```
//  Gives control to the PF2 and PF3 pins to the SPI if needed and restores
// the I/O page register.
af = ar and 0x4000; if ne rti (db); io(SPIFLG0) = sr0; iopg = ax0;
//  Sends the new word on the SPI connection.
rti (db); dm(AudioDelayLineOutput) = i3; nop;
```

5 Implemented demodulator structure

The actual demodulators presented in this application note (see figure 9) are the theoretical ones, thus providing much better results than the classical analogical equivalents, if enough precision is used in all computations. As the purpose of this paper is to prove the possibility of using the ADSP-2199x mixed signals DSPs for implementing digital radio receivers, all computations are kept as simple as possible just to be easier to understand.

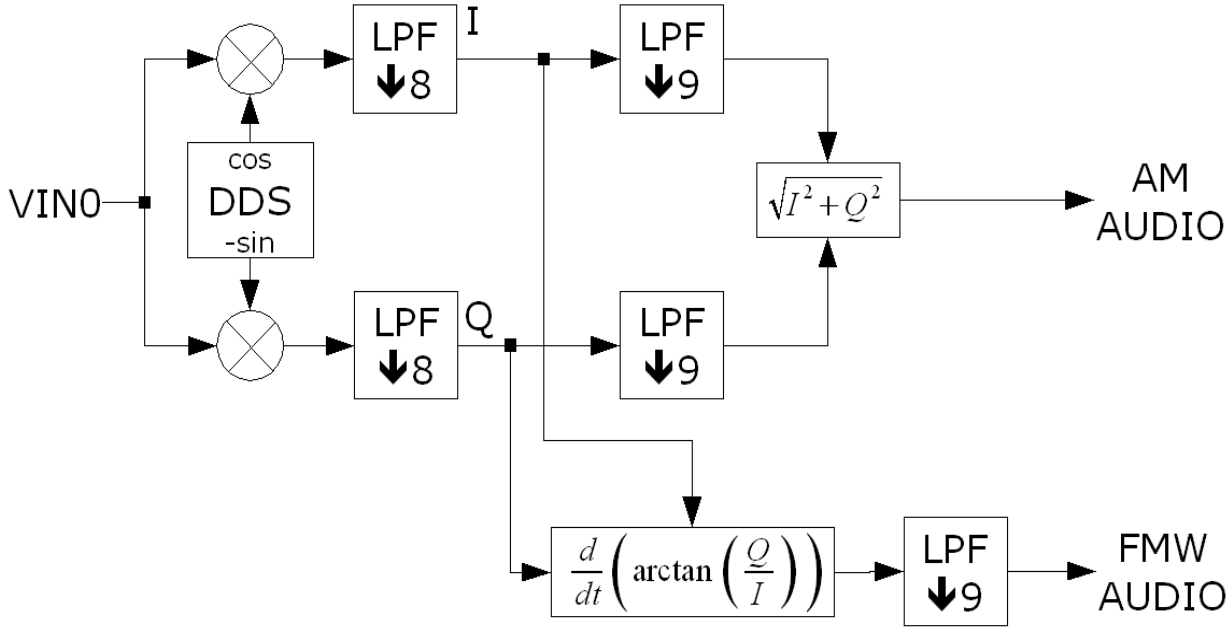

Figure 9 General demodulator structure

As understanding the DSP source code requires good knowledge of assembly code and signal processing and most blocks implemented in the code are very well described in other papers, only the things that are unusual will be presented here, as they are the key to the demodulators implementation in a time effective manner.

5.1 Quadrature DDS and I and Q mixers

Classically, the quadrature oscillator would have a variable holding the sine/cosine phase and would call the sine and cosine computing routines for each input sample. While for low sampling rates this would provide the best quality, the limited number of cycles for each input sample, just 64 in this case, requires a solution that avoids calling twice the 34 cycles sine computing routine.

Of course, the first idea is to use a software replica of a DDS, with a table with sine values large enough to insure the required quality, but also small enough to fit the limited amount of on-chip memory. The upper bits of the sine/cosine phase would simply provide the index to access the table in order to obtain the desired value. But even this technique is too slow for this situation, with over 10 cycles for each input sample.

After refining the implementation of the above, a solution that needs only 7 cycles for each input sample has been established, solution that also implements the two mixers. Apart from the look-up table with sine samples and phase increment values, also two pointers for accessing the sine and cosine values in the same table are used. By having