12.9.2 Intercard connections

With logic signals going between circuit cards, the opportunities for trouble multiply rapidly. There is greater wiring capacitance, as well as longer ground paths through cables, connectors, card extenders, etc., so the ground spikes induced by drive currents during logic transitions are generally larger and more troublesome. It is best to avoid sending clocking signals with large fan-out between boards, if possible, and the ground connections to the individual cards should be robust. For fast signals (characteristic times of order a few nanoseconds, or less) the interconnections must be treated as constant-impedance *transmission lines* (see §12.10.1 and Appendix H), which may be single-ended (coaxial cable) or differential (twisted pair). We'll have much more to say about this soon (§12.10).

If clocking signals are sent between boards, it is important to use a gate (for single-ended signals) or a differential receiver (for differential signals such as LVDS) as an input buffer on each board. In some cases it may be best to use line driver and receiver chips, as we will discuss shortly. In any case, it is best to try to keep critical circuits together on one card, where you can control the inductance of the ground paths and keep wiring capacitance at a minimum. Fast signals (edge times of order 1 ns or less), and especially clocking signals, that go between the circuits on a single card will often be routed as "stripline" or "microstrip" transmission lines (§*1x.1.3*).[58] This can take the form of a single-ended trace above a ground plane (microstrip) or between ground planes (stripline); or it can be a differential pair, with two traces side-by-side or stacked vertically. Such constant-impedance routes will be terminated in their characteristic impedance, commonly 50 Ω (single ended) or 100 Ω (differential), either with "back termination" at the driving end, or with far-end termination, or both. The problems you'll encounter in sending fast signals around through several cards should not be underestimated; they can turn out to be the major headache of an entire project!

12.10 Driving Cables

You can't run digital signals from one instrument to another by just stringing a single conductor between them, because such an arrangement is prone to pickup of interference (as well as generating interference of its own), and also serious degradation of the digital signals themselves. Instead, digital signals are generally piped through coaxial cables, twisted pairs, flat ribbon cables (sometimes with ground plane or shield), multiwire bundled cables, and, increasingly, fiber-optic cables.[59] Let's look at some of the methods used to send digital signals between boxes of electronics, since these methods constitute an important part of digital interfacing. In most cases there are special-purpose driver–receiver chips available to make your job easier.

12.10.1 Coaxial cable

If you've never dealt with fast signals going through cables, you're in for a surprise.

A. The wrong way

Here's a typical mistake that we see again and again: you've got some digital signals coming out of a "Digital I/O" interface card, for example the PCI-6509 from the popular series of data acquisition products made by National Instruments. This puppy plugs into a computer motherboard's PCI slot, and gives you 96 bits of bidirectional digital I/O, grouped in 12 bytes, each byte of which can be configured as input or output. As an output, each bit generates full-swing 5 V CMOS logic levels (i.e., 0 V and +5 V), with plenty of drive capability (24 mA sink or source), enough to drive easily loads like solid-state relays, small mechanical relays, bright LEDs, and the like.[60]

The mistake is to connect this digital output to a length of coax, and expect it to arrive safely at the far end, as in Figure 12.102A. The thinking goes like this: we've got at least ±24 mA of drive, which should be able to drive the ∼200 pF of a 2-meter length of coax (C=100 pF/m) just fine; after all, $I = C\,dV/dt$ predicts a rise time of ∼20 ns, assuming a typical switching output current of ∼40 mA into the capacitive load. So, what's the problem?

The problem is that we have to treat a coax cable as a transmission line, rather than as the low-frequency approximation of a lumped capacitance, when dealing with signals that are changing on a time scale comparable with (or shorter than) the round-trip delay time of the cable. If you try to use the logic output to drive the cable directly, you'll get a messy waveform at the far end, with overshoots and polarity reversals, producing incorrect recovery of the waveform (and even destruction of the far-end gate). But you can magically fix the problem by simply adding a 50 Ω

[58] This is mandatory, for example, with the fast logic families known as ECL-100K, ECL-100E, ECL-100EL, and ECL-100EP.

[59] Coaxial cables (affectionately called "coax") and twisted pairs are examples of *transmission lines*, discussed in detail in Appendix H.

[60] Look up the 74LVC4245A, if you want to see the specs on the voltage-translating (3 V ↔ 5 V) octal transceivers used in this product.

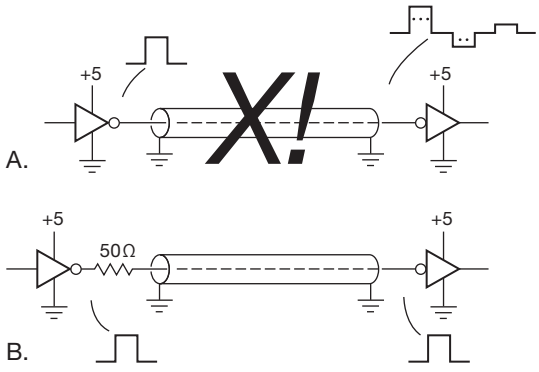

Figure 12.102. A. Driving a length of cable with a logic output produces a mangled waveform of overshoots and polarity reversals. B. Adding a ∼50 Ω series resistor effects a magic cure.

series resistor at the driving end (Figure 12.102B): the far-end waveform becomes a good replica of the driving waveform.

Let's look at this situation in more detail, beginning with the "wrong" way, then progressing through three configurations that solve the problem, each of which have advantages and disadvantages. We'll wind up with the magic fix of Figure 12.102B, known as "series termination," which is well suited to digital-logic signals.[61]

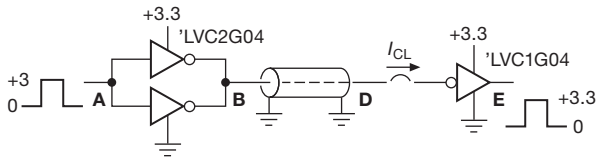

Figure 12.103. Test circuit that attempts to send digital logic signals directly through a length of cable whose far end is "unterminated". The disastrous results can be seen in Figures 12.104–12.106. *Don't do this!*

To illustrate the problem, we hooked up the circuit of Figure 12.103 and ran a pulse pattern through it. You can see the measured results in Figures 12.104–12.106. In Figure 12.104 the far end has been left unconnected: the first transition reaches the far end 12 ns later, where it bounces off the open end (with unchanged polarity), producing an output voltage nearly double the step size; things get messy as the signals bounce back and forth, reversed in polarity with each bounce off the source end, and decaying slowly with each bounce, but always adding the new source signal transitions. The signal at the far end looks terrible – and it

[61] But not to *every* kind of signal: in the world of RF and video the method of "double-ended termination" is used universally.

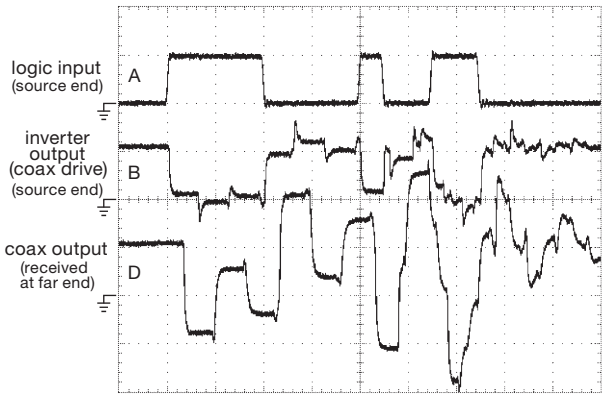

Figure 12.104. Waveforms seen in the circuit of Figure 12.103, when driven with a pulse pattern clocked at 20 ns, and with the far end of the 8 foot (2.4 m) RG-58 coax cable unconnected. The 3.3 V logic-level drive to the cable produces far-end swings of nearly 15 V peak-to-peak! Horizontal: 40 ns/div; Vertical: 3 V/div.

swings to +8 V and −6 V, even though we are driving the cable with 0 V and +3.3 V only. Imagine what will happen when we connect a logic inverter at the far end!

You needn't imagine. In Figure 12.105 we've attached the far end inverter, running also from +3.3 V. Its input protection diodes clamp the renegade waveform, limiting negative swings to roughly a diode drop, but allowing occasional positive swings to +8 V (the 'LVC1G04 has 5 V–compliant inputs, regardless of supply voltage). It's an ugly situation, and it's not surprising that the recovered logic output has some false transitions.

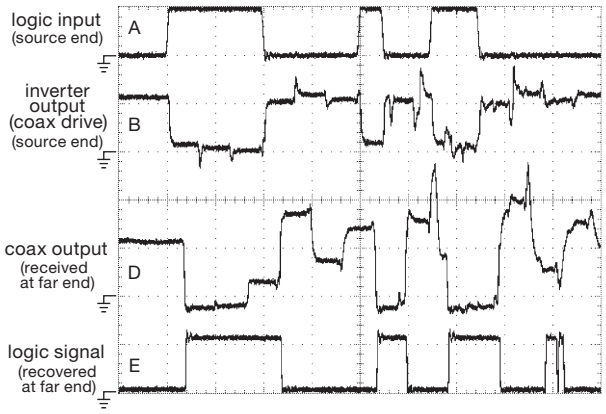

Figure 12.105. Same as Figure 12.104, but with the 'LVC1G04 inverter connected at the far end. The clamping effect of the inverter's input protection diodes reduces the swing at the coax output. This unhealthy situation produces some false transitions at the output; it can also destroy the output inverter.

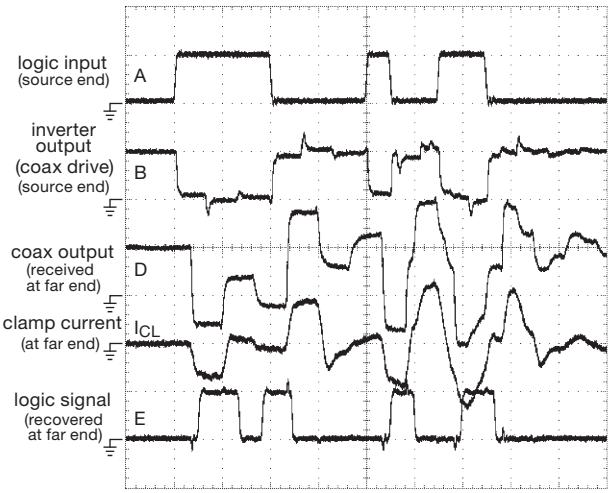

Figure 12.106. Same as Figure 12.105, but with 5 V logic ('LVC2G04 drivers, 'HCT04 far-end receiver). Note the substantial clamping currents, and the error-prone signal recovery. Horizontal: 40 ns/div; vertical: 5 V/div and 20 mA/div.

Turning now to the original scenario – a 2 m cable connecting a digital I/O card to some instrument – the driving signal comes from a hefty 5 V logic output, and the receiver (within some commercial instrument) is likely to be something like an 'HCT04 (5 V logic, with TTL threshold ∼1.4 V). Figure 12.106 shows what happens. It's not a pretty picture: once again the flailing far-end signal is partially clamped, but reaches peaks of +10 V and −5 V, with corresponding clamp currents of ±25 mA. This exceeds the "absolute maximum" input clamp current specification of ±20 mA.[62] And the recovered output is a mess. It's pretty clear that you just cannot hook things up this way.

12.10.2 The right way – I: Far-end termination

The solution is to *terminate* the cable in its *characteristic impedance* $\mathbf{Z}_0$ (see Appendix H), which for most coaxial cables is 50 Ω (resistive).[63] There are three different ways this can be arranged, namely far-end termination, double termination (both ends), and near-end termination ("series termination," "source termination," or "back termination").

Far-end termination is the easiest to understand; see Figure 12.107. The *amazing fact* about transmission lines (explained in Appendix H) is that the addition of a simple resistor of value $R = Z_0$ at the far end suppresses all reflections and, furthermore, makes the cable input look like a pure resistance equal to R. Amazing, because *all capacitance disappears*.

The good news is that this solves the problem. The bad news is that Z_0 is uncomfortably low, usually 50 Ω, which requires rather high driver output currents (20 mA per volt of drive). But you can do it, with hefty logic chips, typically several in parallel, operating with modest logic supply voltages (for example, +3.3 V or less).[64]

Figure 12.107 shows one way to do this, using several sections of 'LVC04 or 'AC04 in parallel to drive the near end of a length of 50 Ω coax that is terminated with a 50 Ω resistor at the far end. The 'LVC2G04 is fully specified at ±24 mA output current and 3 V supply; the corresponding figure for the 'AC04 is ±12 mA. So we're pushing things just a bit here, demanding +60 mA of source current. (No worry, however, if we replaced the single 50 Ω termination with a 100 Ω–100 Ω divider at the far end. That would require ±30 mA of drive, for which we could probably get away with a single 'LVC1G04.)

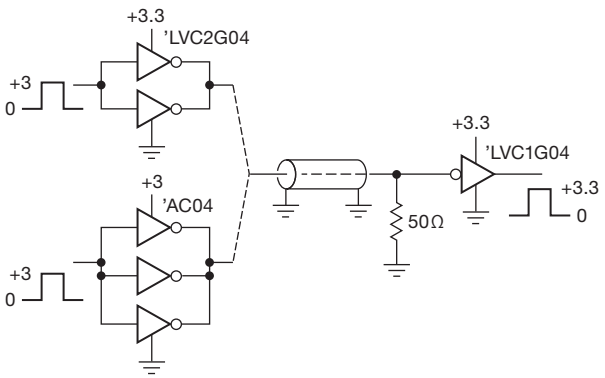

Figure 12.107. Digital logic levels driving end-terminated 50 Ω coax. Apart from losses, the received signals are full-swing replicas of the driving signal.

But it works: Figures 12.108 and 12.109 show the nice waveforms you get, in this case with a 10 m (33 foot) length, operating at the same 20 ns clocking rate. Note the clean waveforms throughout (with a small bit of ringing evident with the 'AC04 driver, probably caused by

[62] For which the datasheet warns "Stresses beyond those listed under 'absolute maximum ratings' may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated under 'recommended operating conditions' is not implied."

[63] A big exception is the video community, which has chosen 75 Ω (typified by RG-59 coax); and in pulse circuits you occasionally see 93 Ω (RG-62).

[64] A nice trick that reduces the needed drive current by a factor of 2 is to use a pair of 100 Ω resistors at the far end, configured as a divider between the supply and ground.

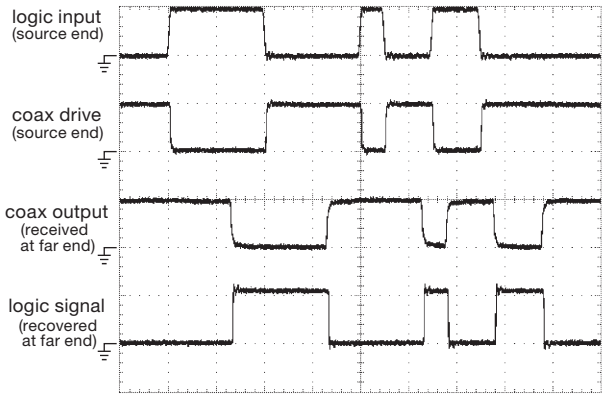

Figure 12.108. Waveforms from the circuit of Figure 12.107 ('LVC2G04 driver), with the same 20 ns clocked pulse pattern as in Figures 12.104–12.106, but with a 10 m length of coax. Horizontal: 40 ns/div; vertical: 3 V/div.

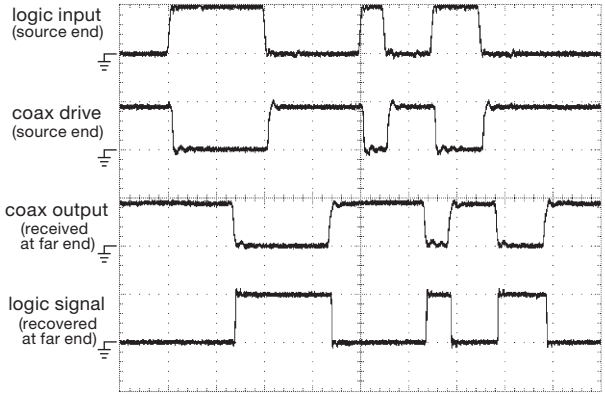

Figure 12.109. Same as Figure 12.108, but with 'AC04 driver.

the greater inductances in power and ground leads of its DIP-14 package, as compared with the compact SOT23-6 surface-mount package of the 'LVC2G04).

If you're worried about the substantial drive current needed to push full rail-to-rail logic-level swings into a 50 Ω load, and if you are not operating at maximum logic speeds, you can substitute a "MOSFET low-side driver" IC for the paralleled logic inverters we used. For example, the venerable TC4420-series of MOSFET drivers accept logic-level inputs, and generate a robust rail-to-rail output swing between ground and the supply voltage, which can range from +4.5 V to +18 V. The peak output current ranges from 1.5 A to 9 A in this series – no problem driving 50 Ω! These are dual drivers, available in many package styles (including DIP) and in inverting or noninverting flavors; they cost about a dollar in small quantity (see Table 3.8 on page 218). There are literally hundreds of MOSFET driver chips available, with some very nice choices from companies like Fairchild, IXYS, Microchip, ST, and TI, among others; most are happy to operate from 5 V supplies (where they have reduced output-drive capability, but they're still pretty muscular; the 9 A IXDD609, for example, can source or sink 2 A when powered from a 5 V supply). Don't be discouraged by the rather relaxed speed specifications on datasheets (e.g., rise and fall times of 20 ns or more), because these are usually specified into the horrendous capacitive loads (1000–10,000 pF, aided and abetted by the Miller effect) of switching power MOSFETs. You'll do much better when driving a 50 Ω far-end terminated coax, which presents a pure resistive load to the driver. For example, the IXYS IXDD509 9 A noninverting driver specifies ∼25 ns rise and fall, but that's into 10,000 pF! From the curves later in the datasheet, however, you'll discover rise and fall times of 4 ns or less for supply voltages anywhere between +5 V and +30 V.[65]

A. The right way – II: Double-ended termination

The high currents needed to drive end-terminated 50 Ω coax can be remedied somewhat by adding a series resistance at the cable's input, equal in value to its characteristic impedance (i.e., 50 Ω; see Figure 12.110). Then the driver sees a load of 100 Ω (the series resistor plus the 50 Ω seen at the cable's input). That's sometimes called "double-ended termination." It has an additional advantage, namely that any signals reflected from the far end are swallowed by the input resistor, which acts effectively as a termination for signals traveling backward.

It works, but now the output-signal amplitude is half that of the driver, because the cable's input resistance forms a voltage divider with the series resistor at the input. That's why we used 5 V logic at the input, combined with inverters at the output whose thresholds are in the range of 1.2–1.4 V: either the TTL threshold 'ACT04 (see Figure 12.111), or logic running from a lower +3 V supply (see Figure 12.112). The 'ACT04 specifies input logic levels of $<$0.8 V and $>$2.0 V; the corresponding figures for the 'LVC1G04 (powered from +3.0 V) are $<$0.8 V and $>$1.7 V.

[65] In spite of the good rise and fall times, there tend to be substantial delay times, regardless of load capacitance, at low supply voltages; for this IXYS part, the delay times are of order 30 ns with a +5 V supply, dropping to half that at V_S=10 V. From a practical point of view, this limits operation to speeds of a few megahertz. A more serious problem at high frequencies is the rising rail-to-rail supply current, and consequent power dissipation, at frequencies above a megahertz that are due to the high internal capacitance of the large output MOS devices. Some parts show plots of no-load supply current versus frequency in their datasheets.

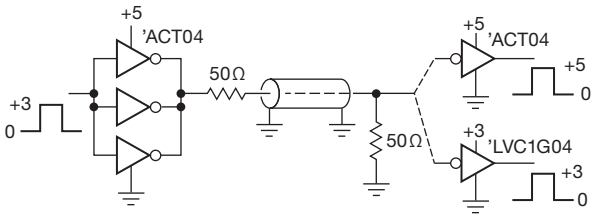

Figure 12.110. Digital logic driving double-end terminated 50 Ω coax. The received signal amplitude is half that of the driver logic output, therefore ~2.5 V in this example.

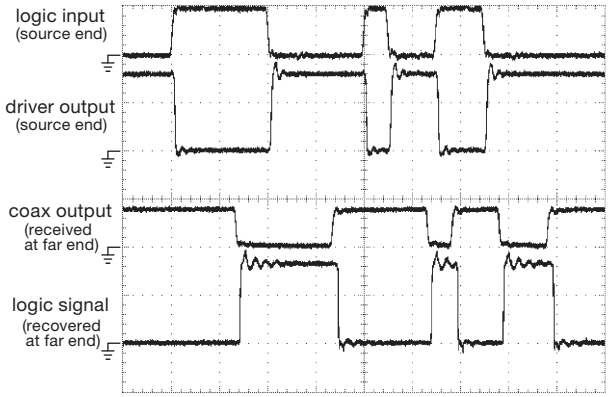

Figure 12.111. Waveforms from the circuit of Figure 12.110 ('ACT04 receiver), with the same 20 ns clocked pulse pattern as in Figures 12.104–12.109, and with a 10 m length of coax. Horizontal: 40 ns/div; vertical: 3 V/div.

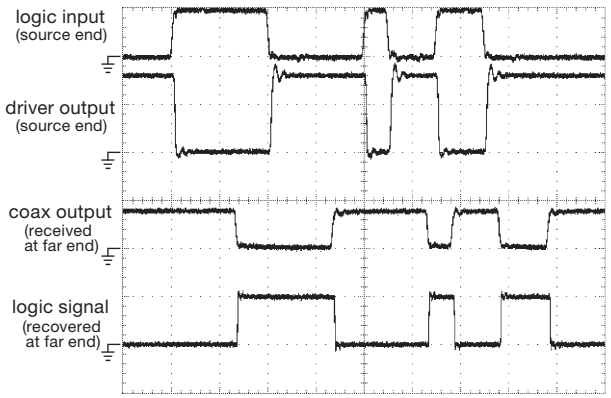

Figure 12.112. Same as Figure 12.111, but with 'LVC1G04 receiver.

The method of choice, for RF and video Because double-ended termination results in a received signal that is half the amplitude of the unloaded driver, it is not well suited for *digital logic* applications like this. It's better to use series termination (next subsection). Lest we leave the wrong impression, however, we note that the use of double-ended termination is *universal* in the RF and video worlds: all such signal sources are built with an output impedance equal to that of the cable impedance (50 Ω for RF, 75 Ω for video), and the far end of every cable is terminated in that same resistance. And the factor-of-two amplitude reduction is addressed by arranging the open-circuit amplitude of every signal source to be exactly twice what is ultimately needed at the (properly terminated) far end. You see this in signal and function generators – the output amplitude, as measured with a 'scope probe, is double what you've set it for (because it assumes you've attached a 50 Ω load resistor, or a cable with 50 Ω far-end termination). And in the video world you'll find plenty of "video buffer amplifiers" that provide a gain of exactly ×2, to compensate for the corresponding loss when driving a terminated cable. Figure 12.113 shows an example from LTC: their LT6553 is a triple video buffer (three independent amplifiers, to handle color analog video), with an internally set gain of 2, intended for driving 75 Ω video cable, as shown. It has impressive bandwidth (650 MHz) and slew rate (2500 V/μs), and can drive a ±3.5 V swing into the 150 Ω load it sees in this circuit, when powered from ±5 V supplies. Their companion LT6554 has the same specs, but with unity gain.

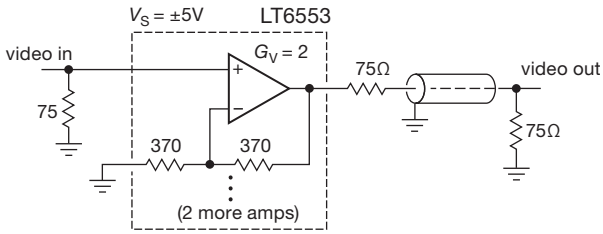

Figure 12.113. A video buffer amplifier drives a 75 Ω load through 75 Ω coax. The voltage gain of G_V=2 compensates for the ×2 signal attenuation caused by the series resistor at the amplifier's output.

B. The right way – III: Series termination

To summarize the previous methods: end termination with direct drive requires lots of drive current, but it delivers the full driver signal at the far end. Double-ended termination reduces the drive current, but attenuates the driver signal amplitude by a factor of 2.

There's a third way, which captures the best of each: use a series resistor at the source end and *no* termination at the far end. This is sometimes called "series termination" or "back termination" (Figure 12.114). It exploits a property of open-ended transmission lines, namely that there is a

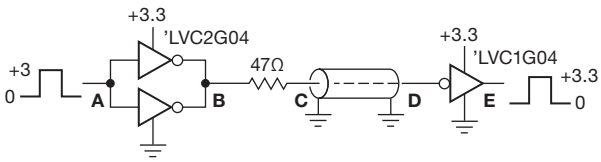

Figure 12.114. Series termination presents a load of twice the cable's characteristic impedance (therefore 100 Ω), while delivering the full driver amplitude at the far end. We recommend this method for logic outputs on instrument panels.

full-amplitude reflection from the far end, of the same sign as the incident signal, thus producing an output amplitude of twice the incident amplitude. But, because the incident amplitude is half the driver's output (owing to the voltage divider formed by the series resistor and the cable's input impedance), the net result is an output just equal to that of the driver. Voilà – full output swing, without the need to drive the cable's low impedance. And we don't get the ugly situation of signals bouncing back and forth (as in Figures 12.105 and 12.106), because the series resistor acts as a proper end termination for the backward-going signals. Instead you get one reflection off the far end, which gets swallowed at the source end, with the production of some steppy waveforms at the near end (Figure 12.115).

Series termination is the method of choice for piping logic signals through cable. This technique has the pleasant properties of (a) presenting a (less severe) load impedance equal to twice the cable's characteristic impedance, and (b) requiring *no* continuing drive current after a logic-level change has propagated down and back. To expand on this latter point: immediately after a step change, current flows from the driving gate, $I = V_{CC}/2Z_0$, or 50 mA for 5-volt logic driving a 50 Ω cable; but this ceases after a round-trip propagation delay. We have +50 mA for low-to-high steps, and −50 mA for high-to-low steps. Often we'll estimate the R_{OUT} of the driving gate and reduce the value of the added source resistor accordingly. But if the total source resistance is too low (less than 50 Ω), the returning echo will overshoot V_{CC} (for low-to-high steps) or undershoot ground (for high-to-low steps) for a time equal to a propagation delay. If the overshoot is high enough (i.e., more than 10% for a 5 V supply) it will cause current to flow in the gate's output clamp diode.

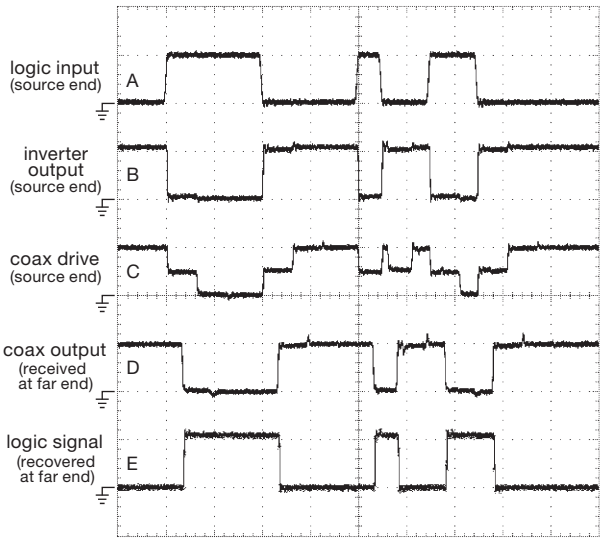

Figure 12.115. Waveforms from the circuit of Figure 12.114, with the same 20 ns clocked pulse pattern as in Figures 12.104–12.112, and with an 8-foot (2.4 m) length of coax. Horizontal: 40 ns/div; vertical: 3 V/div.

The logic family known as "AUC Little Logic" (one of TI's low-voltage logic families, one or two gates in 5-lead or 6-lead SMT packages) has an unusual output structure that is well suited for driving PCB traces or lengths of coax. Its output impedance is a reasonable approximation to a series termination, so you can drive a 50 Ω transmission line directly from the logic gate output, without any series resistor. The family is optimized for a +1.8 V supply voltage. As described in the Application Report[66] the output stage consists of three paralleled inverters, such that the driving impedance changes during a logic transition: it starts low (for high drive current), then becomes an approximate match to the transmission line, suppressing ringing or reflections. Also, there is no clamp diode from the output to the positive supply, so the output is "3.6 V tolerant" and is not damaged by back reflections from the open-ended transmission line.

Although they suggest cable or trace lengths only to 15 cm or so, we found that these devices work well for substantially longer runs of cable. Figure 12.116 shows the signals when driving a 30 cm length of 50 Ω cable (RG-316 "skinny" coax) with 100 MHz NRZ[67] data (twice the rate used in Figures 12.104–12.112 and 12.115). How far can you go with this driver, and at these speeds? Figure 12.117 shows the logic input pattern, and the (open-ended) coax output (points "A" and "D" in Figure 12.114), when driving lengths of 50 Ω coax[68] up to 5 m long. Looks pretty good, to us!

[66] *Application of the Texas Instruments AUC Sub-1-V Little Logic Devices*, SCEA027A (September 2002).

[67] "Non-return to zero," a fancy name for just sending each bit value, as a logic level, for one clock period.

[68] RG-141 for 500 cm, RG-316 for the rest.

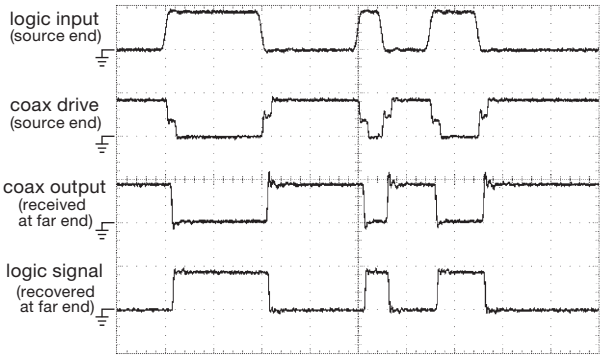

Figure 12.116. Waveforms from a 74AUC1G04 inverter operating at 1.8 V and directly driving a 30 cm length of coax, with the same pulse pattern as earlier, but at a 10 ns clock rate. The four traces (top to bottom) correspond to points A, C, D, and E in Figure 12.114, but with no resistor. The approximate half-step at the coax input confirms a drive impedance close to 50 Ω. Horizontal: 20 ns/div; vertical: 2 V/div.

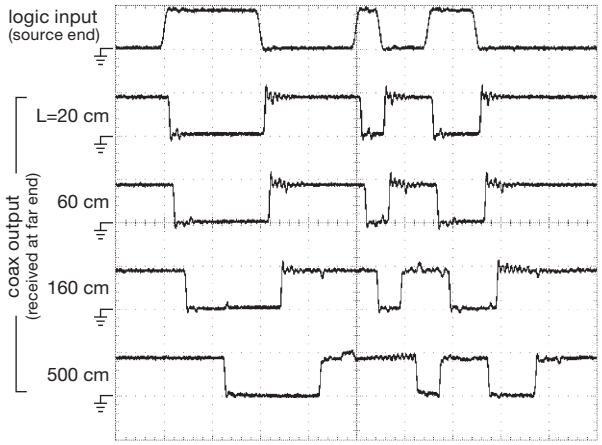

Figure 12.117. Same setup as Figure 12.116, showing received waveform at far end of coax cables with the indicated lengths. Horizontal: 20 ns/div; vertical: 2 V/div.

C. Driver pre-emphasis and receiver equalization

The measured waveforms above look pretty good. But we're not really pushing the limits, at these data rates of 50–100 Mbps NRZ. Things get pretty messy when you try to send upward of a few hundred megabits per second, because the coax cable itself becomes lossy and attenuates higher frequencies (see Appendix H). For example, the popular RG-58A (used for everybody's BNC patch cords) attenuates about 10 dB per 30 m at 500 MHz. You do better, of course, with cables of lower loss (e.g., RG-8 attenuates about 5 dB per 30 m). But these are bulkier – RG-8 cable is 10 mm in diameter, double that of RG-58.

Here's what you can do instead: at either the driver end or the receiver end (or both) you can compensate for the cable's loss (and phase shift) at higher frequencies. This is called *pre-emphasis* and *equalization*, respectively. Figures 12.130–12.132 later in the chapter show how it goes, in the case of a differential twisted-pair cable or printed-circuit differential stripline. For a (single-ended) coax line you could use a chip set like NSC's DS15BA101 and DS15EA101; the latter is a receiver with adaptive equalizer, which can apply up to 35 dB of boost at 750 MHz.[69] You can think of this trick as analogous to "turning up the treble," in an audio system, though here we need to worry about phase as well. It works quite well, owing to the very good signal-to-noise ratio of these large signal swings running through shielded (or twisted-pair) transmission lines. We'll see this next, in connection with LVDS signaling on differential pairs.

12.10.3 Differential-pair cable

There's another way to convey digital signals on cables, namely to use differential signaling, most often by way of a twisted-pair cable.[70] A common example of the latter is "Cat-5" (or Cat-6) Ethernet cable, which is an unshielded cable containing four independent twisted pairs, with characteristic impedances of 100 Ω. Some advantages of differential signaling are the suppression of common-mode interference and ground noise, the ability to use smaller signal swings (hence smaller drive currents),[71] and the large reduction of both radiated noise and signal-rate ground-current fluctuations (from the balanced differential swings, as well as the smaller amplitudes). Two popular differential signaling standards are RS-422 and LVDS. They differ in several ways. Roughly speaking, RS-422 is used for

[69] This chip, like other adaptive cable equalizers that are widely used in digital video systems (conforming to 75 Ω coaxial video serial digital interface standards with names like SMTE 259M, 292M, 344M, or 424M), is meant to be ac-coupled, and works only down to some minimum frequency of order 150 Mbps. See National Semiconductor's App Note AN-1909.

[70] Which may be shielded twisted pair ("STP") or unshielded twisted pair ("UTP"). It need not be twisted, however: you'll see adjacent wire pairs on a flat ribbon cable used for differential signaling, as well as parallel trace pairs ("microstrip," "stripline") on a printed circuit board.

[71] Because the signal is differential, it does not require accurate alignment of thresholds (as, for example, with the fussy single-ended 10K/100K ECL logic, whose ∼0.8 V swing mandates careful temperature compensation); the differential signal pair need stay only in the common-mode range of the receiver, e.g., 0 V to +2.4 V for LVDS, with its ∼0.35 V differential swing.

data rates up to a few megabits per second, and cable runs up to a kilometer. It uses differential *voltage* drive, and is common in industrial control applications. By contrast, LVDS is used for data rates to several gigabits per second, over distances to several meters. It uses differential *current* drive, and is common in short-range high-rate applications such as backplanes (e.g., PCIe[72]) and transmission of serial data (e.g., SATA, Firewire). Both are point-to-point connections, but provide multidrop variants (RS-422 → RS-485; LVDS → M-LVDS).

A. RS-422 and RS-485

These comprise a popular industrial data bus, improving on the venerable RS-232 data link standards (see discussion at §§12.10.4 and §14.7.8), and substantially extending the latter's capabilities, see Figure 12.134. The RS-422 and RS-485 standards[73] specify signal properties used in a differential voltage-drive arrangement like that shown in Figure 12.118. The differential outputs typically swing most of the way between ground and the +5 V rail, although the specification permits differential-output swings as small as ±2 V or as large as ±10 V. The receiver must respond to differential inputs as small as ±0.2 V, over a common-mode range of −7 V to +7 V (−7 V to +12 V for RS-485). The characteristic impedance of differential twisted pair cables is typically 100–120 Ω, so you've got to terminate the far end with that value resistor. You often see both ends terminated (as in Figure 12.118), which is not necessary (or desirable) with unidirectional RS-422, but required with the bidirectional (or multipoint) variant (RS-485).

Figure 12.119 shows measured signals traversing one of the four twisted pairs in a 140 m length of Cat-6 Ethernet cable, with a, by now, familiar bit pattern clocked at 10 MHz (i.e., an NRZ rate of 10 Mbps). Although this is ten times faster than allowed by the RS-422 specification (Figure 12.134), the differential nature of the received signal allows clean logic recovery nonetheless; note the propagation delay of ∼700 ns (mostly caused by the 4.7 ns/m signal velocity in the cable).

Differential signaling provides impressive immunity to common-mode interference. The latter may arise from nearby signal-carrying wires or from radiated signals (from intentional emitters like radio/TV, wireless networks, etc.);

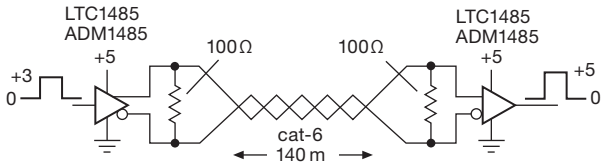

Figure 12.118. Differential voltage signaling with RS-422/485 over a length of unshielded twisted pair, using transceiver chips with 3-state outputs (each chip has a separately enabled Tx and Rx, sharing a common differential pair).

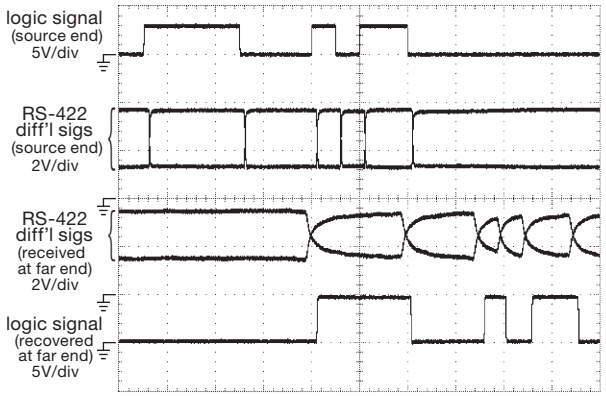

Figure 12.119. Waveforms from the circuit of Figure 12.118, with a 100 ns clocked pulse pattern, and a 140 m length of Cat-6 network cable. Horizontal: 200 ns/div.

it's also common to see a difference in ground potentials between equipment that is plugged into different power outlets. We have seen as much as a volt or two of 60 Hz ac between instruments in the same room!

We hooked up the circuit in Figure 12.120 to illustrate common-mode rejection in RS-422 signaling. The pseudo-random noise generator adds a ∼15 Vpp noise voltage into the (floating) ground of the signal source, as seen from the destination end. The results are shown in Figure 12.121: both the 5 V logic signal and the RS-485 differential pair signals are hopelessly noised up, reaching peak levels of −7 V and +12 V (the specified receiver common-mode range) as seen from the destination end. But the magic of common-mode rejection recovers the original logic intact. Note that the noise is very much "in-band," with wiggles on the time scale of the digital data. Although the figure captures a single-shot data burst, the result is robust – we ran it many times, with never an error.

Slew rate limiting If you don't need the speed, it's wise to choose driver ICs of lower slew rate because you get less signal coupling from signals on adjacent pairs. RS-422 and

[72] Shorthand for "PCI Express," itself shorthand for "Peripheral Component Interconnect Express."

[73] Officially known as ANSI TIA/EIA-422 and TIA/EIA-485; however, most engineers continue to use the original designations "RS-422" and "RS-485," or, more loosely, they'll just say "422" or "485." In context it's clear what they mean.

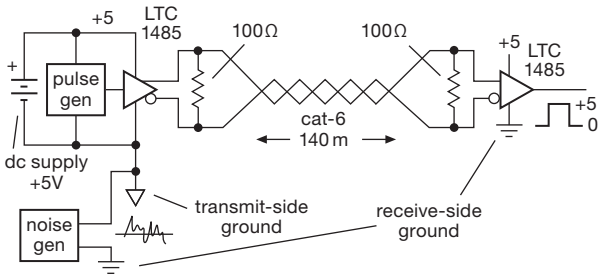

Figure 12.120. Torture test for RS-422 common-mode noise rejection. We floated the driver-side circuitry, and drove its "ground" with band-limited noise of ∼15 Vpp amplitude, relative to far-side ground.

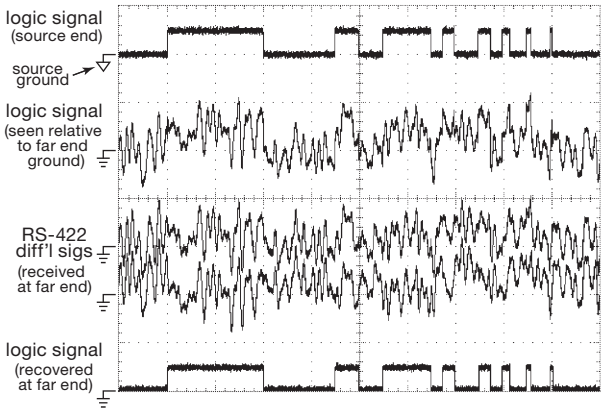

Figure 12.121. Waveforms from the circuit of Figure 12.120. As seen from the far end, the input logic signal is buried in the noise, extending from −7 V to +12 V. The individual differential signals are likewise a mess, but the recovered logic output is error free. The signal pattern includes pulses with widths from 200 μs down to 5 μs. Horizontal: 100 μs/div; vertical: 10 V/div.

RS-485 drivers are available with a selection of slew rates, for example the Maxim MAX3293-95 series, with specified rates of 250 kbps, 2.5 Mbps, or 20 Mbps. Other examples include the MAX481–489- and 1481-1487-series, the LTC2856–2858-series, and the 65ALS176 versus the 75ALS176B.

RS-422 versus RS-485 RS-485 is basically RS-422 with some additions that make it possible to have several drivers share a single signaling pair: this requires that the drivers have an ENABLE input, so they can be put into a high-impedance (noninterfering) state, analogous to the use of three-state drivers on a shared (single-ended) data bus line.[74] RS-485 interface chips (e.g., the classic 75ALS176, or LTC1485) usually combine a transmit–receive pair, sharing the same differential signal lines ("half duplex"),[75] with complementary ENABLE pins (called DE and RE′); such a *transceiver* chip looks like Figure 12.122.

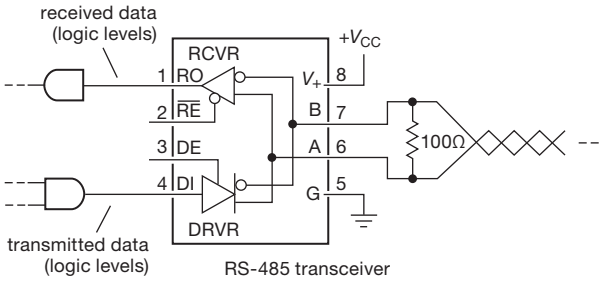

Figure 12.122. An RS-485 *transceiver* combines driver and receiver in one package, sharing a single twisted pair. Separate ENABLE pins (usually tied together) permit operation in either direction. The pin numbering shown is an industry standard.

Along with the ENABLE feature that is necessary for multiple drivers (called "multipoint"), the RS-485 specification beefs up some other RS-422 specifications: (a) it expands the receiver common-mode input range (−7 V to +12 V), thereby allowing symmetrical common-mode swings (up to ±7 V) around the traditional power-supply and signaling levels of 0 V and +5 V; (b) it lowers the minimum load resistance (to 54 Ω), which is necessary because a driver at either end requires end-termination at the other end (thus a 100–120 Ω at *both* ends). The RS-485 specification also (c) expands the allowable common-mode range that can be applied to a driver *output* (in the disabled state), from a diode drop beyond the rails (RS-422) to the full −7 V to +12 V range of an RS-485 input. This is necessary, of course, because the driver and receiver are both connected permanently to the signal lines in a typical bidirectional (or multipoint) arrangement; if the driver's output clamped to ground or the positive rail, it would defeat the receiver's wider common-mode range.

Most RS-485 transceivers satisfy also the narrower RS-422 specifications, so you might as well just use one of the hundreds of available RS-485 interface chips (from manufacturers like Analog Devices, Intersil, LTC, Maxim, or TI), even if you are sending data in only one direction; simply enable what you are using at each end. Some common choices are the 'ALS176 (and its many imitators –

[74] Good references: Analog Devices AN-960 and AN-727, National Semiconductor AN-759, Maxim AN-3776, and TI SLLA112.

[75] An interface chip that brings RS-485 driver and receiver pairs out separately is called "full duplex."

65LBC176, 65HVD1176, etc.), the 75176 and 75ALS180 classics, the LTC1480/5, and the ADM1485.

Isolated RS-422/485. Scenario: a "factory floor," with a host of automated machine tools and with products moving along on conveyor belts (think of a scene from the ever-informative "How It's Made" series on cable channels like Discovery Channel or Science Channel). Sensors send digital information through cables that snake along on overhead cable trays, and wind up at a central computer control; the same cables, or different ones, carry commands back to actuators. This centrally controlled network choreographs factory activities, ultimately directed toward the practical (and profitable) goal of… making stuff. It's common to see these signals piped around as differential RS-485 or a variant like the Process Field Bus (PROFIBUS®) that uses a similar "physical-layer" signal.[76]

These activities may be spread out over distances of hundreds of feet, and may go between buildings. Let's face it, with all that heavy machinery thumping along, you're likely to see common-mode transients that exceed even the "generous" specifications of RS-485.[77] The solution here is to use *isolated* interface chips, in which the RS-485 signals have their own independent ground, galvanically isolated from the ground of the logic signals. This requires a second dc power source, of course, floating with respect to the logic signal ground.

There are many isolated RS-485 interface chips to choose from. The clever designers use various tricks to make fast digital signals cross a gap that can withstand a kilovolt or so. For example, the LTC1535 (or MXL1535) "isolated RS485 Transceiver" uses small capacitors to couple the (modulated) digital signals (Figure 12.123). It also helpfully includes a high-frequency (∼420 kHz) oscillator, whose output you can use with a transformer to isolate and rectify to make the isolated dc for the RS-485 signal side. The ISO15/35 from TI is a less-expensive alternative, which omits the oscillator; you've got to provide isolated dc from scratch (either from an isolated dc–dc converter, or an ac-line-powered dc supply).[78]

Another approach is to couple with small (chip-scale) transformers. This technique is used in the ADM2485 series of isolated transceiver from Analog Devices, which also provides an oscillator output that can be used for generating the isolated dc (like the LTC1535). Or, you can add the popular MAX845 or MAX253 chips, which generate a pair of complementary square wave outputs at ∼0.75 MHz, suitable for directly driving a small isolation power transformer, good to a half-watt or so of isolated dc. The electronics industry makes it easy to use this part: major transformer companies even offer "MAX845 transformers."

An interesting technique is used in the IL3485 from NVE Corp, namely the "giant magnetoresistive effect" (GMR, widely used in hard-disk drives to sense the magnetically stored bits on the rotating platters). Instead of conventional transformer coupling, where a secondary winding senses flux *changes*, the NVE part uses GMR to sense the field directly.

Finally, isolated transceivers like the MAX1480/90 use optocouplers for the digital signals. And, to make your life really easy, they include an internal transformer to generate isolated dc power. These hybrid ICs include also the diodes and capacitor to complete the power circuit, so all you need to provide is a +5 V logic-side supply. That's the good news; the bad news is that these puppies will set you back twenty bucks (in small quantities), compared with $5–$8 for the capacitor- or transformer-coupled chips.

Ethernet PHY When thinking about moving digital data across isolation barriers, don't forget about the optical couplers and fibers earlier in this chapter, and also about isolation via pulse transformers, as used, for example, in local-area networks such as Ethernet. Figure 12.124 shows the physical layer ("PHY") of an Ethernet link, with pulse isolation transformers (which everyone just calls the "magnetics."). They provide excellent rejection of common-mode interference (and of course of differences in potential at the ends), using both a transformer and a common-mode

[76] PROFIBUS transceivers have higher maximum data rates, typically 30 Mbps or 40 Mbps, and generally meet the (less-rigorous) specifications of RS-422 and RS-485; check out parts like the 65ALS1176, ADM1486, ISO1176, or ISL4486. Some other industrial buses that use RS-485 signaling include BITBUS, Data Highway (DH-485), INTERBUS-S, Measurement Bus (DIN 66348), Optomux, P-NET, and Series 90 (SNP).

[77] As eloquently expressed in the MAX1480 datasheet, "The RS-422/485 standard is specified for cable lengths up to 4000 feet. When approaching or exceeding the specified maximum cable length, a ground-potential difference of several tens of volts can easily develop. This difference can be either DC, AC, at powerline frequency, or any imaginable noise or impulse waveform. It is typically very low impedance so that if a connection between the two grounds is attempted, very large currents may flow. These currents are by their nature unstable and unpredictable. In addition, they may cause noise to be injected into sensitive instrumentation and, in severe cases, might actually cause physical damage to such equipment." Amen.

[78] You sometimes see industrial RS-485 hookups with bused dc (+24 V or +48 V "telecom dc") bundled in the same cable as the RS-485 signal pair(s); hang an isolated dc–dc converter onto that, at each transceiver node, to power the RS-485 signal side of a transceiver like the ISO15/35.

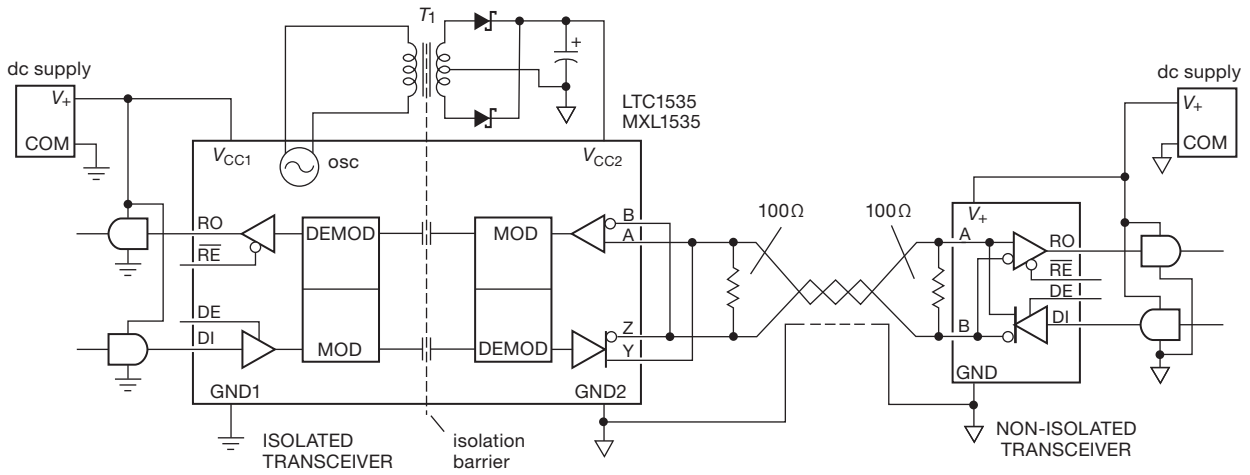

Figure 12.123. An "isolated" RS-485 transceiver galvanically separates the circuitry connected to the cable pair from the logic-level circuitry. The far-end ground ("GND2," triangular symbol) is brought back to the isolated transceiver, but is never connected to the local ground ("GND1," normal symbol). Such an arrangement prevents ground loops, and can accommodate common-mode and ground offset potentials of hundreds of volts. Additional isolated transceivers can be connected ("multidrop") along the span, referenced to the same GND2, each with its own isolated RS-485-side power supply; such midspan "stubs" must be kept short in length, and without 100 Ω terminating resistors.

choke, as shown.[79] If you've got a microprocessor in your system already, why invent your own isolation hardware, when Ethernet works just fine? And, if you need to go greater distances, you can use a media converter to carry Ethernet signals on fiber. Check out the offerings from manufacturers like Allied Telesis.

B. LVDS

In contrast to RS-422, which is intended for modest data rates (10 Mbps and below) over relatively long cables (to 1 km), the LVDS (low-voltage differential signaling, also known as RS-644) standard is aimed at much higher data rates (to 1 Gbps and above) over shorter cables (to ∼10 m) or even shorter circuit-board trace runs. Rather than driving the wire pair with crisscrossing *voltages* of a few volts amplitude, LVDS switches *currents*: an LVDS driver sinks and sources 3.5 mA (nominal) into a wire pair that is terminated at the far end in its characteristic impedance (usually 100 Ω). That produces a differential voltage at the far end of ±350 mV. The driver is obliged to maintain a common-mode voltage, nominally +1.2 V; so the receiver sees crisscrossing voltages of approximately +1.0 V and +1.4 V. The relatively low common-mode voltage was chosen deliberately to accommodate both driver and receiver chips operating at low supply voltages. That's important because digital chips are moving steadily to lower voltages, and the

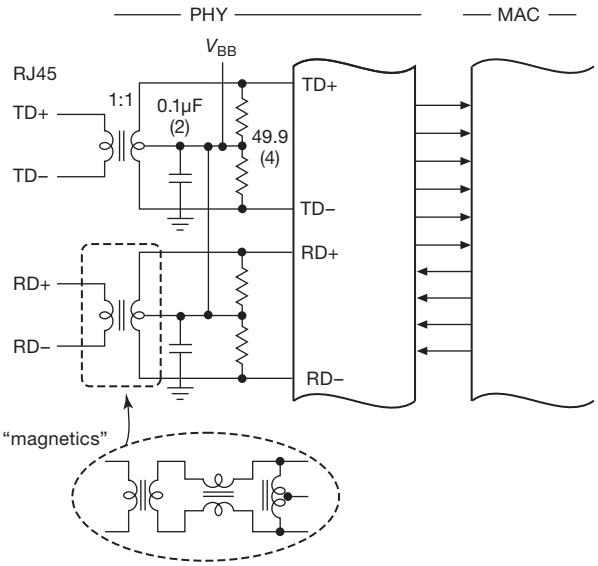

Figure 12.124. Ethernet uses transformer coupling and common-mode chokes for robust isolation of its differential signaling. Closest to the wiring is the physical layer ("PHY"), followed by the media-access control layer ("MAC," as in "MAC address").

[79] You get additional robustness by adding surge-protecting components, notably the Transient Blocking Units (TBU™) from Bourns. These are little 2-terminal elements that go in series with the signal leads from the transformer: they act like a low resistance, up to a critical current, at which point they go into a high resistance state.

LVDS interface is often incorporated right into a complex chip that wants to run at low voltages.

We hooked up the circuit in Figure 12.125 (analogous to Figures 12.120 and 12.121) to illustrate LVDS's rejection of common-mode interference. We injected a trapezoidal waveform at ∼30 MHz into unused pairs of a network cable, and cranked up the amplitude until the received signal reached the specified common-mode receiver limits of 0 V and +2.4 V. The results are shown in Figure 12.126: You can see the two received signal voltages weaving busily past each other, at the command of the input signal (delayed by that speed-of-light nuisance, of course), cleanly recovered by the LVDS receiver.

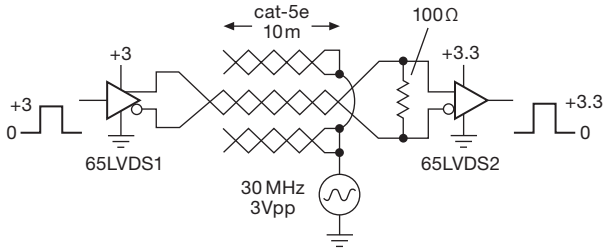

Figure 12.125. Torture test for LVDS common-mode interference rejection. We applied a ∼30 MHz signal to two unused pairs at the far end of a 10 m length of Cat-5e network cable, while transmitting a pulse sequence (clocked at 20 ns) through a different pair.

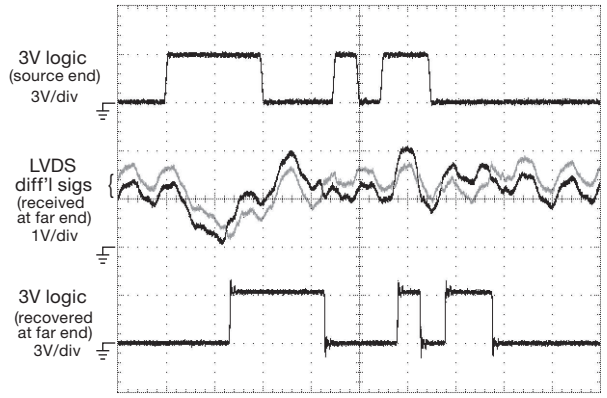

Figure 12.126. Waveforms from the circuit of Figure 12.125. The injected signal adds up to ±1 V of common-mode interference to the ∼400 mV differential 50 Mbps received signal. Horizontal: 40 ns/div.

To illustrate the problem of conveying digital signals between instruments whose grounds differ by a small amount of 60 Hz ac, we rigged up the circuit of Figure 12.127. We "floated" the source device's ground, analogous to the arrangement earlier with RS-422 (Figure 12.120), then drove it with a 2 Vpp sinewave. The results are shown in Figure 12.128, where we've clocked down the data rate so you can see the powerline frequency. Note that LVDS permits only 2 Vpp of added common-mode signal, compared with 14 Vpp for RS-422/485; but with LVDS you get lots of speed, and compatibility with low-voltage logic.[80]

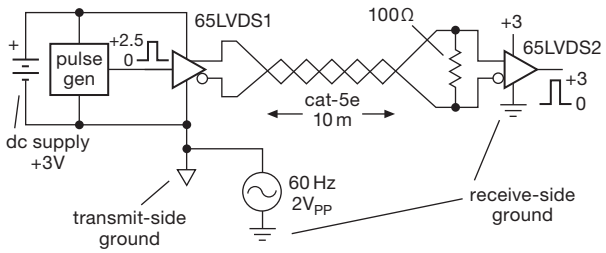

Figure 12.127. Powerline common-mode test setup, using a floating LVDS signal source (complete with pulse pattern generator and power supply).

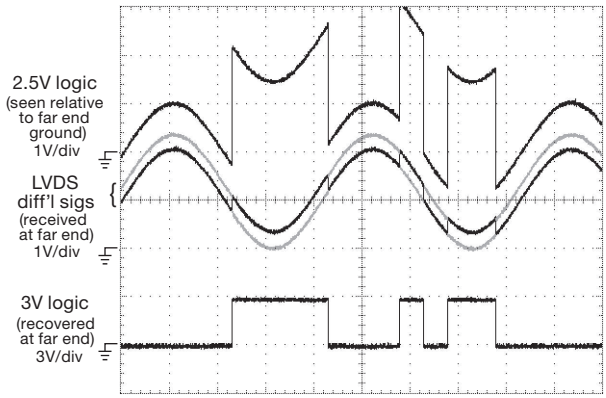

Figure 12.128. Waveforms from the circuit of Figure 12.127. The ∼400 mV differential signal is superposed on the added 2 Vpp 60 Hz sinewave, reaching the specified receiver common-mode limits of 0 V and +2.4 V. Horizontal: 4 ms/div.

The LVDS protocol is widely used in serializer-deserializer links (see also §12.8.4, §12.10.5, and §14.7), in which a fast serial link connects a pair of separated parallel registers. At each end you think you're talking to a parallel port (and you are!), but in between the data travels as serial bits. A common data width is 10 bits, which lets you send a byte plus two extra bits to signify whatever you want (a new byte, or the beginning of a new "frame"

[80] You can get LVDS receivers with extended common-mode capability, for example the 65LVDS34, which specifies operating common-mode range of −4 V to +5 V (they do it with a resistive input divider, so R_{in}=250 kΩ).

of bytes). Figure 12.129 shows how it goes, in this case with a SERDES pair with a relatively relaxed speed specification: the transmit clock, which times the entry of 10-bit data symbols, can be in the range of 16–40 MHz. The serialized data bits that go over on the LVDS link are at ten times that rate, i.e., up to 400 Mbps.[81] You need only one LVDS differential pair, because the deserializer performs "clock recovery" from the transmitted bitstream. In the figure we've omitted some additional details having to do with modes and synchronization.

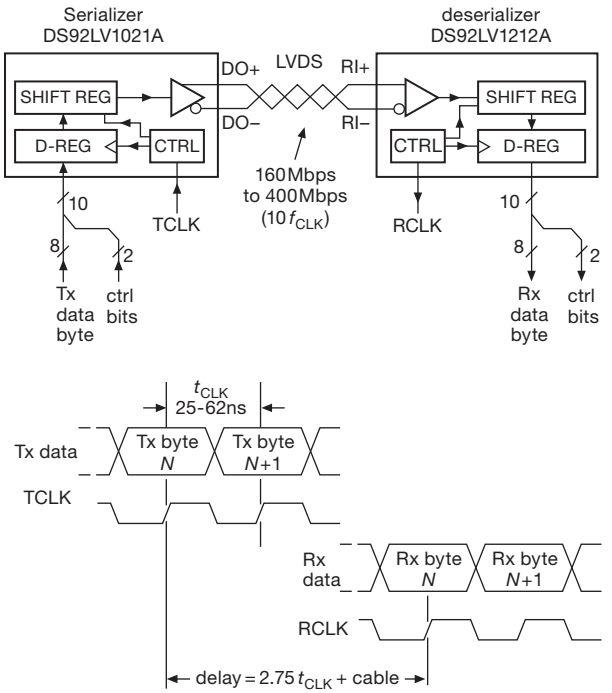

Figure 12.129. A serializer-deserializer pair lets you clock parallel data into a register at the transmit end; it appears, magically, as parallel data at the receive end. There are a few clock cycles of delay, but nothing to write home about.

LVDS signal conditioning As we remarked earlier (§12.10.2C), the use of transmit pre-emphasis and receive equalization can compensate for the frequency-dependent loss of the connecting cable. These *signal-conditioning* techniques reduced greatly the effects of data-dependent jitter, which is also known as *intersymbol interference*, or ISI.

This ISI effect is worth a few words of explanation: because of the lowpass filtering effect of the losses in cables or printed-circuit traces, particularly when operating at very high bitrates (gigabits per second), the initial received signal voltage at the beginning of each bit cell depends on the previous bit (or bits), and so the time to cross the threshold will vary somewhat, depending on the previous bit(s). Intersymbol interference plagues high-speed communication in all forms. You can see this effect in the 1.5 Gbps recovered LVDS waveforms in Figure 12.132, in which the 'scope traces, triggered by the clean clock signal, show the scatter of signal levels on the data line carrying a pseudo-random data pattern. This is called an "eye diagram;" it is a standard tool for visualizing noise and jitter in assessing signal quality in a clocked data stream.

The use of pre-emphasis and equalization can significantly extend bitrates and transmission distances, as we remarked earlier in §12.10.2C in connection with coaxial cables. Figure 12.130 shows the scheme, and Figure 12.131 shows how pre-emphasis boosts the voltage levels following each transition; and when combined with receiver equalization (high-frequency boost), the result is a clean replica of the original input signal. Such signal conditioning is used in various driver and receiver ICs, for example the DS25BR100/200/400 and DS25CP102 series from NSC; see their App Note AN-1957 for details and illustrative waveforms.

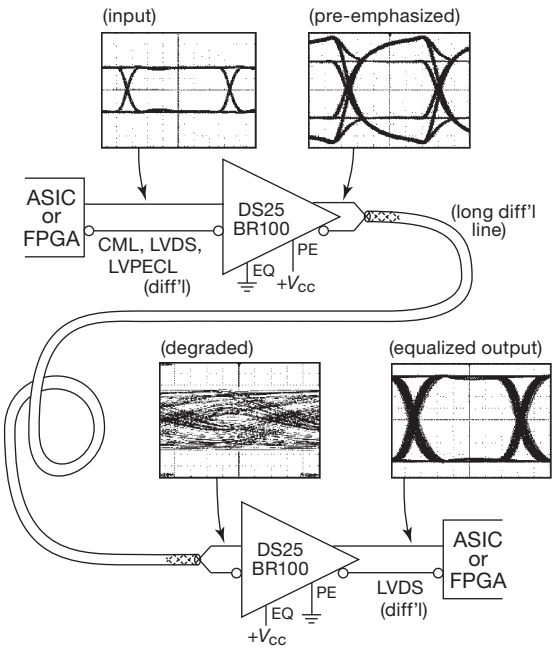

Figure 12.130. Driver pre-emphasis, receiver equalization, or both can compensate for cable losses when transmitting at high data rates over lossy media.

[81] The serial bitrate is actually 12 times the clock rate, because the serializer adds two clocking bits.

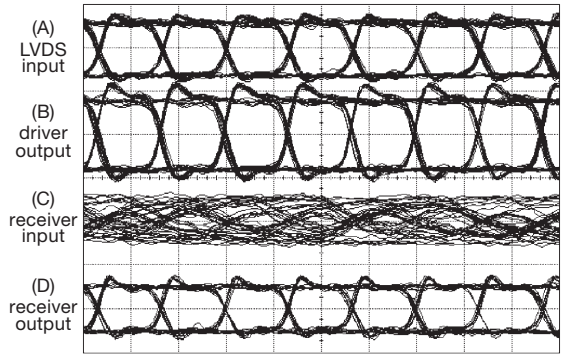

Figure 12.131. Waveforms from a 1.5 Gbps pseudorandom LVDS signal traversing a 2.5 m length of differential stripline on a printed circuit board. A. Input to DS25BR120 driver; B. Driver output, with pre-emphasis; C. Input to DS25BR110 receiver; D. Receiver output, with pre-emphasis and equalization. Vertical: 500 mV/div; horizontal: 500 ps/div.

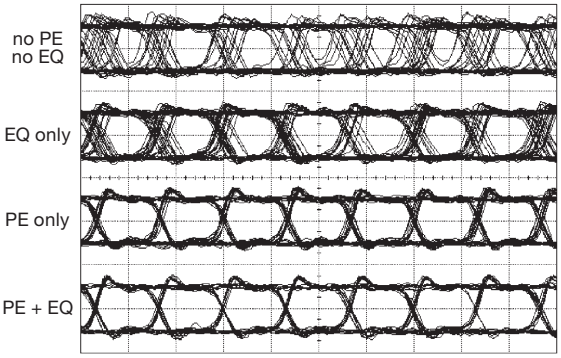

Figure 12.132. Receiver output waveforms ("eye diagrams") from the same setup as Figure 12.131, showing the effect of varying amounts of driver pre-emphasis and receiver equalization. The bottom trace represents PE=+9 dB and EQ=+8 dB at 1.5 GHz. Vertical: 500 mV/div; horizontal: 500 ps/div.

12.10.4 RS-232

This is a signaling format that goes back to the 1960s, intended originally for low-speed (<19.2 kbps) serial links between alphanumeric terminals (e.g., the legendary DEC VT-100) and computers. RS-232 has been revised several times and is now known officially as EIA232; although RS-232 ports are still seen on some computers and instruments, the standard is considered old-fashioned, and arguably headed for the trashcan. It's still with us, however (and perhaps always will be[82]). Although not part of the standard, the data is usually sent as asynchronous 8-bit serial data bytes, with a synchronizing START bit, and one or two STOP bits (these extra bits allow the receiver to resynchronize after each byte; see §14.7.8). The usual bit rates (also not specified in the RS-232 standard itself) are power-of-2 multiples of 300 bps (thus 300, 600, 1200, 2400, 4800, 9600, and 19,200 bps, supplemented with an interleaved set consisting of 14.4 kbps, 28.8 kbps, 57.6 kbps, and 115.2 kbps). Thus, for example, when you set up a serial port for "9600 8N1," you are sending groups of 8 bits, with one START and one STOP bit, and no parity, at 9600 bits/s. Note that the synchronizing bits (START and STOP) are included in the overall bitrate measure; for this example, then, you're sending a payload of 960 bytes/s.

What the standard *does* specify are the signaling voltages, load resistance and capacitance, and slew rates, along with the connector pinouts. We'll see RS-232 again, in the context of computer communications, in Chapter 14. But here we stick to the use of RS-232, at the physical level (hence the term "PHY"), as a way to drive cables with digital data. The RS-232 voltage levels are bipolarity, with legal driver output voltages of either -5 V to -15 V (logical 1, also known as "MARK"), or +5 V to +15 V (logical 0, also known as "SPACE"). RS-232 drivers and receivers are inverting, so MARK corresponds to logic HIGH at the input of a driver or at the output of a receiver (see Figure 12.135). As indicated in the figure, RS-232 is single ended; and, because of the relatively low signaling rates, the cable is unterminated. That's a terrible thing to do, normally, because rapidly changing signals (i.e., on a time scale shorter than the round-trip signal travel time) reflect from the open end, as we've seen. RS-232 finesses that problem by specifying a maximum slew rate (30 V/μs), which is slow enough for typical cable runs of fifteen meters (the original specified maximum, later replaced with a maximum load capacitance) or less. To be in compliance with the standards, the load capacitance must be 2500 pF or less;[83] and the load resistance is a nominal 5kΩ ($\pm$2kΩ). RS-232 drivers must withstand a continuous short to ground or to any dc voltage in the range $\pm$25 V.

[82] It's hard to completely dismiss RS-232, when so many test instruments are happily working well with it, as one of their several interface choices. The move toward USB is being stolen by Ethernet instead (or other replacements), leaving an RS-232 connector as the easy-to-implement port. Unlike more advanced serial interfaces like USB and Firewire, RS-232 does not require any "smarts" – there's no initialization or negotiation needed. Partly for this reason, most microcontrollers include an easy-to-use serial port controller. And for connection to contemporary laptop computers and the like, you can get USB-to-RS-232 adapter pods, for example from FTDI.

[83] Which corresponds to 50 m of Cat-5 twisted pair (50 pF/m), or 25 m of 50 Ω coax (100 pF/m).

So, RS-232 rumbles along, slowly. The good news is that it can ride along in a multiwire unshielded cable, which need not act as a well-behaved transmission line; the limited slew rate minimizes both crosstalk and reflections. The bad news is that the driver needs both positive and negative supply voltages of at least ±5 V. The original driver–receiver chips (1488/1489, and their CMOS follow-ons DS14C88/89 and MC145406) required such dual supplies (±9 V, nominal), which is asking a lot in a digital system (like a computer motherboard) that runs on positive supply voltages only. Maxim was the first to introduce RS-232 drivers with an on-chip charge pump (flying capacitor) voltage doubler and voltage inverter, to generate ±10 V from a single +5 V; they named it (naturally) the MAX232. There are now dozens of such chips (some with the capacitors included in the package, e.g., the MAX203 or LT1039), spanning a range of maximum speed, power consumption, supply voltage, number of drivers and receivers in one package, and so on. As an example, the MAX3232E is a dual transceiver (two drivers, two receivers) that runs from a single +3 V to +5.5 V supply, requires four external 0.1 μF capacitors, and is guaranteed to meet RS-232 specifications to 120 kbps; it comes in five different packages (SMT and DIP). The -E suffix designates enhanced robustness to electrostatic discharge, namely ±15 kV ESD-protected, specified with the Human Body Model (HBM, recall §12.1.5) of a 100 pF charged capacitor in series with a 1.5kΩ resistor.

The RS-232 interface specification includes a number of additional control signals, intended for hardware "flow control" when a terminal is connected to a computer (see §14.7.8); these have names like Data Terminal Ready (DTR), Data Set Ready (DSR), Request to Send (RTS), and Clear to Send (CTS). You can ignore all of this if you simply want to use an RS-232 interface chip pair to send low-speed digital data over a wire connection. In fact, these are often ignored even in computer serial ports: you can simply use the transmit data (TD) and receive data (RD) lines, plus ground, and do the flow control with software (more on this in Chapter 14). Note, however, the curious (and confusing) official naming of the actual signal lines TD and RD: you would expect that TD is the output from a driver, which should be connected to RD of the (distant) receiver. Not so! In RS-232 nomenclature, a device is either Data Communications Equipment (DCE) or Data Terminal Equipment (DTE); the signal sent by the latter is called TD *at both ends of the link!* (and likewise for RD). Go figure. In practice, most engineers ignore this confusion; they call an outgoing signal TD and an incoming signal RD, at whichever end it's happening.

Figure 12.133 shows received RS-232 signals and the recovered +3.3 V logic-level digital data. We used a pair of MAX3232E dual RS-232 transceivers, powered from +3 V, and driven by a random byte generator[84] emitting standard serial-formatted data at 115.2 kbaud. The 'scope capture, triggering on the transition from negative STOP bit to positive START bit, shows many such bytes superposed, so the data bits attain both values. The recovered logic signal is clean after 10 m of twisted pair (approximately its rated limit), but after 140 m there is significant spread in the recovered data's edge timings, caused by intersymbol interference (§12.10.3B) in the slow-transitioning RS-232 signal. Here, the slow transition time is comparable to the time for one bit (1 "UI," or unit interval), causing the time to cross the 0 V threshold to vary somewhat, according to the preceding bit values. You can see this effect in the long cable data in Figure 12.133, where the received cable signal falls short of its asymptotic voltage levels (approximately ±4.5 V) by as much as a volt.[85]

When sending digital signals on wires over substantial distances, or in electrically noisy environments, there

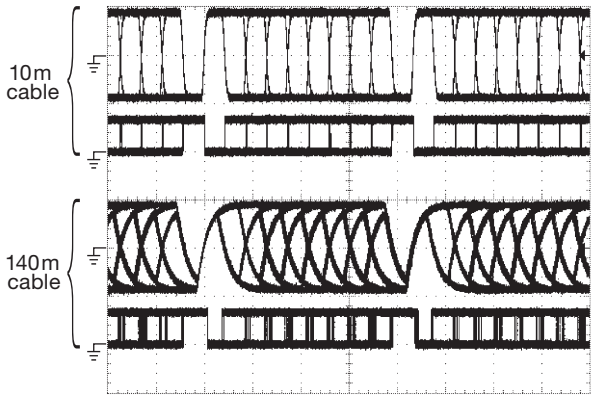

Figure 12.133. RS-232 signaling uses single-ended voltage drive, alternating between positive and negative voltage levels. These 'scope captures show received random bytes ("8N1") at 115 kbaud (top trace of each pair) after transmission through two lengths of Cat-5e/6 network cable; the drivers and receivers are inverting, as can be seen in the recovered logic signal (bottom trace of each pair). The marginal performance with the longer cable is consistent with Figure 12.134's limit of ∼10 kbaud for this cable length. Driver/receiver: MAX3232E at +3 V. Horizontal: 20 μs/div; vertical: 5 V/div.

[84] The one used to generate the ∼250 MB of random bytes on the *Numerical Recipes* CD-ROM.

[85] Users of RS-232 often use slower data rates, the most popular of which is 9600 baud, at which these effects are insignificant.

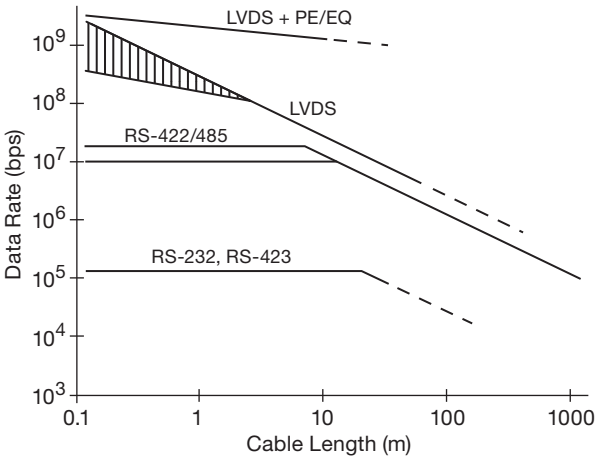

Figure 12.134. Approximate data rate limits versus cable length, for several signaling protocols. Your actual mileage may vary, depending on cable quality and interference environment. Note the speed improvement gained with pre-emphasis and equalization ("PE/EQ," see Figures 12.130–12.132).

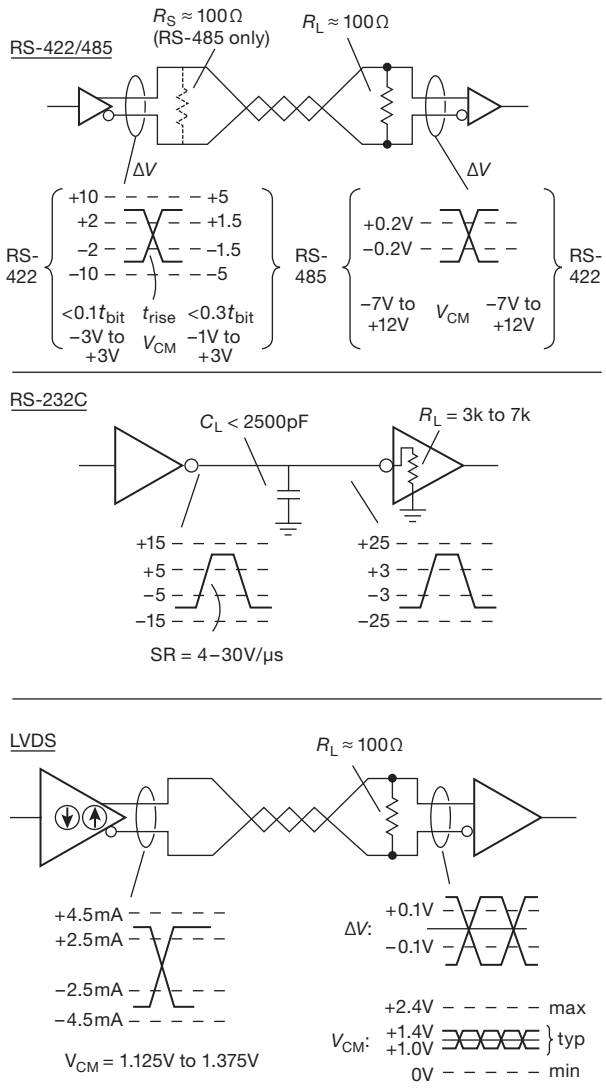

Figure 12.135. Allowable driver and receiver signal levels for LVDS, RS-422/485, and RS-232 signaling. Except for the bipolarity (and single-ended) RS-232, the output voltages from the drivers are of positive polarity only. LVDS outputs are *current-mode* drive (which are converted to a differential voltage signal at the cable's terminating resistance), at a common-mode voltage of +1.25 V; the others are *voltage-mode* outputs. All of the receivers respond to voltages, whether single ended (RS-232) or differential (LVDS, RS-422/485).

are always the problems of ground currents and of injected noise. Galvanic isolation is the best solution. Although RS-232 does not receive as much attention as RS-422/485, there is at least one good isolated driver, namely the ADM3251E from Analog Devices. It uses internal transformer coupling (with a modulator–demodulator pair) for the transmit and receive data, and an additional transformer (plus rectifier and regulator) to generate the isolated dc; the only external components are five 0.1 μF for the (internal) charge pumps.

One final comment: we've been talking mostly about the RS-232 *physical layer* (the actual voltages on the cable), with its awkward bipolarity signaling, as a simple tool for direct transmission of digital data. In the real world, RS-232 is ordinarily used in conjunction with asynchronous serial data sources, for example between a host computer's "serial COM port" and a device such as a modem or programming pod. In that role, you need to use driver and receiver interface chips (like the MAX3232) at each end, to translate between the logic-level signals and the RS-232 signals. Although we're less than enthusiastic about RS-232 signals, we believe that the use of the simple asynchronous serial data protocol *without conversion to RS-232 voltage levels* will continue to be useful. That's because it is the last of the uncomplicated serial interface standards – later serial protocols such as USB, Firewire, and SATA require substantial brainpower to negotiate and operate the link. Most microcontrollers include one or more serial ports (called UARTs, or COM ports), which are easy to use, and which can talk to any computer via a serial-to-USB converter such as the popular FTDI TTL-232R-3V3. This handy device plugs into a USB host at one end, and gives you a serial port with +3.3 V logic levels (for direct connection to a microcontroller, or whatever) at the other end.

12.10.5 Wrapup

In contemporary practice LVDS is quite popular, owing to its combination of high speed (to 3 Gbps and beyond), low emission of interference, low power, and compatibility across low-voltage logic families.[86] It's used widely for high-speed signal transport and distribution on circuit boards and backplanes, as well as through short runs ($\lesssim$10 m). You can get discrete drivers and receivers in small packages (e.g., 65LVDS1/2), and many complex ICs include LVDS driver/receivers for serialized data, for example a SERDES pair like the DS92LV1023/1224, or a complex FPGA like the Xilinx Spartan-3 or Altera Stratix series.

RS-422/485 is used for longer runs, often in industrial environments, where the data rates are lower (to 10 Mbps) but the common-mode noise is higher. And RS-232 survives, in spite of frequent predictions of its demise, for simple data links at low rates. For applications where the signal must be isolated from interference (both incoming and outgoing) you'll see shielded cables, either as shielded twisted pairs for differential signaling, or as coax cables for single-ended (or analog) signaling. It's also common to see galvanically isolated RS-485 interface chips, employing electrostatic (via capacitors), magnetic (via transformers, or GMR), or optical (via LEDs) techniques to bring the data across a several-kilovolt isolation barrier. Finally, fiber optics (§12.8) provides a totally noise-immune and galvanically isolated digital link, capable of very high data rates and long runs, in exchange for higher costs in driver–receiver components.

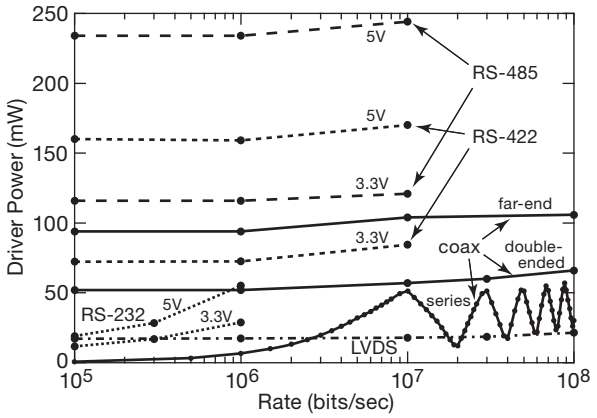

Figure 12.136. Measured driver power versus bitrate, for several single-ended (coax) and differential (twisted-pair) cable-driving configurations. All tests used a 10 m length of RG-58A coax or Cat-5e network cable, driven with alternating 1s and 0s (i.e., a square wave whose frequency is half the bit rate). The far-end termination resistance was 50 Ω for coax (except open circuit for "series" terminated), and 100 Ω for twisted pair (except 5 kΩ for RS-232) with another 100 Ω at the source end for RS-485. Note the effect of reflections from the mismatched end termination (5 kΩ) with RS-232, particularly evident here because of the single-frequency square-wave drive.

Figure 12.134 is a rough guide to speeds and lengths of LVDS, RS-422/485, and RS-232 signaling. Figure 12.135 summarizes the signal characteristics of these link standards. And Figure 12.136 compares their driver power requirements (i.e., the driver chip's supply voltage times current), along with that of 50 Ω coax.[87]

[86] There are several closely related protocols, namely PECL (positive emitter-coupled logic), LVPECL (low-voltage PECL), and CML (current-mode logic). The latter is used, for example, in digital-video links such as DVI and HDMI.

[87] The driver ICs we used for these power measurements were: coax – 74LVC2G04 (both sections in parallel); LVDS – 65LVDS1; RS-232 – MAX3232E; RS-422/485 – LTC1485 (5 V) or LTC1480 (3.3 V).

Review of Chapter 12

An A-to-S summary of what we have learned in Chapter 12. This summary reviews basic principles, facts, and application advice in Chapter 12.

¶A. Logic Interconnections.
The subject of this chapter is the interconnection of digital logic signals and logic devices to... *everything*, where "everything" includes (a) other logic devices, (b) input sources (switches, optoelectronics, cables), and (c) output devices (dc and ac power loads, optoelectonics, cables). So it's a long chapter, rich with multiple themes. Here we try to organize these diverse topics into manageable paragraphs.

¶B. Logic Families.
§12.1.1. Contemporary digital logic is owned by CMOS, with the minor exception of some emitter-coupled logic families (ECL, PECL, and LVPECL) and some BiCMOS (ABT, BCT). CMOS inputs draw no current, and their logic thresholds are usually close to mid-supply, except for the -T suffix types (HCT, ACT, AHCT, VHCT) where the threshold conforms to that of legacy bipolar TTL ($\sim$1.4 V); see Figure 12.5.

The 74HC[T] logic is good for easy prototyping and general-purpose use, and is available in DIP or SOIC; substitute 74AC[T] or 74LV for applications needing higher speed. For lower-voltage applications at moderate speeds, use 74LVC or 74VCX (available in SOIC and mini-logic packages only). For operation up to 15 V you can use 4000B logic, but this stuff is *slow*. Finally, for differential signaling use LVDS (or LVPECL, if specified) drivers and receivers. See Figures 12.2 and 12.3 for logic-family speed versus power, and speed versus supply voltage.

¶C. Input and Output Characteristics.
§§12.1.2A and 12.1.2B. CMOS *inputs* draw no current when $0V \leq V_{in} \leq V_{DD}$ (and even for inputs a few tenths of a volt beyond). For most logic families, the input protection diodes clamp to ground and V_{DD}, thus causing substantial input current for input voltages as much as a volt beyond V_{DD} or below GND; this can cause SCR latchup. However, some families use zener-like clamping, allowing input swings well beyond the rail (and even when unpowered); examples include 5V-tolerant 74LVC, and 3.3V-tolerant 74AUP. See Figure 12.4.

CMOS *outputs* swing to the rails, looking resistive for small load currents (Figure 12.7). The output current capability varies widely among families; within a family it increases with increasing supply voltage. See Figure 12.6 for a summary of thresholds, and valid input and output voltage ranges. Note that the outputs of (essentially obsolete) bipolar TTL logic (and also of digital ICs with nMOS outputs) do not saturate to the supply rail; see §12.4.5A.

¶D. Interfacing between Logic Families.
You can mix logic types as long as you respect logic-level input requirements. At the basic level this means that (a) you can always make a direct connection between logic running at the same voltage; (b) logic running from a higher voltage (e.g., +5 V) can drive lower-voltage logic if the latter's input is "tolerant"; and (c) lower-voltage logic can drive higher-voltage logic if the latter has "TTL thresholds" and the former is powered from at least +2.5 V. With dozens of logic families, most capable of operating over a range of supply voltages, you've got a lot of possibilities. Most are dealt with in §12.1.3, and presented pictorially in Figure 12.9.

¶E. Driving Digital Logic Inputs.
§12.1.4. A mechanical switch with pullup (§12.1.4A) generates the right logic levels, but with *bounce*. That may be OK for some applications; but for clean edges you need a debouncer, illustrated in a half-dozen variations in §12.1.4B. A logic input can come instead from a *comparator* (a subject summarized in ¶G below, and treated in detail in §12.3), whose output may have active pullup to the same V_{DD} as the driven logic, or it may come from an "open-drain" (or open-collector) terminal; in the latter case you need an external pullup to V_{DD}. Figure 12.25 shows representative circuit configurations.

When driving logic inputs from signals of any type, be careful not to overdrive the input (whether from a signal source of greater swing, or from transmission-line effects – see the egregious example in Figures 12.103 and 12.104); and do not leave unused logic inputs floating.

Contemporary logic devices are engineered to withstand substantial static-electricity insults (defined in terms of the human-body model, HBM, of 100 pF in series with 1.5kΩ, §12.1.5), but it's best not to tempt fate; be careful to discharge yourself, use antistatic materials, etc. See also the earlier discussion in §3.5.4H.

¶F. Driving External Loads from Logic Outputs.
§12.4. You can drive small loads (LEDs, SSRs, small mechanical relays) directly from logic outputs (§12.4.1 and Figure 12.39), taking care to respect the output drive capability (voltage swing, current), and, for a mechanical relay, adding a diode clamp across its coil. As shown in the figure, you can use an open-collector driver (ULN2003, 75468) to

accommodate voltages and currents as high as 100 V and 350 mA respectively. There are analogous drivers for use with serial data (from a microcontroller), see Figures 12.40 and 12.41, and the listing in Table 12.3 on page 819.

For driving heftier loads you can append an external transistor or power-driving module. For driving *positive* loads see §12.4.2, and Figures 12.42 and 12.43. For driving *negative* or *ac* loads see §12.4.3 and Figure 12.44. Serious power switching requires *fault protection*, see §12.4.4 and Figures 12.45–12.48.

¶G. Comparators.

A comparator is a high-gain uncompensated differential amplifier, used to signal which of two analog voltage inputs is greater. It is an important interface between analog signals and the digital world. Comparators were introduced in §4.3.2, and treated in detail in §12.3 (with listings in Tables 12.1 on page 812 and 12.2 on page 813).

As with op-amps, comparators come in a range of speeds ($<$1 ns to tens of μs), supply voltages (total supply voltages from as little as 1.1 V to as much as 40 V), input offset voltages (0.25 mV to 10 mV), input bias currents (1 pA to $>$10 μA), input common-mode voltage ranges (to both rails, one rail, or neither), and quiescent currents ($<$1 μA to tens of mA). But, unlike an op-amp, which is normally operated in the linear region where the (analog) output voltage avoids saturation, a comparator's output is digital; it lives at the extremes. It may be used to drive logic, or it may drive an ON/OFF load like a relay or an LED. To accommodate various loads, comparator output stages come in a half-dozen variations (Figure 12.33), including (a) rail-to-rail (like an op-amp); (b) open-drain or open-collector; (c) logic-level with auxiliary logic-rail pin V_L; (d) logic-level with both V_L and GND pins; and (e) floating-transistor output stage.

Comparator miscellany: comparators are usually configured with some hysteresis (Schmitt trigger) to prevent multiple transitions and oscillation. Some comparators have quite limited differential input voltage ranges (as little as a few volts). For comparators with BJT input stages, the input currents may undergo an abrupt step at zero differential voltage (e.g., Figure 12.36). The delay and switching time of a comparator depends on the input overdrive (Figure 12.38).

¶H. Optoelectronics.

§12.5. Continuing the theme of "logic-to-everything," we must include among the latter *humans*, with senses that delight in visual presentations.[88] There's much richness here, including emitters, indicators, displays, detectors, and couplers, as illustrated in the photographs in Figures 12.58 (page 830), 12.71 (page 837), 12.80 (page 841), 12.84 (page 844), and 12.95 (page 853). To remind you of the breadth of optoelectronics, here is a condensed listing of devices (expanded in ¶¶I–M below) in the family tree:

Emitters: LED; Laser diode; Electroluminescent

Displays: LED-based; LCD-based; VFD-based; OLED; E-ink

Detectors: Photodiode; Phototransistor; Photoresistor; Bolometer; APD; PMT; HAPD; Microchannel plate

Couplers: LED-input or logic-input, with transistor-, thyristor-, or logic-output

Other: Opto-interrupter, proximity and ranging; Barcode reader; Optical mouse.

¶I. LED Indicators.

These have diode-like *I–V* curves, but with greater (and color-dependent) forward voltage drops (Figure 2.8). They come in many form factors (panel mount, PCB mount, arrays, 7-segment displays), sizes, viewing angles, and colors (§12.5.1). You can drive them from a logic signal of adequate swing, with a current-limiting series resistor (mindful that blue, bright-green, and white LEDs have forward drops of 3 V or more), or you can add an external transistor for higher voltage or current (see Figure 12.61). For serious illuminations purposes you're better off using a current-sensing dc-dc switching converter (Figure 12.62). By contrast, to drive *laser diodes* you use feedback from the integrated monitor photodiode to set the drive current (see Figure 12.67 and associated text).

¶J. Displays.

A popular device for numeric or hexadecimal data is the 7-segment (or 16-segment) *display* (§12.5.3A), available in multicharacter sticks of both "dumb" and "smart" varieties (see Figures 10.90 and 12.77, respectively). Once you've bought into the idea of a multicharacter display device, you should consider the multicharacter (and multiline) LCD smart display (Figure 12.78) and the nicer-looking (and compatible) VFD display, seen in Figures 12.71 and 15.25.

¶K. Detectors.

In the detector arena, *photodiodes* convert light to a proportional photocurrent, and can be operated in either of

[88] As a colleague once remarked, as we were admiring the sunset while walking along the beach, "nice graphics."

two modes – *photovoltaic* (self-generating, into a short circuit or virtual ground); or *photoconductive* (back-biased) – see Figure 12.81. The latter is faster, but suffers from leakage current and increased noise. A *phototransistor*, which operates only in photoconductive mode, combines a photodiode and transistor, for greater gain, but with reduced speed; similarly (and more so) for a *photo-Darlington*. The *avalanche photodiode* exploits the phenomenon of avalanche multiplication in a photodiode biased close to breakdown; each detected photon causes multiple electrons to be collected, thus providing higher gain (in linear operation), or (if biased still further) a full-sized "Geiger-mode" pulse for each detected photon. A different class of detector is the cadmium sulfide *photoresistive* sensor, which behaves like a linear resistance ($I \propto V$) that depends on illumination; CdS sensors are slow, and they are becoming a rarity owing to RoHS regulations (cadmium is not good for you), but they are useful in applications where you want a light-controlled linear resistance (see for example Figure 7.21). Finally, a still-popular old-school photodetector is the *photomultiplier* tube (PMT, §12.6.2), in which a photoelectron is accelerated and collides with successive electron-multiplying dynodes to produce a cascade of some 10^6 electrons at a collecting anode (Figures 12.82 and 12.83).

¶L. Optocouplers and Solid-state Relays.

Optocouplers (also known as opto-isolators, or photocouplers) consist of an opto-emitter combined with a detector in an opaque package (§12.7). They are used to convey digital (and sometimes analog) signals between circuits with separate grounds; such *galvanic isolation* prevents ground loops in sensitive circuits, it permits safe switching of ac powerline circuits, and it allows communication and control for circuits operating at a high voltage.

All optocouplers use an LED at the *input end*, in most cases simply providing the anode (+) and cathode (−) terminals (so you have to limit the driving current with an external resistor, see Figure 12.85A); some optocouplers include current limiting (e.g., SSRs, Figure 12.92C), while some others (e.g., high-speed logic-to-logic couplers, Figure 12.86C) accept a logic-level input signal. At the *output end* there are many configurations: transistor or Darlington (Figure 12.85), digital logic with open-collector or active pullup (Figure 12.86), active-pullup MOSFET-driver (Figure 12.87), ac-output SSR (Figure 12.92), and dc-output SSR in many flavors (Figure 12.91). There are also analog-oriented optocouplers (Figures 12.88, 12.89, and 12.90), ac-input optocouplers (Figure 12.94), and a whole range of industrial *input/output modules* (§12.7.7).

¶M. Opto-interrupters, Proximity Sensors, and Angle Encoders.

An *opto-interrupter* is an LED-detector pair with an open gap (Figure 12.84), widely used to sense end-of-travel in mechanical devices. An optical *proximity sensor* bounces a beam off an external object; the *distance measuring sensor* is a variant that incorporates a position-sensitive detector that uses parallax (triangulation) to measure object distance (to $\sim$1 m or so). Optical *shaft angle encoders* measure shaft position; they come in incremental (quadrature square-wave) and absolute versions, with resolutions of 32 to 128 counts per rotation (for panel controls) to 30,000 or more for high-resolution shaft encoders.

¶N. Fiber Optics.

Light travels happily, and unimpeded, through glass or plastic fiber. For long-haul high-datarate communications you use single-mode glass fiber (with rates to 100 Gbps or more per fiber); but for more modest data communications you can use multimode fibers, ranging from the simplest TOSLINK 1 mm plastic fiber to the very popular 62.5/125 μm graded-index glass fiber. Transmitter and receiver modules are widely available (§12.8.1–12.8.3), as well as high speed transceiver modules that include all necessary driver/receiver electronics (§12.8.4).

¶O. Digital Signals and Long Wires.

Fully 25% of the chapter (§12.9) is devoted to the problem of sending digital signals somewhere else, where the *somewhere* may be on the same PCB, or through a backplane, or through a cable to a remote electronic device. As simple as these may seem, numerous pitfalls await the unwary: for example, we were surprised (but should not have been) when encountering errors while sending parallel digital data of only modest speed (HC logic) through just 4″ of flat ribbon cable. Read the following paragraphs, to avoid membership in the club of the unwary.

¶P. Short Links.

For the *short links* that stay on a PCB, the distributed capacitive loading and consequent ground-current spikes cause logic glitches (§12.9.1A), requiring liberal power-supply bypassing (which is always salutory) and low-inductance ground plane layouts. The problem is less severe in synchronous systems; but it's more severe with fast logic, where interconnections may require constant-impedance transmission-line layouts (see §*1x.1.4*). These problems are more pronounced for signals moving between circuit boards or along backplanes, where you may need to use line driver and line receiver ICs, perhaps in

combination with controlled impedance traces that are properly terminated; the latter is standard practice with LVDS (differential) signaling, as seen for example in Figure 12.127.

¶Q. Cable Runs.

Longer signal runs require *cable* (§12.10), usually *coaxial* (e.g., the ubiquitous RG58 BNC patch cords), multiwire *ribbon*, *twisted pair* (e.g., the comparably ubiquitous cat-5 or cat-6 network cable), or *fiber optic* (generally recognizable by their brightly colored svelte profile). With the exception of "slow" waveforms,[89] signal cables must be treated as *transmission lines* (Appendix H), with their *characteristic impedance*[90] Z_0. For coax, Z_0 is usually 50 Ω (but 75 Ω for video), while twisted-pair is usually 100 Ω. The significance of Z_0 is that *a signal applied to a cable, terminated with a load resistance equal to its characteristic impedance, is completely absorbed by the load*; it does not reflect. No other load impedance can make that statement.

¶R. Terminating Digital Coax.

So you can (a) put a 50 Ω resistor across the far end of a coax line (a "terminated line," or "matched line") and drive it with your digital signal (which sees a pure 50 Ω resistive load – the capacitance disappears!), as in Figure 12.107; or (b) drive such a terminated line with a signal of the same impedance, as in Figure 12.110, noting that the output signal amplitude is reduced by ×2; or (c) keep the matched source resistor but omit the load resistor, as in Figure 12.107, to preserve the full output amplitude. In the latter case there *is* a reflection of substantial amplitude, which however is hospitably absorbed by the source resistor – see the highly educational waveforms in Figure 12.115. See §12.10.1 for pretty pictures, and plenty of detail.

¶S. Differential Cable.

Differential signals are treated similarly, with a termination connected across the pair. For long hauls (to ∼1 km) and modest speeds (10 kbps–10 Mbps) the long-lived RS-422/485 protocol is popular, whereas the LVDS protocol is used for higher datarates (10 Mbps–1 Gbps) over shorter distances; see the speed limit contours in Figure 12.134. To achieve the highest datarates over significant lengths of cable one commonly uses driver *pre-emphasis* and receiver *equalization*, see Figures 12.130–12.132 and associated discussion.

[89] Those whose transition times are much longer than the round-trip travel time, $t=2L\sqrt{\varepsilon}/c$, where the factor $\sqrt{\varepsilon}$ accounts for the slower-than-light "velocity factor" of wave propagation in the cable.

[90] Which should really be called characteristic *resistance*.