CLICK HERE for circuit diagram

The "Fast" type Huff & Puff stabiliser architecture was developed by Peter Lawton G7IXH. Please refer to his November 1998 QEX article on the subject for the theoretical background (don't forget the High Res version of his circuit diagram). The "fast" design incorporates a shift-register delay line and compares the current latched state of a crystal reference oscillator against its state Z periods previously. These two states are XOR'ed together, integrated, which produces the feedback signal to control and stabilise the VFO.

This style of stabiliser provides a kind of statistical averaging of the up/down pulses from the XOR gate, compared to the standard Huff Puff stabiliser method. The correction pulses arrive Z times more frequently and therefore the accumulated error in the VFO before it is corrected, is much less. The stabilisation loop operates much more quickly. The "fast" stabiliser has been likened to a whole collection of Z operating simulataneously on the same VFO. With a "fast" Huff & Puff stabiliser, it is possible to stabilise a worse (or much higher frequency) VFO; or alternatively stabilise a comparable VFO with much less frequency ripple.

The formula for step size is:

Step = 10^6 x VFO^2 / (z x M x xtal)

where VFO is the VFO frequency in MHz,
z is the number of stages of delay,
xtal is the crystal reference frequency in MHz, and
M = 2^n where n is the number of divide-by-2 stages in the VFO divider.

For my minimalist simple 2-chip version of this architecture, I applied several simplifications relative to the original "fast" design. The more stages of delay are involved, the better the performance of the stabiliser. Therefore I would have liked to use the 4517 128-bit shift register. However, this IC was never manufactured in a high speed 74HC-family variant, so the only alternative would have been the original 4000-series CMOS. This would have required a relatively low frequency crystal reference oscillator in order to not violate the setup and hold requirements at the serial input of the shift register. In previous "fast" designs there is a high speed D-type flip flop in front of the 4517, which effectively removes any significant speed requirement from the 4517. But I didn't want to add an extra IC (this IS a minimalist design after all) so I chose instead to use an 8-bit shift register type 74HC164.

The first stage of the shift register is used in place of the D-type flip flop (74HC74) of previous designs, and holds the "current" state of the crystal reference oscillator. The final stage of the shift register holds the state of the crystal reference oscillator 7 periods ago. In my design the number of delay stages is therefore 7. Which isn't massive, but this IS a minimalist design as I have to keep reminding you, and anyway 7 stages is still a significant performance improvement

over the "classic" Huff Puff architecture.

In a "fast" design, unlike the "classic", the higher the crystal reference frequency the better. This is because it enables us to use less stages of VFO division, which means that the number of correction pulses per second can be increased (for the same step size) which maintains tighter control over the VFO. Building high frequency crystal oscillators has always been problematic for me and over 16MHz I go to the canned oscillators. This also saves an extra IC, though one could argue that the canned oscillator most probably contains an IC.

I used the internal oscillator of the 74HC4060 as the VFO. If one wished to use an external VFO, it would be fed into pin 11 of the 74HC4060 which would then function merely as a divider. I used a standard 5mm red LED as varicap as usual, which provides both VFO tuning and the means for the Huff & Puff correction to be applied. With the components shown, the frequency range was 9,903 to 9,954 kHz. The capacitance of the varicap is small relative to the 470pF capacitors in the LC-circuit. If they were reduced the tuning range could be increased significantly. I didn't bother, since I was only trying to prove the design rather than use it in an actual project (yet).

Finally, instead of using a 74HC86 quad XOR-gate IC, I built a XOR gate from a transistor, five diodes and three resistors. The XOR gate doesn't operate at high speed so any old NPN transisor will be Ok. Remember that the output of an XOR gate is "1" if either of the inputs are "1", but is "0" if both are "1" (or if both are "0"). It is important that the ratio of R4 to R5 + R6 is relatively high. Otherwise the output voltage level signifying one kind of "0" (both inputs "1") will not be the same as the other kind of "0" (both inputs "0") due to voltage from the integrator capacitor coming back through R4.

I actually built this circuit without too much attention, and it worked first time! I didn't have the 74HC

The picture shows my prototype, which is built "dead-bug" over an unetched single-sided-PCB groundplane.

An article "Een Minimalistische Bijsloffer" derived from this project and written in Dutch by Ron Brink PA2RF, was published in the May 2007 issue of "Electron".