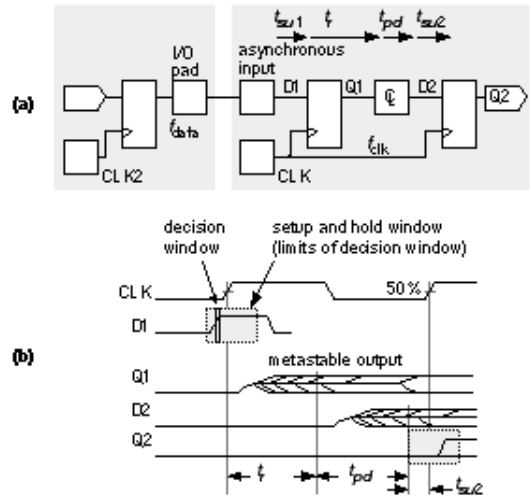


FIGURE 6.16 Metastability. (a) Data coming from one system is an asynchronous input to another. (b) A flip-flop has a very narrow decision window bounded by the setup and hold times. If the data input changes inside this decision window, the output may be metastable-neither ’1’ or ’0’.

Experimentally we find that the probability of upset , p , is

$$p = T_0 \exp - t_r / t_c \text{ , } (6.2)$$

(per data event, per clock edge, in one second, with units $Hz^{-1} \cdot Hz^{-1} \cdot s^{-1}$) where t_r is the time a sampler (flip-flop or latch) has to resolve the sampler output; T_0 and t_c are constants of the sampler circuit design. Let us see how serious this problem is in practice. If t_r = 5 ns, t_c = 0.1 ns, and T_0 = 0.1 s, Eq. 6.2 gives the upset probability as

$$p = 0.1 \exp \frac{-5 \times 10^{-19}}{0.1 \times 10^{-9}} = 2 \times 10^{-23} \text{ s , } (6.3)$$

which is very small, but the data and clock may be running at several MHz, causing the sampler plenty of opportunities for upset.

The mean time between upsets (MTBU , similar to MTBF-mean time between failures) is

$$MTBU = \frac{1}{p} = \frac{\exp t_r / t_c}{p} \text{ , } (6.4)$$

$$pf_{clock} f_{data} \qquad f_{clock} f_{data}$$

where f_{clock} is the clock frequency and f_{data} is the data frequency.

If t_r = 5 ns, t_c = 0.1 ns, T_0 = 0.1 s (as in the previous example), f_{clock} = 100 MHz, and f_{data} = 1 MHz, then

$$\text{MTBU} = \frac{\exp(5 \times 10^{-9} / 0.1 \times 10^{-9})}{(100 \times 10^{6})(1 \times 10^{6})(0.1)} = 5.2 \times 10^{8} \text{ seconds}, \quad (6.5)$$

or about 16 years (10^8 seconds is three years, and a day is 10^5 seconds). An MTBU of 16 years may seem safe, but suppose we have a 64-bit input bus using 64 flip-flops. If each flip-flop has an MTBU of 16 years, our system-level MTBF is three months. If we ship 1000 systems we would have an average of 10 systems failing every day. What can we do?

The parameter t_c is the inverse of the gain-bandwidth product , GB , of the sampler at the instant of sampling. It is a constant that is independent of whether we are sampling a positive or negative data edge. It may be determined by a small-signal analysis of the sampler at the sampling instant or by measurement. It cannot be determined by simulating the transient response of the flip-flop to a metastable event since the gain and bandwidth both normally change as a function of time. We cannot change t_c .

The parameter T_0 (units of time) is a function of the process technology and the circuit design. It may be different for sampling a positive or negative data edge, but normally only one value of T_0 is given. Attempts have been made to calculate T_0 and to relate it to a physical quantity. The best method is by measurement or simulation of metastable events. We cannot change T_0 .

Given a good flip-flop or latch design, t_c and T_0 should be similar for comparable CMOS processes (so, for example, all 0.5 m m processes should have approximately the same t_c and T_0). The only parameter we can change when using a flip-flop or latch from a cell library is t_r , and we should allow as much resolution time as we can after the output of a latch before the signal is clocked again. If we use a flip-flop constructed from two latches in series (a master-slave design), then we are sampling the data twice. The resolution time for the first sample t_r is fixed, it is half the clock cycle (if the clock is high and low for equal times-we say the clock has a 50 percent duty cycle , or equal mark-space ratio). Using such a flip-flop we need to allow as much time as we can before we clock the second sample by connecting two flip-flops in series, without any combinational logic between them, if possible. If you are really in trouble, the next step is to divide the clock so you can extend the resolution time even further.

TABLE 6.2 Metastability parameters for FPGA flip-flops. These figures are not guaranteed by the vendors.

T / s t / s

FPGA	T_0 / s	t_c / s
Actel ACT 1	1.0E-09	2.17E-10
Xilinx XC3020-70	1.5E-10	2.71E-10
QuickLogic QL12x16-0	2.94E-11	2.91E-10
QuickLogic QL12x16-1	8.38E-11	2.09E-10
QuickLogic QL12x16-2	1.23E-10	1.85E-10
Xilinx XC8100	2.15E-12	4.65E-10
Xilinx XC8100 synchronizer	1.59E-17	2.07E-10
Altera MAX 7000	2.98E-17	2.00E-10
Altera FLEX 8000	1.01E-13	7.89E-11

Sources: Actel April 1992 data book, p. 5-1, gives C1 = $T_0 = 10^{-9}$ Hz $^{-1}$, C2 = 1/ t_c = 4.6052 ns $^{-1}$, or t_c = 2.17E-10 s and T_0 = 1.0E-09 s. Xilinx gives K1 = T_0 = 1.5E-10 s and K2 = 1/ t_c = 3.69E9 s-1, t_c = 2.71E-10 s, for the XC3020-70 (p. 8-20 of 1994 data book). QuickLogic pASIC 1 QL12X16: t_c = 0.2 ns to 0.3 ns, T_0 = 0.3E-10 s to 1.2E-10 s (1994 data book, p. 5-25, Fig. 2). Xilinx XC8100 data, t_c = 4.65E-10 s and T_0 = 2.15E-12 s, is from October 1995 (v. 1.0) data sheet, Fig.17 (the XC8100 was discontinued in August 1996). Altera 1995 data book p. 437, Table 1.

Table 6.2 shows flip-flop metastability parameters and Figure 6.17 graphs the metastability data for f_{clock} = 10 MHz and f_{data} = 1 MHz. From this graph we can see the enormous variation in MTBF caused by small variations in t_c . For example, in the QuickLogic pASIC 1 series the range of T_0 from 0.3 to 1.2 ¥ 10^{-10} s is 4:1, but it is the range of t_c = 0.2 - 0.3 ns (a variation of only 1:1.5) that is responsible for the enormous variation in MTBF (nearly four orders of magnitude at t_r = 5 ns). The variation in t_c is caused by the variation in GB between the QuickLogic speed grades. Variation in the other vendors' parts will be similar, but most vendors do not show this information. To be safe, build a large safety margin for MTBF into any design-it is not unreasonable to use a margin of four orders of magnitude.

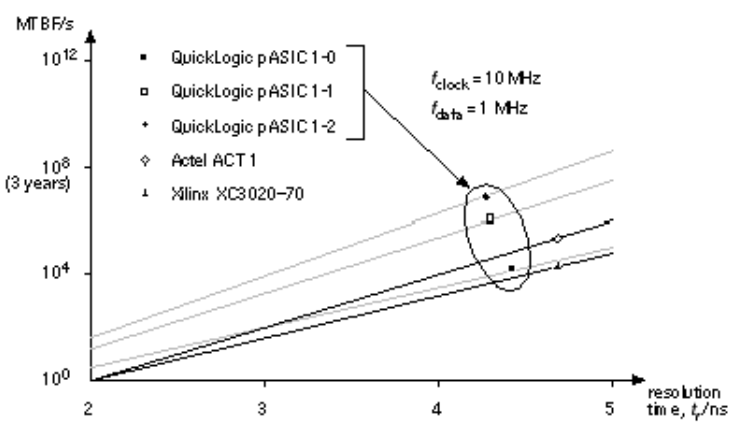


FIGURE 6.17 Mean time between failures (MTBF) as a function of resolution time. The data is from FPGA vendors' data books for a single flip-flop with clock frequency of 10 MHz and a data input frequency of 1 MHz (see Table 6.2).

Some cell libraries include a synchronizer , built from two flip-flops in cascade, that greatly reduces the effective values of t_c and T_0 over a single flip-flop. The penalty is an extra clock cycle of latency.

To compare discrete TTL parts with ASIC flip-flops, the 74AS4374 TTL metastable-hardened dual flip-flops , from TI, have t_c = 0.42 ns and T_0 = 4 ns. The parameter T_0 ranges from about 10 s for the 74LS74 (a regular flip-flop) to 4 ns for the 74AS4374 (over nine orders of magnitude different); t_c only varies from 0.42 ns (74AS374) to 1.3 ns (74LS74), but this small variation in t_c is just as important.

6.5 Clock Input

When we bring the clock signal onto a chip, we may need to adjust the logic level (clock signals are often driven by TTL drivers with a high current output capability) and then we need to distribute the clock signal around the chip as it is needed. FPGAs normally provide special clock buffers and clock networks. We need to minimize the clock delay (or latency), but we also need to minimize the clock skew.

6.5.1 Registered Inputs

Some FPGAs provide a flip-flop or latch that you can use as part of the I/O circuit (registered I/O). For other FPGAs you have to use a flip-flop or latch using the basic logic cell in the core. In either case the important parameter is the input setup time. We can measure the setup with respect to the clock signal at the flip-flop or the clock signal at the clock input pad. The difference between these two parameters is the clock delay.

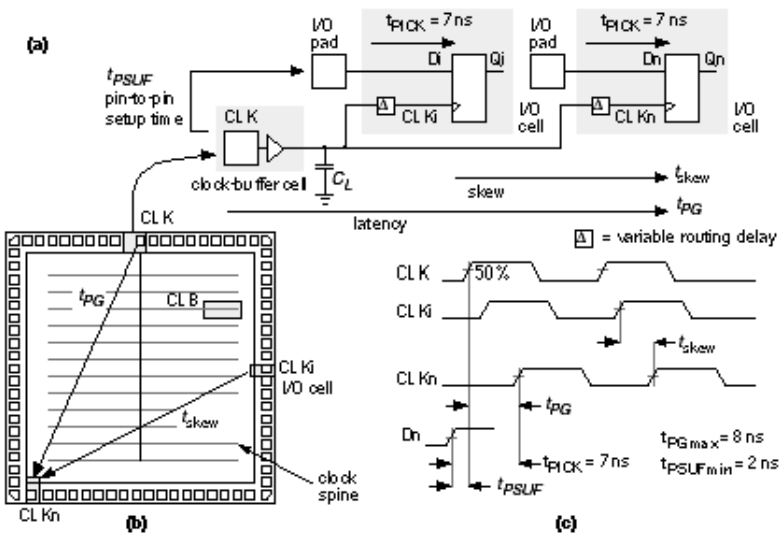


FIGURE 6.18 Clock input. (a) Timing model with values for a Xilinx XC4005-6. (b) A simplified view of clock distribution. (c) Timing diagram. Xilinx eliminates the variable internal delay t_{PG}, by specifying a pin-to-pin setup time, $t_{PSUFmin}$ = 2 ns.

Figure 6.18 shows part of the I/O timing model for a Xilinx XC40005-6. [1]

- t_{PICK} is the fixed setup time for a flip-flop relative to the flip-flop clock.
- t_{skew} is the variable clock skew , the signed delay between two clock edges.
- t_{PG} is the variable clock delay or latency .

To calculate the flip-flop setup time ($t_{PSUFmin}$) relative to the clock pad (which is the parameter system designers need to know), we subtract the clock delay, so that

$t_{PSUF} = t_{PICK} - t_{PG}$. (6.6)

The problem is that we cannot easily calculate t_{PG} , since it depends on the clock distribution scheme and where the flip-flop is on the chip. Instead Xilinx specifies $t_{PSUFmin}$ directly, measured from the data pad to the clock pad; this time is called a pin-to-pin timing parameter . Notice $t_{PSUF\ min}$ = 2 ns ? $t_{PICK} - t_{PG\ max}$ = -1 ns.

Figure 6.19 shows that the hold time for a XC4005-6 flip-flop (t_{CKI}) with respect to the flip-flop clock is zero. However, the pin-to-pin hold time including the clock delay is t_{PHF} = 5.5 ns. We can remove this inconvenient hold-time restriction by delaying the input signal. Including a programmable delay allows Xilinx to guarantee the pin-to-pin hold time (t_{PH}) as zero. The penalty is an increase in the pin-to-pin setup time (t_{PSU}) to 21 ns (from 2 ns) for the XC4005-6, for example.

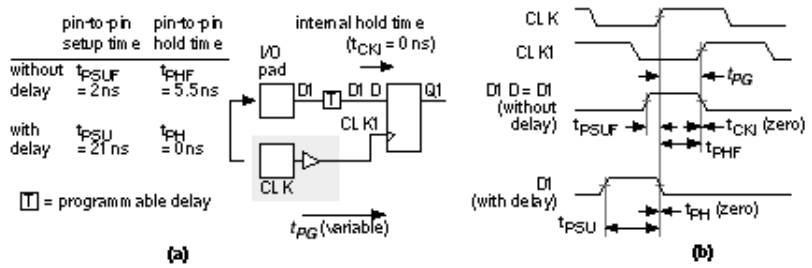
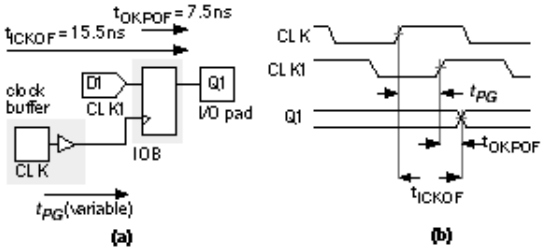


FIGURE 6.19 Programmable input delay. (a) Pin-to-pin timing model with values from an XC4005-6. (b) Timing diagrams with and without programmable delay.

We also have to account for clock delay when we register an output. Figure 6.20 shows the timing model diagram for the clock-to-output delay.

FIGURE 6.20 Registered output. (a) Timing model with values for an XC4005-6 programmed with the fast slew-rate option. (b) Timing diagram.



---

1. The Xilinx XC4005-6 timing parameters are from the 1994 data book p. 2-50 to p. 2-53.

6.6 Power Input

The last item that we need to bring onto an FPGA is the power. We may need multiple VDD and GND power pads to reduce supply bounce or separate VDD pads for mixed-voltage supplies. We may also need to provide power for on-chip programming (in the case of antifuse or EPROM programming technology). The package type and number of pins will determine the number of power pins, which, in turn, affects the number of SSOs you can have in a design.

6.6.1 Power Dissipation

As a general rule a plastic package can dissipate about 1 W, and more expensive ceramic packages can dissipate up to about 2 W. Table 6.3 shows the thermal characteristics of common packages. In a high-speed (high-power) design the ASIC power consumption may dictate your choice of packages. Actel provides a formula for calculating typical dynamic chip power consumption of their FPGAs. The formula for the ACT 2 and ACT 3 FPGAs are complex; therefore we shall use the simpler formula for

the ACT 1 FPGAs as an example [1] :

TABLE 6.3 Thermal characteristics of ASIC packages.

Package [2]	Pin count	Max. power P_{max} /W	q_{JA} /°CW $^{-1}$ (still air) [3, 4]	q_{JA} /°CW $^{-1}$ (still air) [5]
CPGA	84		33	32-38
CPGA	100		35	
CPGA	132		30	
CPGA	175		25	16
CPGA	207		22	
CPGA	257		15	
CQFP	84		40	
CQFP	172		25	
PQFP	100	1.0	55	56-75
PQFP	160	1.75	33	30-33
PQFP	208	2.0	33	27-32
VQFP	80		68	
PLCC	44		52	44
PLCC	68		45	28-35
PLCC	84	1.5	44	
PPGA	132			33-34

Total chip power = 0.2 (N ¥ F1) + 0.085 (M ¥ F2) + 0.8 (P ¥ F3) mW (6.7)

where

F1 = average logic module switching rate in MHz

F2 = average clock pin switching rate in MHz

F3 = average I/O switching rate in MHz

M = number of logic modules connected to the clock pin

N = number of logic modules used on the chip

P = number of I/O pairs used (input + output), with 50 pF load

As an example of a power-dissipation calculation, consider an Actel 1020B-2 with a 20 MHz clock. We shall initially assume 100 percent utilization of the 547 Logic Modules and assume that each switches at an average speed of 5 MHz. We shall also initially assume that we use all of the 69 I/O Modules and that

each switches at an average speed of 5 MHz. Using Eq. 6.7 , the Logic Modules dissipate

$$P_{LM} = (0.2)(547)(5) = 547 \text{ mW}, \quad (6.8)$$

and the I/O Module dissipation is

$$P_{IO} = (0.8)(69)(5) = 276 \text{ mW}. \quad (6.9)$$

If we assume the clock buffer drives 20 percent of the Logic Modules, then the additional power dissipation due to the clock buffer is

$$P_{CLK} = (0.085)(547)(0.2)(5) = 46.495 \text{ mW}. \quad (6.10)$$

The total power dissipation is thus

$$P_D = (547 + 276 + 46.5) = 869.5 \text{ mW}, \quad (6.11)$$

or about 900 mW (with an accuracy of certainly no better than ± 100 mW).

Suppose we intend to use a very thin quad flatpack (VQFP) with no cooling (because we are trying to save area and board height). From Table 6.3 the thermal resistance, q_{JA} , is approximately 68 °CW $^{-1}$ for an 80-pin VQFP. Thus the maximum junction temperature under industrial worst-case conditions (T_A = 85 °C) will be

$$T_J = (85 + (0.87)(68)) = 144.16 \text{ °C}, \quad (6.12)$$

(with an accuracy of no better than 10 °C). Actel specifies the maximum junction temperature for its devices as T_{Jmax} = 150 °C (T_{Jmax} for Altera is also 150 °C, for Xilinx T_{Jmax} = 125°C). Our calculated value is much too close to the rated maximum for comfort; therefore we need to go back and check our assumptions for power dissipation. At or near 100 percent module utilization is not unreasonable for an Actel device, but more questionable is that all nodes and I/Os switch at 5 MHz.

Our real mistake is trying to use a VQFP package with a high q_{JA} for a high-speed design. Suppose we use an 84-pin PLCC package instead. From Table 6.3 the thermal resistance, q_{JA} , for this alternative package is approximately 44 °CW $^{-1}$. Now the worst-case junction temperature will be a more reasonable

$$T_J = (85 + (0.87)(44)) = 123.28 \text{ °C}, \quad (6.13)$$

It is possible to estimate the power dissipation of the Actel architecture because the routing is regular and the interconnect capacitance is well controlled (it has to be since we must minimize the number of

series antifuses we use). For most other architectures it is much more difficult to estimate power dissipation. The exception, as we saw in Section 5.4 "Altera MAX," are the programmable ASICs based on programmable logic arrays with passive pull-ups where a substantial part of the power dissipation is static.

6.6.2 Power-On Reset

Each FPGA has its own power-on reset sequence. For example, a Xilinx FPGA configures all flip-flops (in either the CLBs or IOBs) as either SET or RESET. After chip programming is complete, the global SET/RESET signal forces all flip-flops on the chip to a known state. This is important since it may determine the initial state of a state machine, for example.

---

1. 1994 data book, p.1-9

2. CPGA = ceramic pin-grid array; CQFP = ceramic quad flatpack; PQFP = plastic quad flatpack; VQFP = very thin quad flatpack; PLCC = plastic leaded chip carrier; PPGA = plastic pin-grid array.

3. q_{JA} varies with die size.

4. Data from Actel 1994 data book p. 1-9, p. 1-45, and p. 1-94.

5. Data from Xilinx 1994 data book p. 4-26 and p. 4-27.

6.7 Xilinx I/O Block

The Xilinx I/O cell is the input/output block (IOB) . Figure 6.21 shows the Xilinx XC4000 IOB, which is similar to the IOB in the XC2000, XC3000, and XC5200 but performs a superset of the options in these other Xilinx FPGAs.

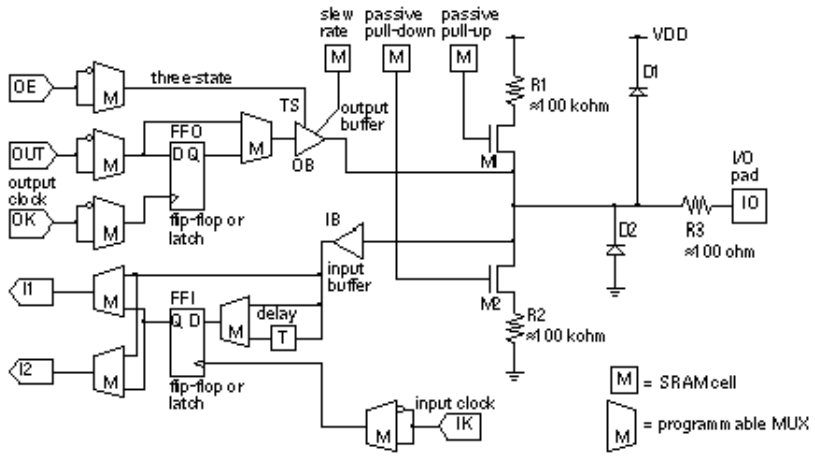


FIGURE 6.21 The Xilinx XC4000 family IOB (input/output block). (Source: Xilinx.)

The outputs contain features that allow you to do the following:

- Switch between a totem-pole and a complementary output (XC4000H).
- Include a passive pull-up or pull-down (both n -channel devices) with a typical resistance of about 50 k W .
- Invert the three-state control (output enable OE or three-state, TS).
- Include a flip-flop, or latch, or a direct connection in the output path.
- Control the slew rate of the output.

The features on the inputs allow you to do the following:

- Configure the input buffer with TTL or CMOS thresholds.
- Include a flip-flop, or latch, or a direct connection in the input path.
- Switch in a delay to eliminate an input hold time.

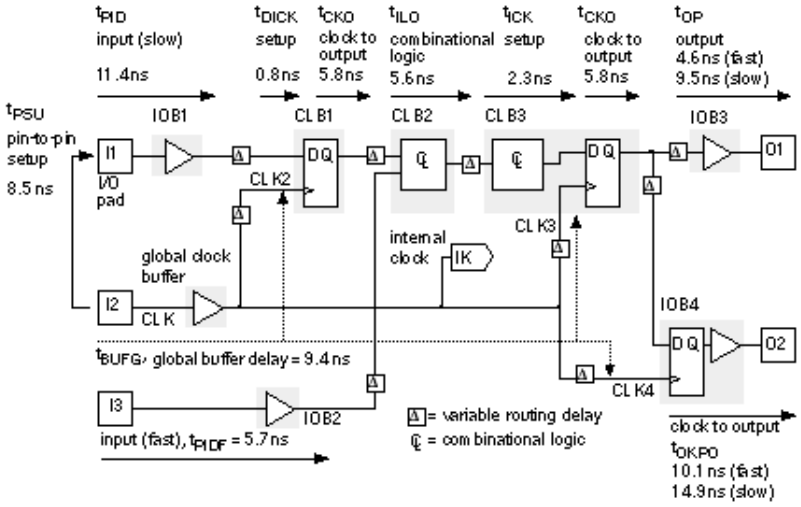


FIGURE 6.22 The Xilinx LCA (Logic Cell Array) timing model. The paths show different uses of CLBs (Configurable Logic Blocks) and IOBs (Input/Output Blocks). The parameters shown are for an XC5210-6. (Source: Xilinx.)

Figure 6.22 shows the timing model for the XC5200 family. [1] It is similar to the timing model for all the other Xilinx LCA FPGAs with one exception-the XC5200 does not have registers in the I/O cell; you go directly to the core CLBs to include a flip-flop or latch on an input or output.

6.7.1 Boundary Scan

Testing PCBs can be done using a bed-of-nails tester. This approach becomes very difficult with closer IC pin spacing and more sophisticated assembly methods using surface-mount technology and

multilayer boards. The IEEE implemented boundary-scan standard 1149.1 to simplify the problem of testing at the board level. The Joint Test Action Group (JTAG) developed the standard; thus the terms JTAG boundary scan or just JTAG are commonly used.

Many FPGAs contain a standard boundary-scan test logic structure with a four-pin interface. By using these four signals, you can program the chip using ISP, as well as serially load commands and data into the chips to control the outputs and check the inputs. This is a great improvement over bed-of-nails testing. We shall cover boundary scan in detail in Section 14.6, "Scan Test."

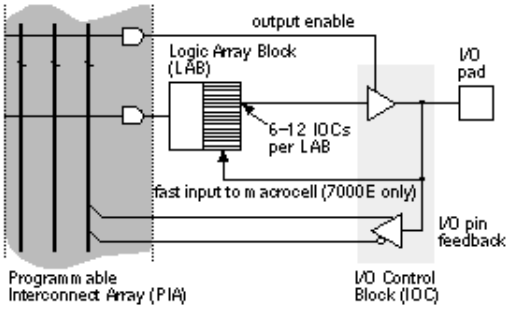
---

1. October 1995 (v. 3.0) data sheet.

6.8 Other I/O Cells

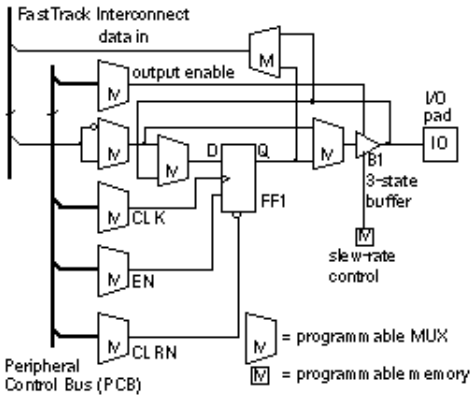
The Altera MAX 5000 and 7000 use the I/O Control Block (IOC) shown in Figure 6.23 . In the MAX 5000, all inputs pass through the chipwide interconnect. The MAX 7000E has special fast inputs that are connected directly to macrocell registers in order to reduce the setup time for registered inputs.

FIGURE 6.23 A simplified block diagram of the Altera I/O Control Block (IOC) used in the MAX 5000 and MAX 7000 series. The I/O pin feedback allows the I/O pad to be isolated from the macrocell. It is thus possible to use a LAB without using up an I/O pad (as you often have to do using a PLD such as a 22V10). The PIA is the chipwide interconnect.



The FLEX 8000 and 10k use the I/O Element (IOE) shown in Figure 6.24 (the MAX 9000 IOC is similar). The interface to the IOE is directly to the chipwide interconnect rather than the core logic. There is a separate bus, the Peripheral Control Bus , for the IOE control signals: clock, preset, clear, and output enable.

FIGURE 6.24 A simplified block diagram of the Altera I/O Element (IOE), used in the FLEX 8000 and 10k series. The MAX 9000 IOC (I/O Cell) is similar. The FastTrack Interconnect bus is the chipwide interconnect. The PCB is used for control signals common to each IOE.



The AMD MACH 5 family has some I/O features not currently found on other programmable ASICs. The MACH 5 family has 3.3 V and 5 V versions that are both suitable for mixed-voltage designs. The 3 V versions accept 5 V inputs, and the outputs of the 3 V versions do not drive above 3.3 V. You can apply a voltage up to 5.5 V to device inputs before you connect VDD (this is known as hot insertion

6.9 Summary

Among the options available in I/O cells are: different drive strengths, TTL-compatibility, registered or direct inputs, registered or direct outputs, pull-up resistors, over-voltage protection, slew-rate control, and boundary-scan. Table 6.4 shows a list of features. Interfacing an ASIC with a system starts at the outputs where you check the voltage levels first, then the current levels. Table 6.5 is a look-up table for Tables 6.6 and 6.7 , which show the I/O resources present in each type of programmable ASIC (using the abbreviations of Table 6.4).

TABLE 6.4 I/O options for programmable ASICs.

Code [1]	I/O Option	Function
IT/C	TTL/CMOS input	Programmable input buffer threshold
OT/C	TTL/CMOS output	Complementary or totem-pole output
nSNK	Sink capability	Maximum current sink ability (e.g., 12SNK is I_0 = 12 mA sink)
nSRC	Source capability	Maximum current source ability (e.g., 12SRC is I_0 = -12 mA source)
5/3	5V/3V	Separate I/O and core voltage supplies
OD	Open drain/collector	Programmable open-drain at the output buffer
TS	Three-state	Output buffer with three-state control
SR	Slew-rate control	Fast or slew-rate limited output buffer to reduce ground bounce
PD	Pull-down	Programmable pull-down device or resistor at the I/O pad
PU	Pull-up	Programmable pull-up device or resistor at the I/O pad
EP	Enable polarity	Driver control can be positive (three-state) or negative (enable).
RI	Registered input	Inputs may be registered in I/O cell.

RO	Registered output	Outputs may be registered in I/O cell.
RIO	Registered I/O	Both inputs and outputs may be registered in I/O cell.
ID	Input delay	Input delay to eliminate input hold time
JTAG	JTAG	Boundary-scan test
SCH	Schmitt trigger	Schmitt trigger or input hysteresis
HOT	Hot insertion	Inputs protected from hot insertion
PCI	PCI compliant	Output buffer characteristics comply with PCI specifications.

Important points that we covered in this chapter are the following:

- Outputs can typically source or sink 5-10 mA continuously into a DC load, and 50-200 mA transiently into an AC load.
- Input buffers can be CMOS (threshold at 0.5 V_{DD}) or TTL (1.4 V).
- Input buffers normally have a small hysteresis (100-200 mV).
- CMOS inputs must never be left floating.
- Clamp diodes to GND and VDD are present on every pin.
- Inputs and outputs can be registered or direct.
- I/O registers can be in the I/O cell or in the core.
- Metastability is a problem when working with asynchronous inputs.

---

1. These codes are used in Tables 6.6 and 6.7 .