Design Notes

A new wide band SDR with many applications

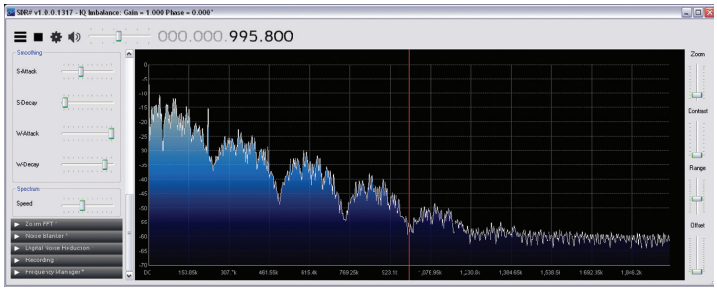

PHOTO 1: The SDR-Play module used with the *SDR#* software as a spectrum analyser. Here is it showing the spectrum of the spread spectrum source of Figure 1, with a chip rate of 250kHz.

SDR-PLAY MODULE. The folks at SDR-Play [1] have come up with a new SDR receiver module based around a chipset that functions in a similar manner to that of the RTL dongle, but offers a wider tuning range and increased sampling resolution. Using a Mirics MSi3101 chip, the module covers the band 100kHz to 2GHz, with a small gap around the 380–410MHz region. It has a 10 bit A/D converter sampling at up to 10MHz and this sample rate means that up to 8MHz of instantaneous bandwidth is available for spectrum display or receiving / decoding wide band modes. The 10 bit A/D resolution means a higher dynamic range can be achieved than the RTL dongles offer. It delivers I/Q samples via USB to the host PC running off the shelf SDR software. As supplied, the module uses the *SDR#* (SDR-Sharp) software – the same as the RTL dongles. An installation disc is supplied with the hardware that includes an interface driver so when *SDR#* is subsequently started, the SDR-Play module is offered as a source option.

The designers have supplied all the necessary drivers on the CD-ROM, but please note that you need to install the drivers first before plugging in the module. Software 'hooks' and interfacing data is also supplied on the CD-ROM, so it should be straightforward for software authors to incorporate this wide band receiver module into their own SDR or test equipment software.

IN USE. The *SDR#* software should be familiar to users of the RTL dongle and once the SDR-Play module is installed, the result will look similar to that, except for the much wider frequency band on offer. With a lower tuning limit of 100kHz, all the LF through to UHF amateur bands are covered. Of these, HF reception is usually the most taxing due to the wide dynamic range of signals within a few MHz anywhere in the HF spectrum. On my active whip antenna it happily received SSB and data signals on the HF bands – although some juggling of the module gain setting (accessible via the configuration menu) was needed when moving from band to band. Stability was sufficient to be able to wrap audio round for demodulation of PSK31 and WSPR signals.

The installation disc also comes with *Mirics* software for DAB and wideband FM broadcast radio. Having never before listened to DAB side by side with the same station transmitted on Band 2 FM, by switching between the two I could definitely hear the subtle differences in the digital broadcast audio quality.

With up to 8MHz bandwidth, the 10 bit sampling means that around 50 to 60dB of dynamic range is available as a spectrum analyser. **Photo 1** shows a screenshot of the SDR-Play module displaying the partially filtered spread spectrum signal of the generator described below.

SPREAD SPECTRUM GENERATOR. In the Data column last month, I described an outline spread spectrum (S/S) scheme that could be used as a starting point for experiments in the new 146 to 147MHz allocation that we will hopefully be able to use before too long. With 1MHz of bandwidth available for 'high speed digital type signals' a 500kHz or 250kHz chip rate spread spectrum waveform seems to fit the bill. The circuit of

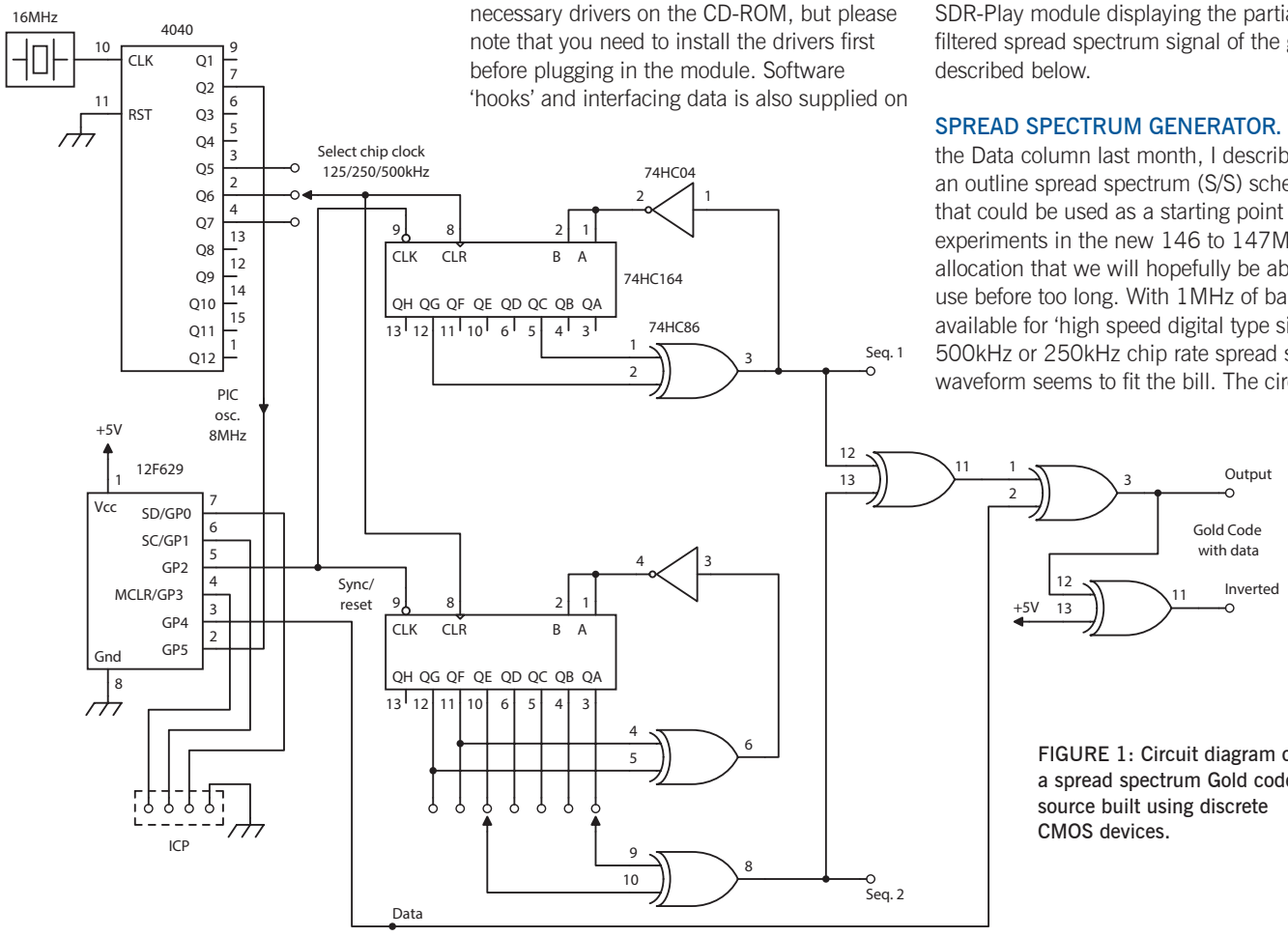

FIGURE 1: Circuit diagram of a spread spectrum Gold code source built using discrete CMOS devices.

Figure 1 is a complete code generator with a source of user data, built using old fashioned CMOS devices to illustrate the basics of the coding.

The spread spectrum source generates a set of 21 different selectable 127 chip Gold codes using a technique similar to that of the GPS satellites. (For more details of the rationale behind this choice, see last month's Data column). Two 74HC164 eight bit shift registers with exclusive-or gates produce the two starting maximum length sequences. The simplistic form of Maximal Length (ML) sequence generator shown last month, with feedback from the XOR gates feed directly back to the shift register input, is the normal way of generating ML sequences. But this suffers from the disadvantage that the all zeros state is not permitted, so to initialise the system, some other value has to be preset. If the seven shift register stages were to contain all zeros, the XOR gate would output a '0' output, which would be fed back and the sequence would lock up. By adding an inverter after the XOR, the all zeros state is now permitted: the all '1' state is now the illegal condition. The only difference having the inverter in place makes is that the generated sequence is the complement of its original. The all-zeros state is a very convenient place to start the two sequences in synchronism as the 74HC164 chips have a reset pin making preload at the start straightforward and meaning the correct Gold code starts properly. The chip clock comes from a 16MHz TTL oscillator module divided down in a binary divider chip.

To complete the S/S source, a PIC microcontroller is included to generate user data that modulates the final spread spectrum code. The PIC is clocked by the same source as that driving the shift registers, so by including a divide by 127 routine inside the PIC code and resetting the shift registers in synchronism with this at switch on, the binary modulation can be applied synchronously to the fast sequence, inverting it at sub-multiples of the 127 bit chip repeat rate in the same way as the GPS satellites transmit their user data.

The use of old discrete logic chips may seem a bit archaic nowadays when a fast processor or gate array would do the job just as well. But they are still readily available, cheap, and are probably lying unwanted in junk boxes; and it's really not that complicated to build up! More details of the source, including PIC code can be found at **[2]**. My breadboard, built rats-nest style, is shown in **Photo 2**.

A 16F family PIC controller running with a 20MHz crystal is capable of generating the S/S signal at a chip rate of up to about 125kHz for slower systems; the faster 18F family running at 40MHz will allow up to 250kHz chip rate. In a software implementation, the easiest – and fastest – way to generate the code is simply to store the complete sequence pattern in memory. 127 bits only takes up 32 bytes, so even the basic 16F PIC devices can hold all 21 sequences in a table, selected by external switches. A version using a field programmable gate array (FPGA) will, of course, allow very fast operation to tens or even hundreds of MHz and could incorporate the controller / data source as well all in one chip. If you have the design tools for such devices, that is!

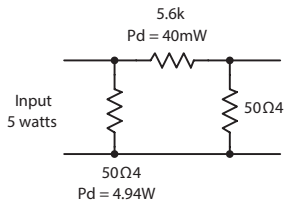

FIGURE 2: Theoretical values of a 47dB 5W attenuator, with power dissipations, designed without taking practicalities of construction into account.

OUTPUT FILTERING. Before using the generated S/S sequence for modulating an RF source, it needs to be low-pass filtered to remove the components at harmonics of the chip rate that would lead to out of band emissions. Filters for such digital signals have to preserve the pulse shape, and special filter types such as Bessel, Linear phase or Gaussian are needed. The spectrum shown in Photo 1 is that of the waveform having passed though a 5th order linear phase filter with a cutoff of 500kHz. I haven't included any details of this filter as the one tested wasn't very good in terms of pulse shape at 500kHz chip rate, so a better design is needed. Reference [2] gives more details and shows the pulse with some distortion after passing through that filter.

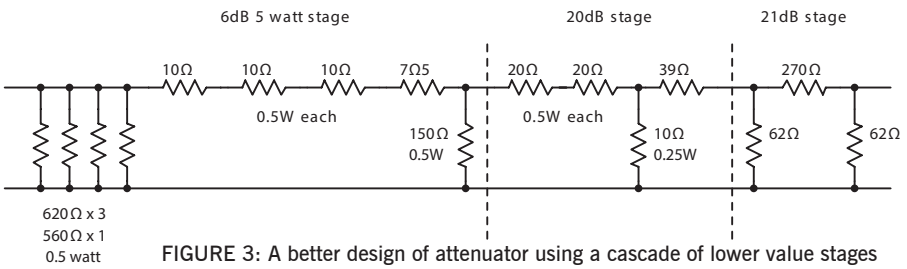

FIGURE 3: A better design of attenuator using a cascade of lower value stages designed to apportion power dissipation between several resistors and reduce the effects of stray capacitance.

BUILDING MATCHED ATTENUATORS.

Discussions on an internet group recently suggest there is still some confusion over how to construct an attenuator for attenuating the output of a transmitter to drive the low level input of a transverter. One contributor wanted to attenuate his 10MHz 5W output to 0.1mW (-10dBm) and therefore needed a 47dB attenuator rated at 5 watts dissipation. There are numerous online tools that will calculate values for π or T attenuators, (try typing 'attenuator design' into you favourite search engine), but they don't give the full story. For example, the contributor stated that he had a design needing one 5.6kΩ and two 47Ω resistors, as shown in **Figure 2**. There are several reasons why a simple three resistor attenuator is not a good choice for such a high value of attenuation. First, the single input resistor has to dissipate nearly all the input power, so just one resistor has to have a high power rating. Second, and most significantly, the 5.6kΩ resistor is quite a high value at this frequency and it will be difficult to avoid stray capacitance across it from changing or degrading the wanted attenuator performance if all the loss has to generated in, basically, one component

A better solution is to cascade several lower value attenuator stages to make up a total of 47dB. This way, lower values of series resistors are used, with much reduced effect from strays. Also, the input power can be shared out over several resistors, or several attenuator stages.

First, use an online tool that gives the dissipation in each resistor – such as my spreadsheet in **[3]**. Note in particular that when an attenuation of 6dB is chosen, the dissipations in the first two resistors are equal, each dissipating one third of the input power. The same applies if π or T designs are chosen. So a suitable cascade could consist of, say, a 6dB first stage, designed to dissipate the bulk of the power, 20dB in the second stage, with only the input resistor dissipating much power and topped of with a third stage of 21dB.

An elegant touch is to cascade alternate π and T stages for a ladder design like the complete 5W 47dB three stage attenuator of **Figure 3**. By using multiple resistors to make up the input high dissipation arms, no more than 0.5W rated resistors are needed. Series or parallel connection is used as appropriate to minimise the effect of stray capacitance or residual inductance in each resistor.

WEBSEARCH

[1] SDR-Play module: www.sdrplay.com
[2] Spread spectrum source material: www.g4jnt.com/SSstuff.zip
[3] Attenuator design spreadsheet: www.g4jnt.com/DownLoad/ATTEN.XLS