Chapter 10
Receiver Fine Points

The superheterodyne receiver, described in the previous chapter, is the cornerstone of modern electronics. The superhet is used in car radios, hi-fi receivers, TV sets, cordless and cellular phones, and even radar detectors. Although the basic block diagram of a superhet is the same regardless of the kind of receiver, however, there are still quite a few differences between different receivers.

Some of the differences we have already looked at, such as double-conversion, or using a converter instead of separate mixer and oscillator. But let's look at some others.

RF Frequency Range

Different receivers tune in different ranges of radio frequency signals. For example, a commercial AM radio will tune from 550 to 1650 kHz, while an FM radio will tune from 88 to 108 MHz; A TV set will receive signals between 54 MHz and about 850 MHz, while a radar detector will receive signals around 10 GHz. Thus the actual circuitry will be different as well. Depending on the frequency range, the antenna will be different, the RF amplifiers and mixers will be different, and even the oscillator must be different.

AM broadcast signal strengths tend to be higher, and so AM broadcast radios usually don't have an RF amplifier (except in car radios), while the FM radio will almost certainly have one. This is usually a cost issue, but there are often technical reasons as well. For example, radar detectors do not have an RF amplifier because amplifying RF signals at radar frequencies is just too expensive. High frequency communications receivers, on the other hand, almost always have an RF amplifier stage, or even two.

Selectivity

The required selectivity depends on the bandwidth of the signal. Selectivities often range from just a few hundred Hertz (for receivers designed for receiving Morse code or slow-speed digital signals) to 6 MHz (for a TV receiver) or even more.

The proper selectivity is often obtained with several tuned circuits tuned to the IF frequency, often with double-conversion to provide lower selectivity without having image problems (see the previous section.) But other kinds of filters are also sometimes used, such as ceramic or crystal resonators, or SAW filters — Surface Acoustic Wave filters. These kinds of filters use mechanical, rather than electrical resonance, to provide a narrower bandpass. But more important, they provide a more stable and repeatable response without having to be individually adjusted.

Stability

They say that a good politician is one who, once bought, stays bought. The same is true of receivers — once a receiver is tuned to a station, it should stay tuned to it. When it does, we say that it is stable; when it doesn't, then we say that it *drifts*.

In a superheterodyne receiver, stability is most affected by the oscillator. In general, oscillator stability is most of a problem when the carrier frequency is high, but the bandwidth is small. In this case, even a small percentage change of the oscillator frequency can cause the signal to drift out of the passband of the receiver.

For broadcast AM receivers, stability is not much of a problem, except perhaps in car radios whose temperature may change tremendously from the time you start driving on a cold day until the heater warms up the entire area under and around the dashboard.

Oscillator drift at higher frequencies is much more of a problem, and so the FM radio's oscillator will probably be quite different. Oscillators in AM radios tend to be spread out on a printed circuit board, while oscillators in FM radios will be very compact, and may even be enclosed in a shielded, metal case to prevent outside effects from affecting the circuit. An automatic frequency control (AFC) circuit (described later) may be used to keep the frequency constant. The AM radio will probably use a converter, whereas the FM radio will most likely use a separate oscillator and mixer.

Another difference lies in the detectors — an AM radio may contain a simple diode to rectify and demodulate the carrier, while an FM radio might use a Foster-Seeley discriminator or some other type of FM detector.

Since the bandwidths of AM and FM signals are different, the IF frequencies will be different. AM radios tend to use a 455 kHz IF (frequencies between 250 and 500 kHz are also sometimes used), while the FM radio will probably use 10.7 MHz.

AM/FM radios will use a combination of the above. Most such radios have completely separate RF and mixer/converter sections for each band, but use a common IF section for both. They do this by connecting a 455 kHz IF transformer in series with a 10.7 MHz IF transformer, and using the same transistor to amplify both. There will be separate AM and FM detectors, but then a common audio amplifier for both. There is obviously an art to getting the most performance from the least number of components.

Once you leave the simple, inexpensive AM and FM broadcast receiver and go either to a more expensive hi-fi or car stereo set, or to the type of receiver used in communications (amateur as well as commercial), then new circuits and options come into play. The rest of this section describes some of these.

Types of Oscillators

Some radios (such as the typical home broadcast receiver) are frequently retuned from one station to another; others (like taxicab or oil truck radios) may stay tuned to one particular frequency their entire life. Thus the oscillator in one radio may be tunable, while the oscillator in another radio may be fixed on one frequency for years. The oscillator is therefore an important part of the design.

Inexpensive radios (such as pocket transistor radios) generally use simple LC oscillators; the frequency is set by an inductor and capacitor. Since components tend to change size and value as the temperature changes, special temperature-compensating capacitors are often used to try to keep the oscillator frequency constant. Even so, many FM radios need an AFC or Automatic Frequency Control circuit to keep the radio in tune, because otherwise the LC oscillator would still drift.

Single-frequency radios, on the other hand, generally use *crystals* to set the oscillator frequency. As shown in Fig. 10-1, a crystal is a small piece of quartz, roughly the size of a dime. It has a plated area on each side, and is clamped between two thin spring-like wires, which hold it in its holder. Because it is only held on its edges, it is free to vibrate.

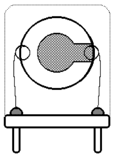

Fig. 10-1. A Quartz crystal

Quartz is the same piezo-electric material we discussed back in Chapter 1: it converts between electricity and motion — when you bend it, it generates a voltage; if you connect a voltage to it, it bends. In this case, the small disk of quartz has a natural resonant frequency, typically in the range of 0.1 to 25 or so MHz, depending on its size. If you connect it in the feedback path of an amplifier, the amplifier starts to oscillate at the resonant frequency of the crystal. This frequency is very stable, and remains constant (within a tiny fraction of a percent) over a long time.

Crystals must be ordered for the specific frequency you need. Some frequencies (such as the 3.579545 MHz color burst frequency in a TV set) are used so often that the crystals are mass-produced in huge quantities, and cost less than $1 each; crystals for other frequencies may have to be individually made at substantially greater cost.

You may remember that we said that crystals typically resonate in the range of 0.1 to 25 MHz or so, whereas the first oscillator in many radios may need to oscillate at much higher frequencies. A higher frequency can be produced by several methods.

A typical case might be an amateur radio which needs a first oscillator frequency of 136.24 MHz. This is usually done by using a crystal around 15 MHz, which oscillates on its third harmonic of 45.41333 MHz; instead of oscillating in one piece, the crystal oscillates in one-third sections at approximately three times its fundamental frequency. This is its third harmonic, which should properly be called its second overtone, but these crystals are often (wrongly) called third-overtone crystals.

Once the crystal generates the 45.41333 MHz signal, a *tripler* circuit then multiplies the frequency by 3 to make the required 136.24 MHz. A tripler circuit is an amplifier which purposely distorts the signal to produce harmonics; a tuned circuit in the output then selects the third harmonic and deletes the rest. (A doubler would be tuned to the second harmonic; although it is possible to build quadruplers, quintuplers, etc., this is seldom done because of technical problems. Instead, combinations of doublers and triplers are usually used.)

Since crystals have a fairly limited range of frequencies, it is common to use chains of doublers and triplers to multiply the frequencies up to what is needed, or else use digital dividers (flip-flops) to divide the frequency down to a lower value.

Multiplying frequencies up to a higher value by using doublers and triplers is easy for certain values. For example, multiplying a frequency by 12 is easily done with a tripler and two doublers (since $3 \times 2 \times 2 = 12$), but multiplying by 11 or 13, or most other numbers, cannot be done with doublers and triplers. Instead, we use a phase-locked loop (described shortly).

AGC, AVC, AFC, and BFO

This alphabet soup describes a few other circuits often found in receivers.

The *Automatic Volume Control* (AVC) or *Automatic Gain Control* (AGC) circuits are basically the same. In each, the output of the detector is sampled to determine how strong a station is. On strong stations, this circuit lowers the gain of the IF amplifier, while on weak stations, this circuit increases the gain. Since strong signals are amplified less, while weak signals are amplified more, the result is to make both strong and weak signals sound the same.

An *Automatic Frequency Control* (AFC) circuit is used in those FM receivers whose oscillators tend to drift in frequency. While modern receivers with phase-locked-loop oscillators don't need the AFC circuit, almost all other oscillator circuits tend to drift and therefore can use AFC. The AFC circuit checks the received signal at the detector to see whether the radio is correctly tuned to the station. If not, then it sends a correction signal back to the oscillator, forcing it to either increase or lower its frequency to bring the station back into perfect tune.

The *Beat Frequency Oscillator* (BFO) is used in some older receivers to receive Morse code and DSB or SSB transmissions. When receiving a DSB or SSB signal, the BFO supplies the missing carrier that is needed to go with the sideband(s). When receiving a Morse code (called a *Continuous Wave* or CW signal), the BFO supplies a continuous signal near the IF frequency, which beats with the received carrier to produce an audio tone; this is what the operator hears and decodes from Morse code into text.

PLL — the Phase-Locked Loop

The PLL or *Phase-Locked Loop* is a fairly recent development in electronics. One of its uses is to multiply crystal frequencies by unusual numbers; it has the stability and accuracy of a crystal oscillator but, unlike a crystal, it is tunable.

As Fig. 10-2 shows, the PLL circuit looks like a traditional servo loop circuit (similar to the one we showed in Fig. 8-4.) Its output comes from a VCO or *voltage-controlled oscillator*. This oscillator generates the frequency we want, but instead of the frequency being controlled by a knob or control of some sort, it is controlled by an input voltage V_c. The VCO can oscillate over a wide range of frequencies, and V_c tells it what frequency to output. Usually, if V_c is high, the output frequency is high, while if V_c is low, the output frequency is low.

Besides going to the output, the f_{out} signal also goes through a digital circuit which divides its frequency by some integer N. N is usually a variable number, which can be dialed in from some front-panel control, or perhaps is set by a small computer chip in the radio.

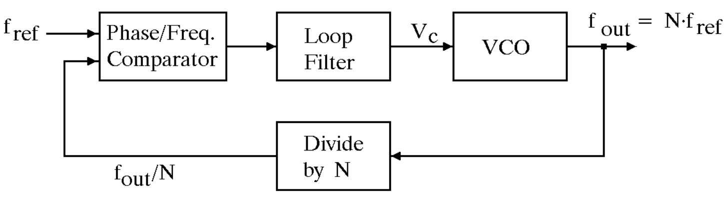

Fig. 10-2. Block diagram of a Phase-Locked Loop

The output from the divider circuit is a feedback signal, which now has a frequency f_{out}/N. It is sent back to a *phase/frequency comparator*, which also receives an input reference signal whose frequency is f_{ref}. The comparator looks at the two inputs, and compares their frequencies. If the frequency of the signal coming back through the divider is lower than the input signal f_{ref}, then the comparator sends out an error signal which goes through the loop filter to the VCO, and tells the VCO "raise your frequency, it is too low." The VCO frequency then goes up, which raises both f_{out} and also the feedback signal, to the point where the two inputs into the comparator match. (Of course, if the feedback signal was too high in frequency, the comparator would tell the VCO to lower it.) In this way, the PLL varies the VCO frequency up and down so it can lock it at the value that makes the two inputs into the comparator equal. When this happens, we say that the loop is *locked*.

In operation, the f_{ref} input signal usually comes from a very stable crystal oscillator. Since the digital divider circuit divides by a number N which is also exact (it is an exact integer), the output signal f_{out} is exactly equal to $N \times f_{ref}$, and is therefore also very stable.

The secret of the system lies in the divide by N circuit. This is a digital circuit which, with just minor changes, can divide by many different values of N, and each different N produces a different output frequency.

For example, suppose you needed an oscillator tunable to any exact MHz between 10 and 20 MHz. It would be almost impossible to get that sort of accuracy from an LC oscillator, and it would be very expensive to use 10 different crystals, one for each frequency. Instead, you could use a PLL with a 1 MHz crystal, and a divider which can divide by any integer between 10 and 20. If the crystal frequency was accurate to, say, ±0.001%, then the output frequencies would be just as accurate. A PLL, used like this to generate many different output frequencies, is called a *frequency synthesizer*.

Phase locked loops are very common today. If you have a car radio with a digital readout, it most likely uses a PLL to set the exact receive frequency. Before PLL's, a six-channel walkie-talkie needed six crystals for the receiver, and another six for the transmitter, just to set the frequencies; modern synthesized walkie-talkies can access hundreds of transmit and receive frequencies with just two or three crystals.

Before we leave phase-locked loops, we should mention a few other uses.

One use is as an FM detector. Suppose we replace the divide-by-*N* circuit with a plain wire (which makes *N* equal to 1, so the output frequency is the same as the input reference frequency.) Further, let the f_{ref} signal, rather than being a very stable reference signal, be an FM modulated signal (perhaps coming from the IF amplifier in a receiver.) As the signal frequency deviates back and forth, the PLL will keep the loop locked, forcing the VCO frequency to vary up and down in step with the input signal. The V_c signal going into the VCO tells the VCO what it has to do to match the input frequency change. In other words, the V_c signal goes up and down with the frequency — it is the demodulated output.

The PLL has one very useful characteristic — given enough time, it can lock onto an input reference signal, even if that signal is buried in noise. It is therefore an excellent circuit for detecting very weak signals. Furthermore, since the VCO output frequency is the same as the input frequency — but without the noise — the PLL can be used to "clean up" noisy signals. The disadvantage, however, lies in the words "given enough time." The more noise there is in the input signal, the slower the PLL will lock to it; hence the signal must deviate slowly or the PLL will unlock and lose it.

Direct Digital Synthesis

Direct digital synthesis is a newer way of generating adjustable, yet very precise frequencies. This approach has only become practical in the last few years, with the arrival of some very fast integrated circuits.

Fig. 10-3 shows the idea. The top waveform in the figure shows a typical sine wave you might want to generate. The first step is to take some very careful measurements of the height of the wave. For example, at the very left, we show an amplitude of 0 volts. A bit further on, the wave has a height of 0.174 volts. A bit farther on, the height is 0.342 volts, and so on. You do this for the entire length of one cycle.

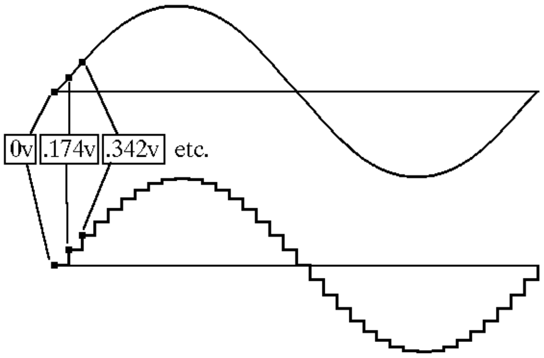

Fig. 10-3. Digital Synthesis of a sine wave

The next step is to build a digital circuit that stores these measurements as binary numbers, and that can output them at high speed. The trick is to control the output speed so that the measurements come out at the exact instant when the wave should be at that amplitude. For example, if you took 10 measurements of one cycle of the sine wave, and wanted a 1 MHz output frequency, then you would need $10 \times 1{,}000{,}000$, or 10,000,000 numbers to be output in exactly one second. You can see why this requires some very fast circuitry.

Once you have this digital circuit which is throwing out numbers at a very fast rate, telling you how high the wave should be at any particular instant, you need to convert those numbers back into voltages. This is done with a *digital-to-analog converter,* which accepts the digital numbers, and converts them into an analog voltage.

Since the digital numbers represent specific points on the curve, and not the full sine wave itself, the output is the lower curve in Fig. 10-3. While this sort of looks like a sine wave, it has a lot of steps in it which represent the missing data. Fortunately, these steps represent high frequency signals which are easily removed with a low-pass filter, giving us a sine wave again.

Subcarriers

A subcarrier is a carrier which is carried on top of another carrier. Here is an example of a common use:

Suppose you have some music, such as what I call elevator music — background music that might be used in elevators or stores — which you'd like to broadcast to paying customers in a city. But you don't want to spend money to build a complete broadcast station; moreover, you don't want the music to be interrupted by commercials or even station call letters. You also don't want the general public to hear this music for free, because you want to charge your customers for providing them with commercial-free background music for their stores or buildings.

For this application, hi-fi music isn't needed, so let's suppose that the music has a frequency range up to perhaps 5 kHz or so. So you amplitude modulate it onto a carrier of, say, 91 kHz. The result is a 91 kHz carrier with sidebands extending from around 86 kHz up to about 96 kHz. These frequencies are well above human hearing, but still too low to efficiently transmit over the air — the antenna would be too long.

You therefore make a deal with a local FM broadcast station to transmit this signal for you. Because your carrier frequency is well above the range that the ear can hear, they can mix it into their audio signal without their own listeners being able to hear it. So we now have a main carrier (the FM station's carrier between 88 and

108 MHz) which carries both the regular FM station's signal, as well as your carrier signal, which occupies 86 to 96 kHz. Your carrier is now a *subcarrier*, riding on top of another carrier.

The system just described is actually quite common; FM stations call it an SCA subcarrier, and may use it not just for background music, but also for foreign-language programs, paging signals, or even digital data such as stock market prices. Another example is the color subcarrier in a TV signal; this is a 3.579545 MHz subcarrier riding on top of a standard TV signal; it carries the color information.

Stereo FM

Stereo FM is another example of how a subcarrier is used to carry additional information on an FM signal, but it is a bit more complicated than the plain SCA approach. Fig. 10-4 shows one way to do this. (Actually, Fig. 10-4 shows how this was done in early stereo transmitters; modern transmitters use a digital approach called TDM, but its explanation will have to wait until a later chapter.)

The signals from the left and right microphones of a stereo signal are amplified by two audio frequency (AF) amplifiers, and sent to a *matrixing circuit* consisting of four resistors. Resistors R_1 and R_2 take the left and right signals (called L and R in the figure) and mix them into a signal called L+R. This *sum* signal combines the left and right signals into one, for those listeners who have a mono rather than a stereo receiver. These listeners therefore get all audio, though both the left and right channels will play through a common channel.

At the same time, an inverter takes the R signal and inverts it into a signal called –R. Resistors R_3 and R_4 combine this with the left signal into the L–R signal. This *difference* signal represents only the difference between the left and right channels. For example, if a singer is picked up equally by the left and right mikes, the two signals will cancel out (since the R signal is inverted) and not appear in the L–R signal. Instruments at the left or right, on the other hand, will appear mainly in one channel or the other, and so will not cancel out.

The L–R signal is modulated onto a 38 kHz subcarrier in a double-sideband (DSB) modulator. If the audio extends from about 0 to about 15 kHz, then the sidebands will extend ±15 kHz out from the subcarrier, or from about 23 kHz to about 53 kHz. Since this is a DSB signal, not AM, there are only the two sidebands — the subcarrier itself is missing.

At the same time, the 38 kHz signal that was used to produce the DSB signal is divided by 2, to produce a 19 kHz signal called the *pilot tone*. This signal is exactly ½ the frequency of the (missing!) 38 kHz carrier.

All three of these signals — the main L+R audio signal, the DSB L–R sidebands, and the pilot tone — are fed to the main FM modulator, which modulates all three of these onto the main FM station carrier. The 38 kHz subcarrier (or actually, its upper and lower sidebands, since the carrier itself is not there) therefore becomes a subcarrier on the main carrier. Fig. 10-5 shows the resulting spectrum of the audio signal carried on the main FM station carrier. It looks like this:

- Main L+R signal , up to 15 kHz
- Pilot tone, 19 kHz
- L–R lower sideband, 23 to 38 kHz
- Missing carrier (not shown), 38 kHz
- L–R upper sideband, 38 to 53 kHz
- Room for possible SCA subcarriers, from 53 to 100 kHz

With signals extending up to a possible 100 kHz, the overall bandwidth of the FM signal can be up to 200 kHz.

Why is the 38 kHz information sent as DSB rather than plain AM or FM? For two reasons.

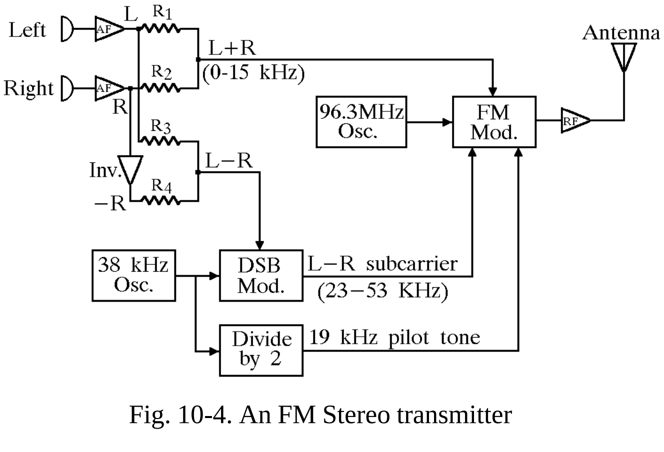

Fig. 10-4. An FM Stereo transmitter

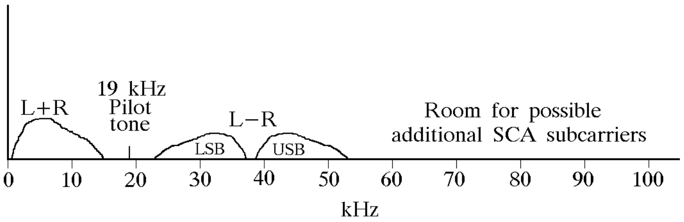

Fig. 10-5. Stereo FM spectrum

First, it is not FM because an FM subcarrier at any reasonable modulation index would have an excessive bandwidth; it would extend down below 23 kHz, and start to interfere with the main L+R channel. And it is not AM because that would increase overall noise. Remember that each time a anything is added to the main carrier, the volume level of the main signal (the L+R signal in this case) has to be reduced, so as to prevent the overall signal from being over-modulated. By omitting the subcarrier itself, and including only its sidebands, less reduction of the main signal is needed. This provides a louder, less noisy signal to listeners far away from the transmitter.

But as we mentioned in chapter 7, DSB is not normally used for transmitting music or high quality audio, since it is too difficult to re- insert the carrier exactly half-way between the two sidebands. This is where the 19 kHz pilot tone comes in. Remember that the 19 kHz pilot tone was generated at the FM transmitter by dividing the 38 kHz subcarrier frequency by exactly 2. If the receiver takes this pilot tone and multiplies its frequency by exactly 2, it will reconstruct the absolutely correct 38 kHz frequency that it needs to put it exactly midway between the two DSB sidebands. The pilot tone is transmitted at a very low level, just strong enough so it can be separated out and its frequency doubled. In this way, the pilot tone helps the receiver provide a carrier at exactly the right frequency, so that the DSB signal can provide good quality audio after all.

Once the receiver demodulates the L–R signal, it uses another matrixing circuit to mix the L+R and L–R signals and produce the pure left (L) and right (R) signals. This is usually shown with the following equations:

(L+R) + (L–R) = 2L (pure left)

and

(L+R) - (L–R) = 2R (pure right)

This rather complex approach to stereo has two inherent advantages. First, the L+R main signal provides a compatible signal for listeners with mono receivers; compatibility was a very important criterion when the FCC decided on this stereo system. Second, the L–R signal is somewhat more prone to noise pickup than the L+R signal; since it is mixed equally into both the left and right channels, the noise comes equally out of both channels. This is more acceptable to listeners than having one good channel and one bad channel. And if the noise does become too high for comfort, the user can always switch the receiver back to mono mode; the radio then receives only the L+R signal, which avoids the noise of the L–R signal.

Incidentally, in addition to the stereo subcarrier, FM broadcast stations can also have one or two SCA subcarriers. Though adding all these signals does reduce the overall signal-to-noise of the station, the economics of charging extra for SCA channels does make it attractive to many stations.

Spread-Spectrum

The concepts of spread-spectrum date back some 50 years, and maybe more, but until recently, the circuitry required to use it was so expensive that only the military found it feasible. In the last few years, however, spread-spectrum equipment has become quite common in many areas.

Consider the following example. Suppose your name is Kilroy, you were supposed to meet a friend in some room, and you want to leave him a message that you've come and gone. The problem is that you have an enemy, and you don't want *her* to know that you've been there.

You could scribble “Kilroy was here” on the wall, but that's too obvious. (Besides, graffiti is ugly!) So you decide to leave a little note that reads KILROY WAS HERE. The problem is — where to put it?

You don't want your enemy to know that you left a message. Even if she knows, you don't want her to find and read it. Worse yet, your enemy might remove it, or perhaps change it to read KILROY WILL BE HERE. Or someone completely new might come into the room, need a piece of paper to write on, and scribble all over your note.

This is a problem that is faced by the military. When sending an important message, (a) you don't want the enemy to know about it, (b) if he knows about it, you don't want him to be able to read it, (c) if he can read it, you don't want him to change it into something quite different, and (d) you don't want a transmission from someone else to interfere with your message and prevent it from getting through.

So back to the room. You spot a telephone directory lying on the table, so you write your KILROY WAS HERE message into it. You put the K on page 55, the I on page 113, the L on page 38, and so on. In other words, you spread your message throughout the directory, hiding it amid the printing that's already there.

In order to find your message, someone would have to know exactly where to look for it. Your enemy won't know that; in fact, she may not even notice the extra few letters written in amongst the “noise” of all the other stuff in the telephone directory. But even if she notices the extra printing, she won't know the order to put the letters back into. (Obviously, though, you'd better tell your friend where to look!)

Now suppose someone else comes into the room, and decides to leave another message in the phone book. This new person is very unlikely to pick the exact same pages that you did, so the new message will probably not interfere with yours. Even if, by some chance, a few of the page numbers he chose happened to be the same ones you already used, at most he might overwrite one or two of your letters. Your message might now read KILXOY WAM HERE, but that's still enough to get the message through. Better than putting your entire message in one place, and taking a chance on its all being obliterated.

This silly little example is actually quite useful in answering some basic questions. For example, how should you pick the page numbers on which to write? There are several ways:

(a) Roll some dice to get completely random page numbers. Nice, because that makes it really tough for someone else to figure out the sequence. It also helps to spread the message throughout the entire phone book. But now you will have to give the list of page numbers to your friend. These numbers would be *random* if there is no pattern to them — even knowing all the past numbers that were used, you can't predict the next few numbers.

(b) Build two sets of loaded dice, one for you, one for your friend, set up so they both roll the same numbers. You roll your dice to get a set of page numbers. Later, your friend rolls his set of dice in exactly the same way, gets the same numbers and bingo. These page numbers look random, but actually they are not — if you have the loaded dice (or know how they were built), you can duplicate the set of numbers at any time. Such a set of numbers is called *pseudo-random*.

(c) Come up with some other way of generating pseudo-random numbers. For example, there are digital circuits that can generate long strings of digits that look random, but are not. Or there are math formulas that can do it. As long as you and your friend both have the same circuit or formula, you can duplicate the same sequence of page numbers.

(d) If you need to do this every day, should you reuse the same numbers each day, or should you start over with a different set? If your main aim is secrecy, then you should use a different set each day. But if you're just concerned with keeping down interference from or to other people, then reusing the same numbers every time can be enough.

Now let's see how this applies to radio signals. The traditional idea is to transmit a radio signal on a carrier with a fixed frequency, and to try to limit the bandwidth as much as possible to avoid interference to (and from) others. But a radio signal like this is easy to find (especially with a spectrum analyzer), and easy to interfere with or jam.

Spread-spectrum radio, on the other hand, takes that signal, spreads it over a very large band of frequencies, and does it in a way that looks quite random and unpredictable (but is not!) It is the very opposite of what communicators have been trying to do since the beginnings of radio.

Spreading a signal out over a large band of frequencies also has the effect of bringing it down into the noise. Consider a glass of water. If the glass has a diameter of, say 2 inches, then it might be perhaps 4 inches tall. Pour the same amount of water into a glass 4 inches in diameter, and the water is now only 1 inch deep. Pour it on the basement floor, and it spreads out over a large area, but becomes only a tiny fraction of an inch deep. In the same way, a 10-watt RF signal all on one frequency stands out like a sore thumb. Spread it out over a few Megahertz, and the amount of power at any one frequency is so small that it's almost impossible to measure. It blends into the noise.

There are several different ways to spread the signal over a large bandwidth; the two most common are frequency hopping, and direct sequence.

Frequency Hopping

Frequency hopping is just what the name says — the transmitter, rather than continuously transmitting on one frequency, is constantly hopping from one frequency to another. This can easily be achieved by using a pseudo-random number generator to drive a phase-locked loop. Both the transmitter and receiver must use the same number generator to make sure that each time the transmitter hops to a new frequency, the receiver will go there too.

Your signal actually uses a lot of bandwidth — it spreads out over a large spectrum — but any particular part of that spectrum is used only a bit. Not only is it hard to find such a signal, but it also generates relatively little interference to others, because it never stays on any one frequency long enough to really bother anyone. It also picks up little interference from others: if there is some other interfering transmitter on a particular frequency, your receiver will be on that frequency only a short time, probably not long enough to bother you or him. (And some frequency hopping systems actually avoid frequencies that are in use by other systems.)

Frequency hopping can be used to send an analog voice signal, but during the times that the transmitter and receiver are hopping from one frequency to another, there would be short breaks or glitches in the signal which would be very annoying. So a more common approach is to use a *codec* — a coder/decoder which does an analog-to-digital conversion to change the analog voice signal to digital data — and then send the digital data in short but rapid bursts. Between the bursts, the transmitter shuts off, switches to the new frequency, turns on, and then sends the next burst of data. The receiver collects the bursts of data, slows them down and converts them into a continuous stream of data which, when converted back to sound with another codec, results in continuous speech.

This method has one other advantage. Once the sound is in digital form, error correction can be used to correct for missing or wrong bits of data. In this way, even if two transmitters occasionally hop on the same frequency and interfere with each other, the error correction removes the resulting errors.

Direct Sequence

Direct Sequence Spread-Spectrum (DS-SS) is completely different from frequency hopping. Let's go back to our analogy of hiding messages in a telephone directory. As we described it, each successive letter of the message KILROY WAS HERE went on a different page. This matches frequency hopping quite well, where each successive part of a signal is sent on a different frequency.

Direct sequence spread-spectrum (DS-SS) is not like that. Imagine that the telephone directory has 100 pages. Take K, the first letter of the message, break it up into 100 pieces, and put a little piece of it on every page of the book. Then do the same for every other letter of the message. At the end, every page of the directory has a tiny bit of every letter. In a sense, your KILROY WAS HERE message is smeared all over the book! That's direct sequence!

Let's use a spread-spectrum cordless phone as an example. Suppose you want to transmit a telephone-quality voice signal having a bandwidth of perhaps 3000 or 4000 Hz, between the handset and the base of the phone. With ordinary AM or FM, it could be sent in a radio signal with a bandwidth as narrow as 6 or 8 kHz (and even less with single-sideband.)

With DS-SS, the process is a lot more complicated. We start by converting the voice signal into a digital signal with a codec. This typically gives us 64K bits per second of digital data (we will discuss this process more fully in a later chapter.)

DETOUR

A quick review: For resistors, a K is 1000 ohms. For computer folk, however, a K is 1024, so 64K would normally be 64 × 1024 or 65,536, not 64,000. In this case, however, the number really is 64,000 bits per second, obtained by multiplying 8,000 measurements per second times 8 bits per measurement.

END OF DETOUR

If we ignored the fact that the 64K bps output of the codec has sharp edges (and therefore harmonics) and tried to modulate this onto a carrier the pre-spread-spectrum way, we would have a total bandwidth of 64 kHz.

Direct sequence spread-spectrum takes an extra step, however, before it modulates the carrier. It takes each bit of the data coming out of the codec, and replaces it with a whole batch of new bits. In a typical spread-spectrum cordless phone, for example, each bit of the codec output is replaced by 32 bits. These bits break up each codec bit into 32 pieces called "chips". One set of 32 chips, called the *N code*, replaces each 0, while another set, called the *P code*, replaces each 1 in the codec output. In Fig. 10-6 (a), we see the signal as it might come out of the codec. In (b) we see what happens when the N code replaces the two 0 bits, and the P code replaces the 1. (In this figure, the N code for the two zeroes is the same; for greater security, more complex systems might change the P and N codes from bit to bit.)

Obviously, the resulting signal has much higher frequency components; in general, replacing the original bits with their P and N codes increases the frequency range by a factor of 32, which increases the bandwidth by a factor of 32 as well. In this case, the bandwidth goes from 64 kHz to over 2 MHz, much more than before. But this means that the transmitted power is spread out over a much wider bandwidth — sort of like spilling a glass of water on the floor. The same amount of power (or water) is still there, but by spreading it over a larger area, the depth at any spot is very small. In some cases, the power may be so small that it's at or even below the normal noise level.

As you can imagine, a direct sequence spread-spectrum receiver is an interesting device. It starts off with a fairly normal superhet design, except that the bandwidth has to be wide — as wide as the signal. All of the inter-

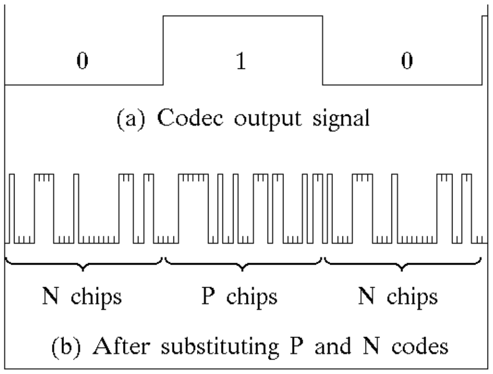

Fig. 10-6. Spread-spectrum chips

esting work occurs after the detector. Because the transmitted signal is spread out over such a wide bandwidth, the output from the detector (usually a phase-modulation detector) looks very much like noise. It would be almost impossible to recover the desired signal, except for one thing — the receiver knows what the P and N codes are supposed to look like! So it knows what to look for.

The circuit that looks for the P and N codes is called a correlator. *Correlation* is a mathematical term which describes how similar two things are to each other by comparing them, item by item. For example, suppose you toss a coin four times, and get tails, heads, heads, and tails; call this THHT. If your friend tosses a coin and also gets THHT, that's a perfect match. You both got tails on the first toss, heads on the second, and so on. But note that this is not the same as THTH. THTH also has two heads and two tails, but they aren't in the same places — the first and second toss were the same, but the third and fourth are different. So only half of the tosses matched.

Now, suppose you toss a coin 100 times, and your friend also tosses 100 times. On the average, you'd expect 50 of your friend's tosses to be the same as yours, and 50 to be different. So if you get somewhere around 50 matches out of the 100, that doesn't show anything special; that's just the way random events happen. We'd say that this is *uncorrelated.*

But if all 100 of your friend's tosses exactly match all 100 of yours, that's suspicious. Sort of like two students who take a true-false test in a subject they know nothing about, and get exactly the same answers (some right, some wrong.) Suspicious, right? I would say that these two sets of answers are correlated — that one student probably copied his answers from the other. That's a correlation of 100% (or just +1.)

But what if the two students get answers that are completely different? Each time one answers TRUE, the other answers FALSE. I'd say that's also suspicious — almost as though one copied from the other, but purposely changed his answers because he decided the other chap was always wrong. In our case, we'd say this is –100% correlation (or just –1.)

So if two signals are very similar, the correlation is close to +1; if they are opposites, the correlation is –1; if they aren't related to each other, the correlation is 0.

Now back to a very simplified explanation of the direct sequence spread-spectrum receiver. The signal coming out of the detector is noisy, but at any instant that noise might be positive or negative. This voltage is sent to a capacitor, which averages that voltage over the length of one chip. At the end of that time, the capacitor voltage may be slightly positive or slightly negative, and the circuit uses that voltage to decide whether that chip seems to be a 0 or a 1. It then sends that chip into a digital circuit which stores it, as well as the 31 previous chips. In other words, this circuit (called a shift register) stores the last 32 chips that have come out of the detector. (The shift register always stores the latest 32 chips; each time a new chip comes out, the oldest chip in the register gets pushed out, so the very last 32 chips are always in the register.)

Now the correlator goes to work. The receiver knows what P and N codes the transmitter is using. So the correlator continuously looks at the 32 bits in the shift register, and compares them bit-by-bit with the 32 bits in the P code and the N code that it knows the transmitter used. Each time it finds a match, it says, "Aha! I got one!" (For those of you mathematically inclined, this process is called *convolution.*)

With 32 bits in a P or N code, there are 2^{32}, or more than four billion possible patterns to the code. So it's not likely that a random signal coming out of the detector is going to exactly match either the P code or the N code. For that matter, since the incoming signal is noisy, it's not likely that even a fairly strong transmitted signal will exactly match the P or N code either. But if most of the bits match (and it's up to the designer to define what is meant by "most"), then the correlator reports that it has recognized a 0 or a 1.

So let's just review some of the key points of the system.

(1) The correlator sort of takes a majority vote on the chips, so it tolerates a certain amount of errors. Even so, it will often make a mistake, so some additional error correction is usually needed.

(2) The P and N codes have to be different enough that there isn't a likelihood of mistaking one for the other. Even their parts have to be different, and this means that only certain P and N codes can be used. Still, there are many possible combinations.

(3) A receiver can only decode a transmitted signal if it knows what P and N codes the transmitter used. Other transmitters using different P and N codes, even though operating on the same frequencies, appear as just noise. While they increase the overall noise in the system, they don't really interfere with reception unless they are very near. Thus many transmitters can use the same frequencies without interference to each other, and they are often difficult to detect.

The FCC has recently opened up three bands (902–928 MHz, 2400–2483.5 MHz, and 5725–5850 MHz) for unlicensed spread-spectrum operation. A variety of equipment, such as wireless headphones, cordless phones, burglar or fire alarms, and wireless modems, are already being marketed. There are even some integrated circuits which contain most of what you'd need for a simple spread-spectrum transceiver. Moreover, direct sequence spread-spectrum (under the name CDMA or Code Division Multiple Access) is being developed for cellular telephones as a means of allowing many more telephones to be used in a given area without interfering with each other. This is a big field, and getting bigger.

Digital Signal Processing

Another interesting new concept which may drastically change radio systems in the very near future is digital signal processing.

A DSP (Digital Signal Processor) is essentially a specialized microcomputer IC, dedicated to processing analog signals. It takes an analog signal, converts it into digital numbers, processes the digital data in some way, and then converts it back into an analog signal.

Until now, DSP circuits have been used in equipment primarily to process the audio. For example, a DSP can analyze the received audio to identify constant signals (such as the whistles produced by interfering stations) and remove them. It can do the opposite too — identify those signals which represent voice signals, and amplify them more than other signals. In this way, the DSP has been used primarily to remove interference and noise.

As another example, several companies now manufacture noise-reduction headphones. Mounted on the headphone is a small microphone which picks up outside noise. A DSP circuit analyzes that noise, and then sends an equal, but opposite signal to the headphone. This opposite signal partially cancels out the outside noise, reducing the noise level. The concept has also been used to counteract machinery noise.

Until recently, DSP's have been plagued by slow speed, with the result that they could barely keep up with audio frequencies. But recent advances in DSP technology have speeded them up to the point where they are becoming fast enough to work at IF frequencies in receivers. This opens up an entire new area for them.

A number of manufacturers are working on receiver designs that consist of just three analog parts — an RF amplifier, oscillator, and mixer. The IF output from the mixer would then immediately be digitized by a DSP, and all other functions, including IF amplification and filtering, and detecting would be done digitally.

The idea is actually driven by cellular telephones. A cellular phone site normally needs a number of receivers, all tuned to different frequencies. Using DSP's would eliminate all this, replacing it with one RF amplifier, one oscillator, and one mixer. The resulting IF signal (which would actually contain a number of different received signals at the same time) would then be processed by several different DSPs, each one recovering the signal from one mobile or handheld cell phone user. When a cell phone switches from one frequency to another, instead of the cellular site having to switch receivers, the DSP would simply be reprogrammed to recover a different signal.

The whole concept is still brand new, and in its infancy. But we can expect the typical radio receiver five or ten years from now to be very different from today's superhet!

Duplex operation

The word *duplex* implies simultaneous transmission in two directions. For example, when you use a cordless or cellular phone, you want to be able to hear the other party even while you speak. The phone must therefore be able to transmit and receive at the same time.

Prior to the use of digital techniques, this was a major design problem. The transmitted signal in the phone is so much stronger than the signal received from the base station (which is much farther away), that it overloads the receiver and reduces its sensitivity. This always required very careful design to keep the transmitted signal out of the receiver.

Digital phones completely avoid the problem by taking turns receiving and transmitting. When a voice signal is converted into digital data, the data can be temporarily stored in a memory circuit and then transmitted in a short burst of high speed data. At the other end, these short bursts are slowed down and connected together with other data into a continuous stream of information. With careful timing, the base station and the phone handset take turns transmitting and receiving, so that it seems to you as though you are receiving and transmitting at the same time, but you are really not. In fact, some of the cellular telephone providers use a technique where several cellular phones take turns transmitting on the same frequency. In this way, frequencies are reused to allow more cell phones to operate in a given area.

Conclusion

As you can see, interesting and exciting things are happening in radio today. The advent of spread-spectrum communications is turning the industry around. Forty years ago, when CB or Citizens' Band radio started, so many people started using it that interference (and bad operating) made the band a shambles. Now spread-spectrum makes it possible for large numbers of people to share the same spectrum space without really interfering with each other.

Potentially, this is a tremendous improvement, but it can also lead to some problems. Just like the military developed spread-spectrum over the years to provide security from enemy detection, so today’s criminals can use these same techniques to avoid detection and capture. There have been numerous government proposals to require equipment manufacturers and communications carriers to design their systems so as to allow federal law enforcement organizations to intercept their signals. We shall see what happens in that area in the future.