

DIGITAL SIGNAL PROCESSING TECHNIQUES FOR RADIO AMATEURS

=====

1. Introduction

Up to about 15 years ago, the design, assembly, testing and field maintenance of electronic circuits followed almost the same guidelines. During the design phase, the function of the circuit was split into smaller units, each corresponding to already tested circuits or available components. In production, after assembly the circuit had to be aligned, tested and eventually repaired by highly skilled test engineers. Highly skilled personnel was also required for field maintenance to spot and replace defective components. Due to the continuously decreasing prices and increasing complexity of available components, mainly integrated circuits, the complexity of the overall circuit became no longer limited by the cost of the components or assembly, but with the time and cost required to design, align, test and maintain a circuit. For instance, repairing a printed circuit board containing 100 TTL SSI function (gates, FFs, counters) ICs when the malfunction only appears at particular environment conditions (higher temperature or only from time to time) is an almost impossible task. Spotting the source of a problem in an analog circuit can be even more difficult and requires expensive test equipment (oscilloscope, spectrum analyzer...) beside a skilled engineer.

Easy to use 8 bit microprocessors solved many of the problems of digital circuit design, testing and maintenance, beside a further reduction of the cost of the hardware. The design time was shortened by introducing complex but reliable microprocessor and peripheral integrated circuits while test and/or debugging routines could be implemented at little if any additional cost in the final product. Although microprocessor technology created some new problems, like software design and maintenance, the overall effect was a great simplification over hardwired logic designs, either hardwired SSI TTL or CMOS logic or custom ICs. In fact, microprocessors replaced hardwired logic in all but the highest speed digital circuits.

Of course, continuous efforts were made to simplify the design and testing of the more difficult analog circuits. For example, a very successful innovation was the introduction of the operational amplifier: the design engineer could finally concentrate on the differential equations describing his problem instead of thinking how to bias his transistors... Some analog problems could be solved more easily if electrical signals were replaced by other physical quantities. For instance, a long delay line is easier to make with mechanical waves and Surface Acoustic wave (SAW) filters can now be found in almost all IF strips of domestic TV sets.

The basic idea of Digital Signal Processing (DSP) is to replace an electrical circuit with its mathematical equivalent, solving the the equations describing the circuit numerically in

real time. The latter can be done either with hardwired logic or with computers. Of course DSP circuits do not require any tuning or alignment, since the tolerances of the components are only limited by the accuracy of the numerical models used. Production line testing and maintenance is limited to check the operation of a microprocessor.

Beside cost advantages, DSP circuits allow the designer to use components that could hardly be implemented with analog electronic components: tuned circuits with arbitrary, even infinite but stable values of Q , easily variable component values during circuit operation (adaptive processing) or very complex algorithms that would require a very large number of high accuracy analog components (Fast Fourier Transform).

The main drawback of DSP is a limited bandwidth of all the signals present in the circuit: all the computations usually have to be performed at a rate of at least twice the signal bandwidth. DSP circuits are therefore limited to IF, audio and video applications. DSP circuits also require very fast logic or powerful microprocessors to provide usable results. This is the reason why DSP has only become popular with advanced 16 and 32 bit microprocessors and corresponding peripherals.

DSP has much to offer to radio-amateurs as well, especially since little if any alignment is necessary. DSP circuits will certainly replace expensive crystal filters within a few years. Some "forgotten" techniques like the phasing method for the generation of a SSB signal will again become popular due to the accuracy (and the ease with which it can be obtained) of DSP circuits. A microprocessor based DSP hardware can be programmed for different circuits. For instance, the same hardware can generate all possible modems for all known amateur modulation standards, including new experiments. DSP techniques can also improve weak signal communications: narrow-band CW filters can be built which ring much less than their analog counterparts and weak signal detection and/or demodulation circuits can be built with performances beyond that of a human ear. Finally, some communication modes could become more popular thanks to DSP, especially image communications.

2. DSP circuit examples

2.1. Principles of DSP circuits

A DSP circuit can generally be split into an interface part including A/D and D/A converters and related circuits to interface to the analog "environment" and a digital part performing some numerical operations on the digitalized analog signals (see Fig. 1). Of course, a DSP circuit may also include digital inputs or outputs, either when the signals are already available in digital format or when a conversion between an analog and a digital format is performed by the DSP circuit

(DSP modems).

The input analog signal is first band limited essentially to prevent interferences called aliasing: a finite number of samples taken by the following sample and hold stage can only represent a limited bandwidth signal. The sample and hold circuit and following A/D converter are triggered periodically at regular intervals called the sampling period or its inverse, the sampling frequency. The latter should be at least twice the signal bandwidth: practical applications require a sampling frequency 2.5 to 3 times the signal bandwidth due to circuit imperfections.

The accuracy of the A/D and D/A conversions affects the available dynamic range of the DSP circuit: if the digital signal format is binary (as usual), then every additional bit increases the dynamic range by 6 dB. Finally, the output of the D/A converter again requires some analog filtering to obtain the desired signal without spurious frequencies or distortions.

The following discussion will concentrate on the digital part of a DSP circuit and in particular on the various algorithms used to generate different circuit functions. The analog circuit designer has available a number of different components to generate the functions desired. The former could be grouped in three groups: linear amplifiers and attenuators to adjust signal levels, energy storing components like capacitors, inductors, delay lines and resonators to generate frequency dependent networks and finally nonlinear components like rectifiers or multipliers (balanced mixers).

The DSP circuit designer has to replace all these components with mathematical algorithms that will be computed on each input signal sample coming from the A/D converter and will provide a regular stream of output signal samples to the D/A converter. Which algorithms are actually available to the DSP designer? Gains and attenuations can simply be performed by multiplying signal samples with a constant. Of course, a separate multiplication has to be performed on each signal sample. Although division could also be used, the hardware required for division is usually more complex (or, in other words, a microprocessor usually needs more time to perform a division than a multiplication operation) and a multiplication with the inverse value is usually used.

Frequency dependent networks are usually built using delay lines. The delays are usually selected to be equal to or to be integer multiples of the signal sampling period: such delays can simply be implemented by using one or more previous signal samples in the computations on the actual signal sample. Circuits using delay lines and feedback can efficiently simulate capacitors, inductors and resonators.

Nonlinear components are represented by nonlinear functions. For instance, full wave rectification can simply be obtained by inverting the polarity of the signal subtracting the value from zero only when the sign is found negative. Multipliers or balanced mixers can be straightforwardly replaced by a multiplication operation. More complex functions, like the square root, trigonometric or other transcendent functions are very time consuming to be computed in real time on every

signal sample. Therefore, function tables are prepared in advance and stored in memory. During real time processing, the operation is limited to retrieve the precomputed value from the lookup table memory (either ROM or RAM).

2.2. Simple linear circuits

Engineers usually describe the operation of linear analog circuits using linear differential equations (this is the reason why such equations were invented for...). The total number of independent energy storing components, like independent springs and masses in a mechanical problem or independent capacitors and inductors in an electrical circuit determines the order (and the complexity) of the resulting differential equation.

A DSP circuit will be a good replacement for an analog circuit if its operation can be described by a similar equation. Since DSP circuits work on an uniform stream of signal samples and not on continuous signals, there is no way to compute derivatives nor to obtain a differential equation. One can however compute differences between successive samples: if the sampling period is sufficiently short these can be considered a good approximation for derivatives. DSP circuits are therefore described with finite difference equations. Although the latter can be arranged in a form similar to the differential equations describing analog circuits to show the similarities, in practice it is usually simpler to solve the difference equation immediately!

A simple RC low-pass and its DSP equivalent are shown on Fig. 2. The RC low-pass is described by a first order differential equation since it contains a single capacitor. Its DSP equivalent includes a delay element and a feedback network. Rearranging the difference equation and considering that the difference between two successive signal samples divided by the time interval T between the two samples can be a good estimate for the first derivative, the difference equation becomes very similar to the differential equation of the analog circuit, except for a gain factor.

A LC tuned resonant circuit contains two energy storing components and is described by a second order differential equation. Its DSP equivalent (shown on Fig. 3) must contain two independent delay elements. Again, the difference equation describing the DSP circuit can be rearranged to become similar to the differential equation of the analog resonator: three successive signal samples are required to compute the approximation for the second derivative.

The analog LC circuit shown and its equation correspond to the ideal case of a lossless resonator. It is well known that such a resonator can not be built in practice, since there are always some loss mechanisms. On the other hand, a lossless resonator (with infinite Q) can readily be built as a (stable!) DSP circuit. The loss (or gain) factor depends on the coefficient that multiplies the output of the second delay element: (-1) corresponds to a lossless resonator. Since the

multiplication with (-1) only involves a sign change, it is always accurate: there is no truncation of the result. The output of such a circuit is a constant amplitude sine-wave, depending only on the initial conditions - energy stored in the two delay elements when the circuit was started.

2.3. Infinite Impulse Response (IIR) or recursive filters

The DSP lowpass and resonator described above are just two members of a more general group of DSP circuits called Infinite Impulse Response (IIR) or recursive filters. These circuits always contain feedback networks and their response to a single input pulse always includes a form of exponential decay. In practice they are used as a DSP replacement for analog filters using discrete components like capacitors, inductors or resonators (quartz crystals).

Fig. 4 shows a simple recursive filter corresponding to a single LC damped resonant circuit. Compared to the lossless resonator, the output of the second delay element is multiplied by a negative constant (-k2) whose absolute value is less than one. The input signal is simply added to the feedback signals and fed into the delay elements. The overall transfer function is similar, except for a gain factor, to that of a loaded LC circuit.

More complex IIR filters can be obtained by a series connection of "single tuned circuit" filters or designing a filter with more delay elements. Compared to analog filters the series connection is simpler since there is no influence from the next filter back to the previous one: no separation amplifiers are required and the overall transfer function is simply a product of all the transfer functions of the single filters.

2.4. Finite Impulse Response (FIR) filters

Finite Impulse Response filters also include delay elements and multiplications with coefficients, but they do not include any feedback loops, as shown on Fig. 5. Their impulse response is therefore limited in time to the sum of all the delay elements used. FIR filters are also built in analog technologies using tapped delay lines. The most popular analog implementation are Surface Acoustic Wave (SAW) filters.

The coefficients of a FIR filter have a straightforward influence on the impulse response of the filter, if one imagines a single pulse propagating along the delay line. FIR filters require more stages and more computations for the same frequency response when compared to IIR filters. On the other hand, their time domain response and group delay can be easily controlled. The differential group delay of a FIR filter becomes zero if symmetrical coefficients are selected: for example k1 equals k5 and k2 equals k4 in the filter on Fig. 5.

FIR filters are therefore very useful where differential

group delay is a problem, like image or data transmissions, either to avoid distorting the signal or to correct distortions already present in the signal. SAW filters have therefore a definite advantage over conventional LC filters in television IF strips. DSP FIR filters will probably be used by radio-amateurs for CW reception, since they ring much less than discrete component filters, which necessarily have an infinite impulse response.

A major application area of DSP is adaptive signal filtering to eliminate distortions: the actual signal distortion is monitored continuously and the data obtained is used to update the coefficients of a distortion correcting FIR filter.

2.5. The discrete Fourier transform and the FFT algorithm

The Fourier transform is an algorithm that computes the spectrum of a signal from its waveform. It has the form of an integral with infinite bounds and the result is a function which is also defined over the interval from minus to plus infinity. Such a mathematical problem is very difficult to solve numerically for an arbitrary input function. The inverse Fourier transform, used to obtain back the original signal, is mathematically almost identical to the Fourier transform itself.

A limited bandwidth signal can be sampled without losing any information. On a limited number N of successive samples an approximate algorithm can be computed called the discrete Fourier transform. The discrete Fourier transform is equivalent to a batch of FIR filters each tuned to its own frequency. From N signal samples N different frequency components can be computed using N FIR filters each having N stages. Since the phases of the various spectral components are not known, all the computations are assumed to be done with complex numbers! The inverse operation, obtaining N signal samples from N frequencies, is very similar to the discrete Fourier transform too and requires the same number of mathematical operations.

The discrete Fourier transform is still a very time consuming computational task, since N^2 complex multiplications and N^2 complex additions have to be performed. The main idea of the Fast Fourier Transform (FFT) algorithm is to change the order of multiplications and additions to reduce the overall number of operations. FFT works on numbers of samples N that are integer powers of 2, $N=2^M$, the number of operations required can be reduced to only $N \cdot M$!

The FFT algorithm has many applications, not limited to DSP and signal processing at all. The most obvious is a spectrum analyzer, unfortunately only for audio bandwidths or slightly above using available microprocessor technology. The possibility to obtain the inverse transform in a similar way makes FFT based filters and other circuits practical.

2.6. Nonlinear functions

Nonlinear functions can be computed either directly or, when the algorithm is too complex to be computed in real time on each sample, by using precomputed function tables. The following example shows the solution of an important practical problem using a precomputed function table.

The most widely used nonlinear function is certainly the fullwave rectifier, usually as an AM demodulator, which can be simply implemented. However, this is not the only way to build an AM demodulator. A major drawback of a full wave rectifier is that it requires to be followed by a low pass filter to eliminate any rests of the carrier frequency. The low pass filter becomes quite complex if the carrier frequency is comparable to the modulation signal bandwidth.

A typical example is the demodulation of weather satellite APT image transmissions. The signal obtained from a VHF FM receiver is an amplitude modulated 2400 Hz sub-carrier with the video signal, which has a bandwidth in excess of 1600 Hz. Any rests of the carrier frequency will cause severe interference patterns on the image when the demodulated signal is sampled to obtain discrete pixels if the pixel sampling frequency is not an exact sub-multiple of the carrier frequency.

An alternative, high performance AM demodulator, that does not require any post-demodulation low pass filtering, is shown on Fig. 6. The amplitude of a sine-wave can be computed from two successive samples, but this operation requires 4 multiplications, two additions and one square root to be computed for each sample. If the sampling frequency is set to 4 times the carrier frequency, the formula becomes much simpler: only two multiplications, one addition and a square root are required. This is still not very practical, since the algorithm to compute the square root is time consuming: it requires a minimum of 5 division operations and as many additions and shifts.

A further speedup can be obtained by considering the actual data formats. The logarithmic A/D converter supplies samples coded to 8 bits: one sign bit, 3 exponent bits and 4 mantissa bits. The sign bit can be discarded immediately, since only the square of the signal is used in computing the function. The desired result is a function of two 7 bit samples, totally 14 bits. These 14 bits are used to address a function table of the size of $2^{14}=16384$ memory locations containing all possible function results. The result is computed by a single access to the lookup ROM!

Function tables are therefore very practical as long as their size is affordable. In the above case, the size of the lookup table is multiplied by four for every additional bit supplied by the A/D converter! On the other hand, the direct computation of transcendent functions can be very time consuming, usually more than 10 multiplications and as many additions. Even some of the most powerful dedicated DSP microprocessors already have an on-chip lookup table ROM for the sine function!

2.7. Fixed frequency oscillators and VCOs

Like their analog equivalents, DSP circuits need oscillators for carrier generation, frequency mixing or signal demodulation. Although models of analog oscillators could also be built using DSP techniques, there are a number of simpler algorithms to generate oscillators in DSP. One was already mentioned: a lossless resonator circuit. Building oscillators can be even simpler if one considers the actual data formats.

For example, the circuit shown on Fig. 7 would generate a continuous slope if the addition were an ideal mathematical operation. However, at a certain time instant, overflow is reached. In the case of integer arithmetics, the value suddenly jumps from the most positive to the most negative number. The circuit generates a sawtooth waveform which is a slowly increasing slope from N_{min} to N_{max} and then a sudden jump back to N_{min} . The amplitude of the oscillator is constant and equal to the maximum dynamic range that can be represented by a given data format.

The oscillator frequency is defined with the ratio of the number added each time compared to the available number range and of course multiplied with the sampling frequency. The number added each time can be a variable or the output of another DSP circuit. In this way a VCO can be built to form a phase locked loop for example.

A sine-wave output can be obtained from a sine function table addressed by the original sawtooth. Triangular and square waves can be obtained by simple operations on the original sawtooth waveform. Harmonic frequencies, sometimes required in PLL carrier recovery circuits, can be obtained by a simple multiplication (with overflow) of the sawtooth signal: due to the higher amplitude, overflows will occur more frequently.

2.8. A practical example: an 1200 bps PSK demodulator

Replacing a simple RC low pass with an expensive microprocessor is probably not worth the effort. DSP circuits become useful when more functions are performed by the same DSP hardware, like a complete demodulator or modem. The following example shows how different DSP algorithms can be combined into a useful circuit.

JAS1, renamed FUJI-OSCAR-12 after its successful launch in august 1986, is a radio-amateur satellite receiving in the 145 MHz band and transmitting in the 435 MHz band. Beside a linear, analog transponder it carries a digital transponder connected to the on-board computer. The latter is usually programmed to work as a multi-user mailbox to support store-and-forward type communications between radio-amateurs. The down-link in the 435 MHz band is a 1200 bps PSK transmission and its bandwidth corresponds to that of available SSB receivers. PSK modulation was chosen since it efficiently uses the satellite transmitter output power and can be efficiently demodulated using coherent demodulators.

The signal at the output of a SSB receiver is suitable to be processed by a DSP circuit. In the actual circuit, the receiver audio output is sampled 9600 times per second with a 8 bit logarithmic A/D converter (telephone CODEC). Using a lookup table the samples are converted to a 16 bit linear format and sent to the circuit shown on Fig. 8.

The signal path is straightforward: the incoming signal samples are multiplied by the locally regenerated carrier. This multiplication generates 32 bit products. This result is truncated so that only the most significant upper 16 bits are used for further processing. A two stage recursive low pass filter follows to eliminate any residual carrier signals. The output of the circuit is a NRZI data stream, which is simply limited and sent to the digital circuits for clock and data recovery (not shown on Fig. 8).

The carrier recovery PLL is slightly more complex. The original PSK transmission does not contain any discrete spectral components at or around the carrier frequency. However, if the square of the signal is computed, a discrete spectral line appears at twice the carrier frequency. Squaring would require an elaborate AGC circuit to keep the PLL loop gain constant, so it was replaced with a simple search for zero crossing transitions in the original PSK signal: this rough approximation generates a discrete spectral line at twice the clock frequency as well.

The transition detector triggers a sampling phase detector operating on a double frequency sawtooth signal coming from the VCO. In fact, the frequency of the sawtooth is only doubled after the sampling detector to reduce the average number of mathematical operations. After adding a phase correction constant the PLL error signal is passed through the loop filter. Since the PLL itself introduces a 90 degrees phase shift in the loop, the low pass loop filter has to be designed to introduce an even smaller phase shift to avoid reaching instability at 180 degrees. The familiar resistor-capacitor network used in analog PLLs can be easily reproduced in DSP too: part of the feedback goes through (-k3) directly to the VCO (resistor divider) while another part goes through a recursive filter (RC low pass).

The sawtooth generated by the VCO is transformed to a sine-wave using a lookup table. Since a 16 bit accuracy is not required, only the most significant 8 bits are used to address the lookup table. The choice of the carrier frequency is arbitrary in theory. In practice, interferences with the bit rate frequency and passband limitations of SSB receivers limit the carrier frequency range to between 1500 and 1800 Hz for a 1200 bps PSK signal.

The circuit shown on Fig. 8 was practically built in the form of a machine code program for a MC68010 microprocessor (shown on Fig. 9) and tested in real time on live satellite signals. Since the execution of this machine code routine only takes 31 microseconds or less for each signal sample while samples are taken every 104 microseconds, the same microprocessor was also used for all data handling from the A/D converter to data demodulation and display, including an "all software" AX.25 controller (a detailed description is

omitted for simplicity).

Practical results have shown that there is a considerable margin on the satellite signal, received with a 10 turn helix antenna and masthead preamp. Reception of error free AX.25 frames was still possible when the signal was more than 10 dB below normal (due to local obstructions), noise was audible and contact with the satellite was already lost due to insufficient up-link performance. With such a high performance demodulator and a higher power transmitter (around 200w) operation with the JAS1 mailbox using omni-directional antennas should be possible.

3. Hardware for DSP

3.1. A/D and D/A converters

Although not related directly to the DSP theory, A/D and D/A converters are important parts of a practical DSP system and may sometimes limit its performance. DSP systems require relatively fast A/D converters with sampling frequencies between 10 kHz and several tens MHz. D/A converters can more easily meet the specifications, since a simple R-2R network D/A can work at video frequencies and above.

Basically, two different types of A/D converters can be used: the successive approximation A/D converter and the "flash" A/D converter. Successive approximation A/D converters compute the digital result in steps, each additional bit taking one step. During all this time the input analog voltage must remain stable: successive approximation A/Ds require a high performance sample-and-hold circuit for correct operation. Due to the internal step-by-step algorithm and the requirement for a high performance sample-and-hold their operation is not particularly fast, up to about 100 kHz sampling frequency. They can however provide accuracies of up to 16 bits and beyond. Telephone CODECs are 12 bit successive approximation A/D converters, the format is however internally converted to 8 bits logarithmic (floating point).

"Flash" A/D converters include internally a large number of voltage comparators, one for each quantization step. Their outputs are connected to a priority encoder which supplies the digital result. "Flash" A/Ds usually don't need any sample-and-hold circuitry since the signal is sampled in an internal latch between the comparators and the priority encoder. Their operation is very fast: cheap types for video applications work up to 20 MHz, others can reach several hundred MHz. Their accuracy is however limited with the number of comparators that can be integrated in a single circuit to about 8 bits (which require 256 comparators, as many latches and a 256 input priority encoder).

D/A converters are usually driven to supply a constant output voltage during the time interval to the next sample. The output of a D/A converter should therefore be a step function, but due to the finite speed of its internal circuit

the output needs some time to settle to the new value. During this settling time spikes may appear on the output due to the internal switching processes. These spikes may cause severe distortions, especially to video signals. The parameter that describes the magnitude of these spikes is called the glitch energy and should of course be minimized if possible.

The step function supplied by the D/A is not the best solution: ideally the D/A should provide very narrow pulses sent to an ideal low pass filter. The step function generates a low pass filter with a frequency function of the form $\text{SIN}(X)/X$. The D/A converter therefore requires at its output both an analog low pass filter to eliminate harmonics and an analog high pass to correct for the $\text{SIN}(X)/X$ spectrum dependence.

3.2. Data formats

When designing a practical DSP system, a suitable digital data format has to be selected. The accuracy of the data format will limit the dynamic range of the system. Errors caused by data format inaccuracies are called quantization noise. When selecting a data format it is not sufficient to find the corresponding A/D converter. The dynamic range may substantially increase in some DSP algorithms. For example, a recursive resonator filter may increase the signal magnitude by its Q factor, a FIR filter may increase the magnitude by the absolute sum of all its coefficients (usually many) and the FFT algorithm may increase the magnitude by a factor equal to the number of signal samples on which the FFT was computed.

A simple and widely used data format are binary integer numbers. The dynamic range of a binary integer is roughly equal to the number of bits multiplied by 6 dB (smaller deviations from this simple rule are caused by different methods of measuring the quantization noise). Thus, a 8 bit binary integer will allow a dynamic range of roughly 48 dB and a 16 bit binary integer format will allow a dynamic range of roughly 96 dB.

Binary integers can be signed or unsigned. Unsigned integers are always considered positive. A signed integer is considered negative if its most significant bit is a logical one. For example, the value of the binary number 1111 is considered 15 (fifteen) if it is an unsigned integer, but it is a -1 (minus one) if it is a signed integer. Microprocessors usually support both integer formats: addition and subtraction instructions are identical for both integer formats, there are only differences in the multiply and divide instructions.

The selection between signed and unsigned integers depends on the actual problem to be solved with the DSP circuit. Signed integers are suitable for signals that can have both polarities, like audio signals in general. Unsigned integers are suitable for single polarity signals, like video signals or image processing in general.

When the absolute magnitude of a signal is not known, the floating point data format (also called real number format) has to be used. The floating point format includes a sign bit, a binary exponent and a mantissa. The exponent itself is a

signed integer while the mantissa is an unsigned integer. Floating point mathematical operations require more hardware and more time to be executed than integer operations, therefore they are not very popular in DSP.

A special case of the floating point format is the telephone CODEC format: 1 sign bit, 3 exponent bits (chord bits) and 4 mantissa bits (step bits), for a total of only 8 bits. A/D and D/A converters for this particular format are available under the name CODEC and are inexpensive since they are widely used in telephone exchanges all over the world. Unfortunately there are two slightly different standards: the American μ -law and the European A-law. Although no mathematical operation can be performed directly on the 8 CODEC bits, the format is very suitable for lookup table processing due to the very small size of lookup tables! Even some of the most powerful dedicated DSP microprocessors have ROM conversion tables for the CODEC formats.

3.3. Hardwired logic DSP circuits

The first real-time DSP circuits were implemented in hardwired TTL logic, since no suitable computers were available. Due to the high complexity and cost, DSP circuits were only used in places where there were no other alternatives. Algorithms were very simple due to the high cost of multipliers. The main function of DSP circuits was usually analog data storage for time expansion or contraction functions, like scan converters between different imaging systems.

Even today high speed video DSP systems require hardwired logic, although integrated on a single chip. The new MAC television standard will probably use such decoders in domestic TV receivers. Another consumer video application is an adaptive FIR filter for automatic ghost elimination.

Hardwired logic DSP circuits are both complex and expensive. Once built they can not be reprogrammed for another function and modifications are difficult. While custom integrated circuits may reduce costs in volume industrial applications, these are out of reach of amateur experimenters. General purpose or dedicated DSP microprocessors are therefore used wherever possible in place of hardwired logic designs.

3.4. General purpose microprocessors

General purpose microprocessors can be used for DSP, but due to speed limitations their use is restricted to audio and lower frequencies. For example, a voice audio signal is usually sampled at 8 kHz, or every 125 microseconds. A typical 8 bit microprocessor takes about 200 microseconds to compute a single 16 bit by 16 bit product yielding a 32 bit result, using ADD and SHIFT instructions in a short machine code program. Therefore, a 8 bit microprocessor can not compute

even the simplest DSP filter in real time. Some simple lookup table algorithms are however possible.

16 bit microprocessors offer an increase in the computing speed of about two orders of magnitude. The latter is not due much to the increased bus speed but rather to the much more powerful instruction set, so that much less instructions are required for the same operation. A larger number of internal registers allows to decrease the number of data transfers on the bus even further. Multiply and divide operations are implemented as single instructions. A 16 by 16 bit multiply only takes about 4 microseconds in a modern 16 bit microprocessor, like the the Intel 80286, NEC V30 or Motorola MC68010.

These components allow many useful DSP circuits to be built in software. If a sampling frequency of around 8 kHz is used (voice bandwidth audio signal), filters of up to about 10th to 15th order can be built. This is sufficient for a complete FSK modem for RTTY, ASCII or PACKET up to 1200 bps and including all post demodulation processing (UART or X.25 controller) and data display to be implemented in software on a single 16 bit microcomputer. A real-time 256 point FFT is also within range of 16 bit microprocessors.

Many actually available 32 bit microprocessors are merely 32 bit bus copies of their 16 bit counterparts and are not significantly faster. Their processing power is limited by the same problems as 16 bit microprocessors: pipeline instruction decoding and relatively long execution times for some instructions.

Pipeline instruction decoding means that an instruction is not executed immediately after it has been fetched from the program memory but it is decoded sequentially and finally executed two or three instruction fetch cycles later. This process does not slow down the execution as long as the program does not contain jumps. After a program jump the content of the pipeline decoder is no longer usable and the execution only resumes after two or three instruction fetch cycles. This process slows down the execution of short program loops considerably.

16 bit microprocessors require a large number of clock cycles to execute complex instructions, like multiple shifts, multiply or divide instructions. The hardware used to execute the latter is usually a state machine with a microprogram ROM. Since hardware shifters, multipliers and dividers are at least an order of magnitude faster than a state machine, considerable improvements in the computing speeds can be expected in future microprocessors.

Floating point co-processors are usually of little use in DSP: their high computational accuracy (up to 64 bit mantissa) is actually not required in DSP while low accuracy (16 bit) integer arithmetics is not faster than that of the master (general purpose) CPU.

3.5. Dedicated DSP microprocessors

As soon as microprocessor based DSP became technically possible, the first dedicated DSP microprocessors appeared. Dedicated DSP microprocessors are optimized according to the specific requests of DSP: very high speed arithmetic operations but (relatively) little memory both for data and program. High computation speeds are achieved both by using hardware adders, shifters and multipliers and by using separate buses for instructions and data.

Since separate buses require a very large number of connections, the memory is usually integrated on the same chip together with the microprocessor and only a few connections are taken out through the pins of the IC. The on chip program memory is either ROM or RAM or both, since accessing external memory usually slows down the microprocessor.

One of the first and now most popular DSP microprocessors is the Texas Instruments TMS32010. Thanks to an on chip hardware multiplier it is able to compute a 16 by 16 bit product in just 200 nanoseconds, more than an order of magnitude faster than general purpose 16 bit microprocessors. State of art dedicated DSP microprocessors, both from the TMS320xx family and from other manufacturers are able to compute a product in less than 100 ns.

A disadvantage of dedicated DSP microprocessors is certainly their complex internal structure making them more difficult to program. Compared to general purpose microprocessors they have a very small addressing range (for example: TMS32010 only 4 kilo-words) and this slows down the microprocessor when using large lookup tables or longer programs or when processing large amounts of data like images. Finally, a dedicated DSP microprocessor typically needs the support of a general purpose microprocessor to work efficiently.

4. Radio-amateur applications of DSP

4.1. Demodulators and modems

The first amateur applications of DSP techniques are certainly different demodulators and modems. These circuits are usually connected between an amateur receiver or transceiver and a home computer to receive and transmit in RTTY, ASCII, PACKET and other digital communications formats. Although amateur modems are usually not very complex, usually just a few operational amplifiers or other audio frequency components, the construction is made difficult due to the large number of transmission standards used.

For instance, FSK RTTY transmissions can have shifts of 170 Hz, 425 Hz or 850 Hz. 170 Hz shift transmissions can use low tones (1275 / 1445 Hz) or high tones (2125 / 2295 Hz). The transmission speed can be 45, 50, 75, 100 or 110 bps. Each of these combinations requires its own filters for best results since the bandwidth of the filters must match the

transmission speed used. While changing the center frequency and the bandwidth of an analog filter at the same time is difficult at best, changing all of the parameters of a DSP filter is restricted to changing a few constants in the program!

Since amateurs usually need only a single modem at the same time, only one DSP processor is required. Changing from one standard to another is performed by simply changing the software running on the DSP processor. New standards, like satellite PSK transmissions, can be implemented quickly writing new software and most important of all at little if any additional cost!

Modems working on voice bandwidth audio signals, as provided or required by standard amateur transceivers, do not require very powerful DSP microprocessors. They can usually be implemented on general purpose 16 bit microprocessors. The same microprocessor can be used for all data decoding, handling and display, since the latter require even less computing power than the DSP algorithm. A prototype RTTY, ASCII or PACKET terminal was built around a single 16 bit microprocessor (MC68010) including a DSP modem. Compared with analog modems built with operational amplifiers, the DSP modem performed much better in heavy QRM or poor signal to noise conditions thanks to the optimized filters for each transmission format.

4.2. All mode receivers and transceivers

State-of-art commercial amateur transceivers use microprocessors only for the front panel control and display functions. The signal processing part of the transceiver is a complex analog circuit including several conversions and intermediate frequencies and a complex multiple loop PLL frequency synthesizer. Some of the analog components are very expensive, like narrow-band crystal filters. During production, the transceiver requires expensive test equipment and skilled technicians for alignment, further increasing the cost of the final product.

A DSP alternative for an "all mode, general coverage" HF receiver is shown on Fig. 10. The front end is similar to analog receivers: a low pass filter followed by a mixer and a first, wide-band IF crystal filter around 45 MHz. After an AGC amplifier to optimize the level the signal is sampled. The sample and hold circuit works as a harmonic mixer, translating the 15 kHz wide signal at 45 MHz down to frequencies acceptable for a dedicated DSP microprocessor. A 16 bit (or even more accurate) A/D converter has to be used to obtain the maximum possible dynamic range.

The DSP microprocessor generates the required filters and demodulates the signal. A 8 bit D/A provides a sufficient dynamic range to drive the loudspeaker. Digital transmissions, like RTTY, can be demodulated directly and supplied in a digital format through a RS-232 port. The DSP processor also controls the AGC amplifier to optimize the useful dynamic

range. Considering the amount of computations to be performed a dedicated DSP microprocessor has to be used. Practical receivers will probably use single chip DSP microprocessors with an internal program ROM storing all the algorithms to receive all the modes (SSB, CW, RTTY...) with optimum bandwidth filters.

A command and control microprocessor, similar to that used in "analog" transceivers, is also required in this DSP oriented receiver. Another interesting detail is that a DSP based receiver only requires a 10 kHz step frequency synthesizer, which can be a single loop PLL. "Analog" receivers require very fine step (10 or 20 Hz) synthesizers with multiple loop PLLs whose phase noise sidebands usually limit the receiver dynamic range.

Finally, a DSP based "all mode, all features" HF receiver does only require little if any alignment! A companion transmitter can be built in a similar way. Considering all the above facts, the introduction of DSP technology means that the size and complexity of a HF receiver can be reduced to that of a VHF handy-talkie FM transceiver bringing at least a five fold reduction of the manufacturing costs! For a complete transceiver the simplification is slightly less spectacular since the transmitter output stage and corresponding power supply remain expensive and a two to three fold reduction of prices can be expected.

The technology required to build DSP based transceivers is available NOW. Since it brings substantial advantages all the manufacturers will be forced either to accept it soon or to disappear from the market.

4.3. Weak signal communications

Up to now weak signal communications required a skilled operator with well trained ears, the final steps in the processing and decoding of weak CW or SSB signals being left to the human brain. Computer aided communications were not used due to the poor performance of available modems and other problems, like interferences generated by the computer itself.

Improvements when receiving weak signals hidden in noise can only be obtained by reducing the data rate: the significant quantity is the energy radiated by the transmitter per each bit of information. The main limitation of the human ear (and related part of human brain) is that the data rate can not be reduced indefinitely: below a certain speaking speed the probability of understanding a word decreases and below a certain keying speed CW characters become difficult to decode.

Radio operators therefore use redundancy: they repeat the same message (either CW or SSB) several times to be understood by their correspondent. Repeating is again limited by the memory available in the human brain...

DSP computers allow, at least in theory, almost unlimited filtering in front of the detector and averaging of the detected signal. The limit is of course defined by the

usable data rate! The most interesting amateur application is certainly low power EME (moon-bounce) communication using (relatively) small antennas. DSP computers will certainly be advantageous at data rates below 1 bit per second. Some DSP experimenters already reported very enthusiastic results, like detecting their own signal echo using a single Yagi antenna and a solid state TX. Maybe EME communications will no longer be a privilege for those having very large antenna systems, but one may still argue that the very low data rates that may result, like 1 bit per hour! can hardly be useful: DSP computers are not able to do miracles!

4.4. Image communications

Amateur image communications have never become very popular, probably due to the problems with transmission standards. Live image transmissions (ATV) have a very high information rate and thus a limited range compared to other communications modes. SSTV, FAX and other slow scan standards require each its own, incompatible hardware.

DSP microprocessors will essentially simplify the hardware required for slow scan standards: just one microprocessor equipped with different software will be sufficient for all present and future transmission standards. Of course, a TV monitor will be used as a display driven by the computer video board. The next step will be simple image processing: zooming, filtering, sharpening and contrast enhancement.

Digital image communications require even more bandwidth (or transmission time) but offer a much better image quality. Researchers all over the world are working on image data compression algorithms and hardware, which also requires high speed DSP microprocessors. Hopefully these new techniques should some day allow a live image transmission at a limited data rate, maybe as low as 64 kbps.

5. A prototype DSP computer

5.1. Hardware configuration

In the beginning of 1987 I started working on an image processing computer designed especially to demodulate, process and display weather satellite APT pictures. At the beginning the project was intended as a state-of-art replacement for now obsolete but famous "APT Scan Converter" developed more than six years ago. The design requirements were a high resolution video, a large memory to store images and a fast bus to update the video as quickly as possible.

According to previous experiences, satellite APT pictures require at least 64 gray levels. 256 gray levels corresponding to 8 bits or one memory location per pixel were selected for convenience. To remain compatible with standard TV monitors a

picture format of 256 useful lines out of 320 was selected with 512 pixels per line. The video board thus contains 128 kilobytes of special dual port memory to avoid slowing down the microprocessor.

It was immediately clear that a 16 bit microprocessor was required already to move the data to the video board. After consulting various manufacturers catalogs and distributors price lists I decided to use a Motorola 680xx series microprocessor and in particular the MC68010. The MC68010 has a 16 Mbyte linear, non-segmented addressing space and 16 general purpose 32 bit internal registers. Other available 16 bit microprocessors had the addressing space divided into 64 kbyte segments which makes it very difficult to handle large amounts of data like images or to write programs longer than 64 kbytes.

At first, a hardware demodulator like that in the APT scan converter was planned, except for using switched capacitor filters in place of op amps. After studying the problem it was found out that the MC68010 was not just able to generate a much higher performance DSP signal demodulator but it could automatically adjust the signal level, reliably extract sync pulses, process the demodulated video signal and display it on the TV monitor as well.

Once decided to do as much processing as possible in DSP with the microcomputer, only a band pass filter was placed in front of the A/D converter. A telephone CODEC IC (MK5156) was selected as the A/D converter since its conversion speed and accuracy correspond to the bandwidth and dynamic range of the signals available at the output of a voice communications receiver (including the APT signal). The MK5156 already includes a sample and hold circuit and an almost independent D/A converter.

The block diagram of the prototype DSP computer is shown on Fig. 11. The computer is built on several printed circuit boards connected together with a mother board with 64 pole "euro-card" connectors. The computer actually includes the following printed circuit board modules:

- A) Processor board including the MC68010, 32 kbytes of EPROM (operating system), 64 kbytes of CMOS RAM (stack and operating system), a keyboard interface and a real time calendar clock.
- B) Video board including 128 kbytes of dual port dynamic video RAM, video D/A converter and all the timing circuitry.
- C) CMOS RAM boards carrying 256 kbytes of battery backed RAM each. Usually four such boards are installed for a total 1 Mbyte of nonvolatile RAM.
- D) Analog I/O board carrying the CODEC A/D and D/A converter, a programmable timing generator and a RS-232 port.
- E) Floppy controller board including a WD2797 floppy controller, a high speed dual serial port (Z8530 SCC) and corresponding support circuitry.
- F) Bus motherboard carrying 8 "euro-card" connectors.
- G) Switching power regulator board including a NiCd RAM backup battery and a very reliable RESET signal generator to protect the nonvolatile RAM content regardless of the actual power-up or power-down timing.
- H) Rotator interface board designed to control a KR5600 type

rotator through the RS-232 port.

Up to date 4 prototypes have been built and tested at different CPU clock frequencies. All the software developed requires a minimum clock frequency of about 9 MHz while a reliable operation of the hardware can be obtained at clock frequencies up to 13 MHz.

5.2. Software developed and tested

The basic DSP routines are usually not very complex, up to 100 instructions using simple integer mathematics, but they are executed several thousand times per second. To use the available computer efficiently programming directly in machine code (or assembly language) is required. This is especially important if a general purpose microprocessor has to be used close to its theoretical speed limit. On the other hand, DSP support routines, like those computing the coefficients of the basic DSP routines or retrieving the final results, are not executed so frequently but may be quite complex and require floating point mathematics and transcendent functions.

In order to fulfill both the above requirements, a high level language compiler that allows the insertion of arbitrary size machine code routines was written. The structure of the high level language is similar to FORTRAN, allowing both single variables and multiple dimension arrays, floating point mathematics, transcendent functions and conditional jumps or calls to labels. Most important of all, it allows a simple and efficient communication (data transfer) between the high level language program and the machine code routines!

Since the economic advantages of using dynamic RAMs are declining, I decided to use CMOS static RAMs in my computer. The content of static RAMs can be easily made non volatile with a small NiCd backup battery. This means that it is no longer necessary to base the operating system on floppy discs or other magnetic storage devices: all the software can remain in its place in the nonvolatile computer RAM where it is actually executed. The floppy disc drive is only used for memory backup.

The software developed includes the operating system written directly in machine code and stored on an EPROM and various application programs written in high level language and stored either in their original form or compiled in the nonvolatile RAM or on floppy discs. The operating system includes a machine code monitor program (hardware debugging, data movements including the floppy disc), a screen text editor program and a high-level language compiler. The operating system allows a simple interrupt driven multi-task provided that the application programs use different peripherals with different interrupt vectors.

Up to date, the following application programs were developed and tested, not including various test and/or support programs and the software actually under development.

A) Universal BAUDOT / ASCII receiving program. Allows to set the tones between 1000 and 2400 Hz, the speed between 45 and

1200 bps and all data standard formats. The received text is displayed on the TV screen and can be stored, on command, in the computer memory and handled later by other programs. Includes a tuning indicator. Tested both in heavy HF QRM and on satellite signals it always performed equal or better than optimized analog modems built with operational amplifier active filters.

B) FSK PACKET RADIO (1200 bps, BELL-202) RX/TX program. Supports standard VHF terrestrial packet radio communications. Includes a terminal program. The received text can be stored, on command, in the computer memory for later use. Text files, prepared by other programs, can also be transmitted.

C) Satellite tracking program. Computes the position of a selected satellite from its Keplerian Elements in real time once per second (using the real time clock) and supplies the result to the antenna rotator interface through the RS-232 port. It includes editing routines for the 40 satellite data sets stored. Different tracking procedures can be selected and during the tracking all the relevant parameters are displayed on the computer screen.

D) Satellite APT images receiving program. Designed especially to receive APT images from polar orbiting weather satellites. Includes a menu of 12 user defined picture formats including line rate and sync pulse data, pixel and line sampling ratios and display parameters. The display parameters include independent horizontal and vertical zooming in very fine steps and a sophisticated gray scale enhancement function that does not saturate any part of the image. All the display parameters can be controlled interactively, without disturbing image reception or other tasks running on the computer.

E) PSK PACKET RADIO (1200 bps) program. Includes a PSK modem with a tuning indicator, but otherwise it is very similar to the FSK packet radio program. Specially designed to communicate with FUJI-OSCAR-12 it can be used for terrestrial PSK communications as well.

The satellite tracking program can be combined with any of the other four programs in a simple interrupt driven multi-task so that only one computer is required to track a satellite and process the received data at the same time.

6. Conclusion

Digital Signal Processing will certainly bring many changes in the design and construction of electronic circuits, at least in the audio frequency range, where many problems can be solved efficiently using microprocessors. Engineers now have a whole group of new components to be considered in their new designs. Of course it was not possible to describe all possible applications of DSP techniques in this article: most applications of DSP probably have yet to be discovered! Similarly, some important DSP algorithms were merely mentioned to exist, like the Fast Fourier Transform.

DSP should be interesting for amateurs not just because it solves some problems but since it offers a much wider space

for experiments than analog circuits can do. Beside improving existing communications modes it is hoped that DSP will make new communications modes possible, like QRP EME operation or live picture transmission over narrow-band channels.

Finally, it is demonstrated that a general purpose 16 bit microprocessor can generate many practical DSP circuits provided that it is not slowed down by inefficient hardware nor by a stupid operating system. Expecting that there is sufficient interest a more detailed description of my prototype DSP computer is planned to follow.

#####

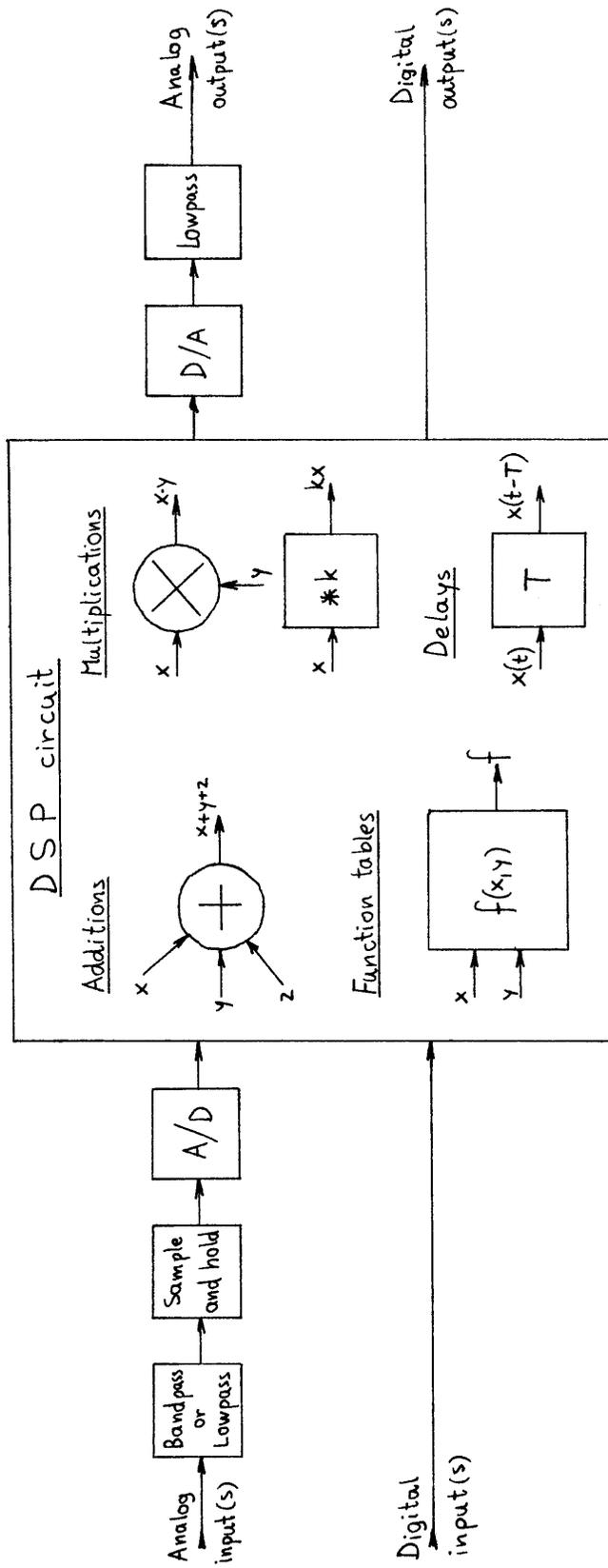
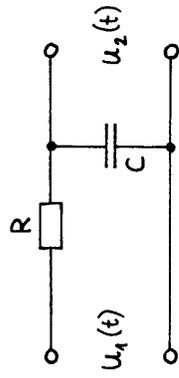


Fig. 1. - DSP circuit principles.

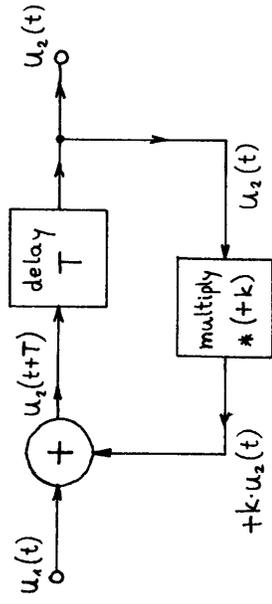
Simple (analog) low pass:



Corresponding differential equation:

$$\frac{d(u_2(t))}{dt} + \frac{u_2(t)}{RC} = \frac{u_1(t)}{RC}$$

Equivalent DSP circuit:



Finite difference equation:

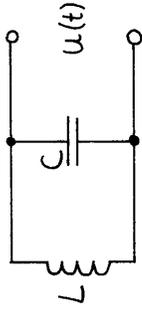
$$u_2(t+T) = u_1(t) + k \cdot u_2(t)$$

or:

$$\frac{u_2(t+T) - u_2(t)}{T} + \frac{(1-k)}{T} \cdot u_2(t) = \frac{u_1(t)}{T}$$

Fig. 2 - A simple analog low pass and its DSP equivalent.

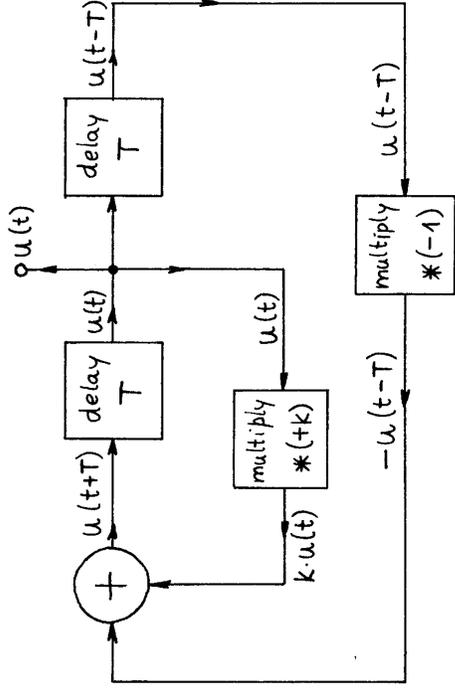
LC tuned circuit:



Corresponding differential equation:

$$\frac{d^2(u(t))}{(dt)^2} + \frac{u(t)}{LC} = 0$$

Equivalent DSP circuit:

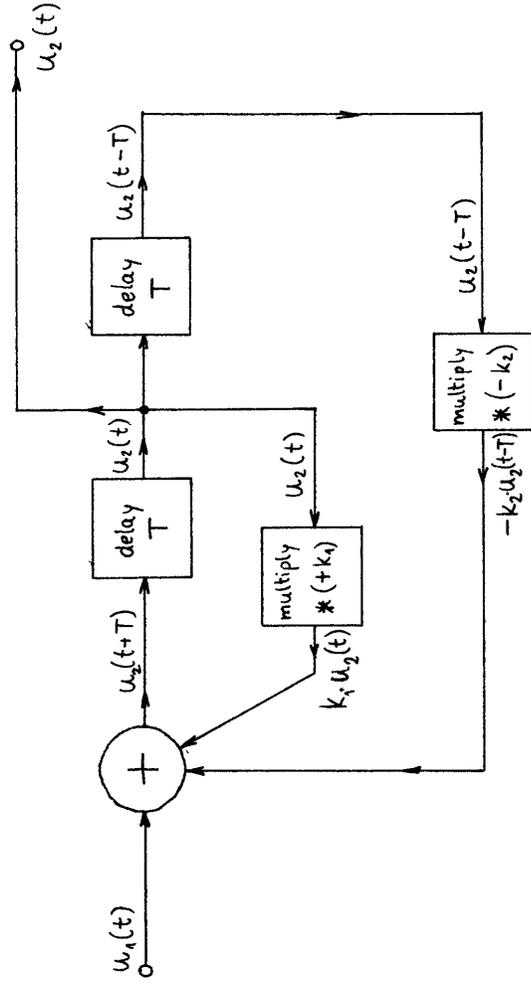


Finite difference equation: $u(t+T) = k \cdot u(t) - u(t-T)$

or:

$$\frac{u(t+T) - u(t)}{T} - \frac{u(t) - u(t-T)}{T} + \frac{(2-k)}{T^2} u(t) = 0$$

Fig: 3 - An LC tuned circuit and its DSP equivalent.



Equation: $u_2(t+T) = u_1(t) + k_1 u_2(t) - k_2 u_2(t-T)$

Fig. 4 - A single "tuned circuit" band pass : Infinite Impulse Response (IIR or recursive) filter.

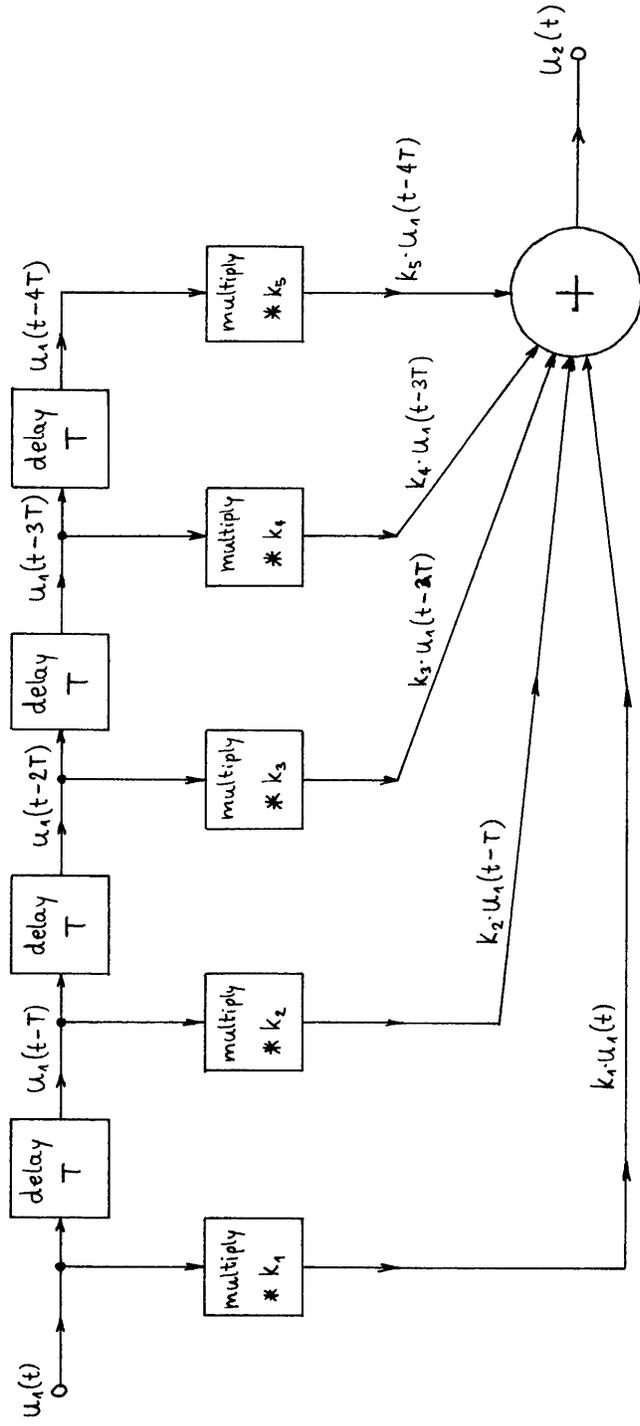
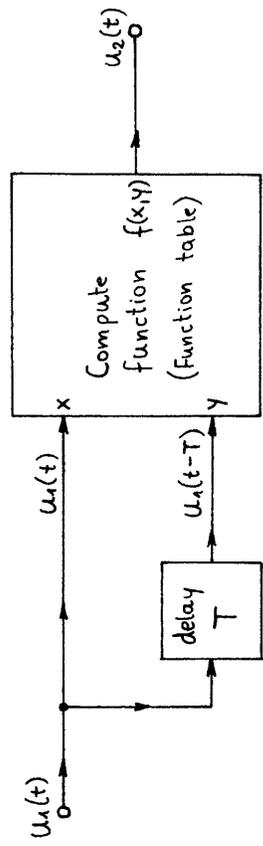


Fig. 5 - A 5 stage Finite Impulse Response (FIR or nonrecursive) filter.



$$f(x, y) = k_1 \sqrt{x^2 + y^2} - k_2 xy$$

$$k_1 = \frac{1}{\sin \omega T} \quad ; \quad k_2 = 2 \cos \omega T$$

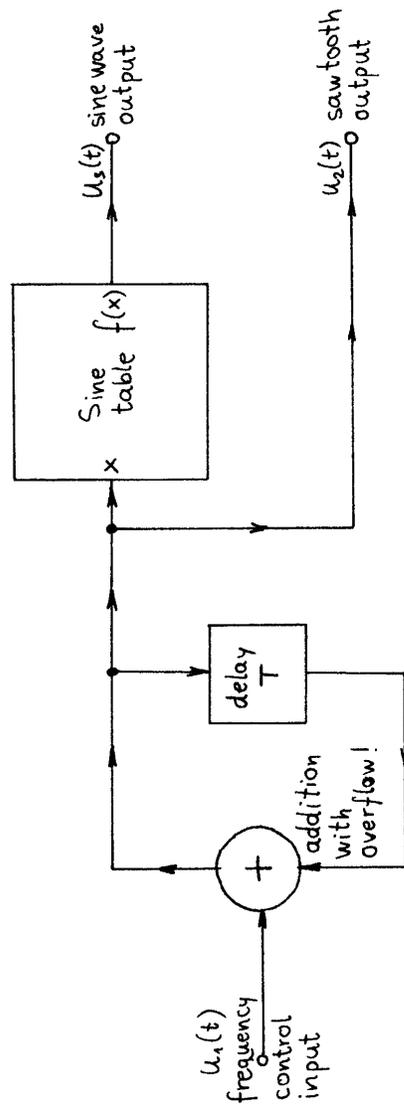
If: $u_1(t) = A \cdot \cos(\omega t + \varphi)$

then: $u_2(t) = f(x, y) = A$

Special case: $\omega T = \frac{\pi}{2}$ (90° degrees)

$$f(x, y) = \sqrt{x^2 + y^2}$$

Fig. 6 - A high performance AM demodulator.



$$f = \frac{1}{T} \cdot \frac{u_1(t)}{(N_{\max} - N_{\min})}$$

Fig. 7 - A sawtooth / sine wave voltage controlled oscillator.

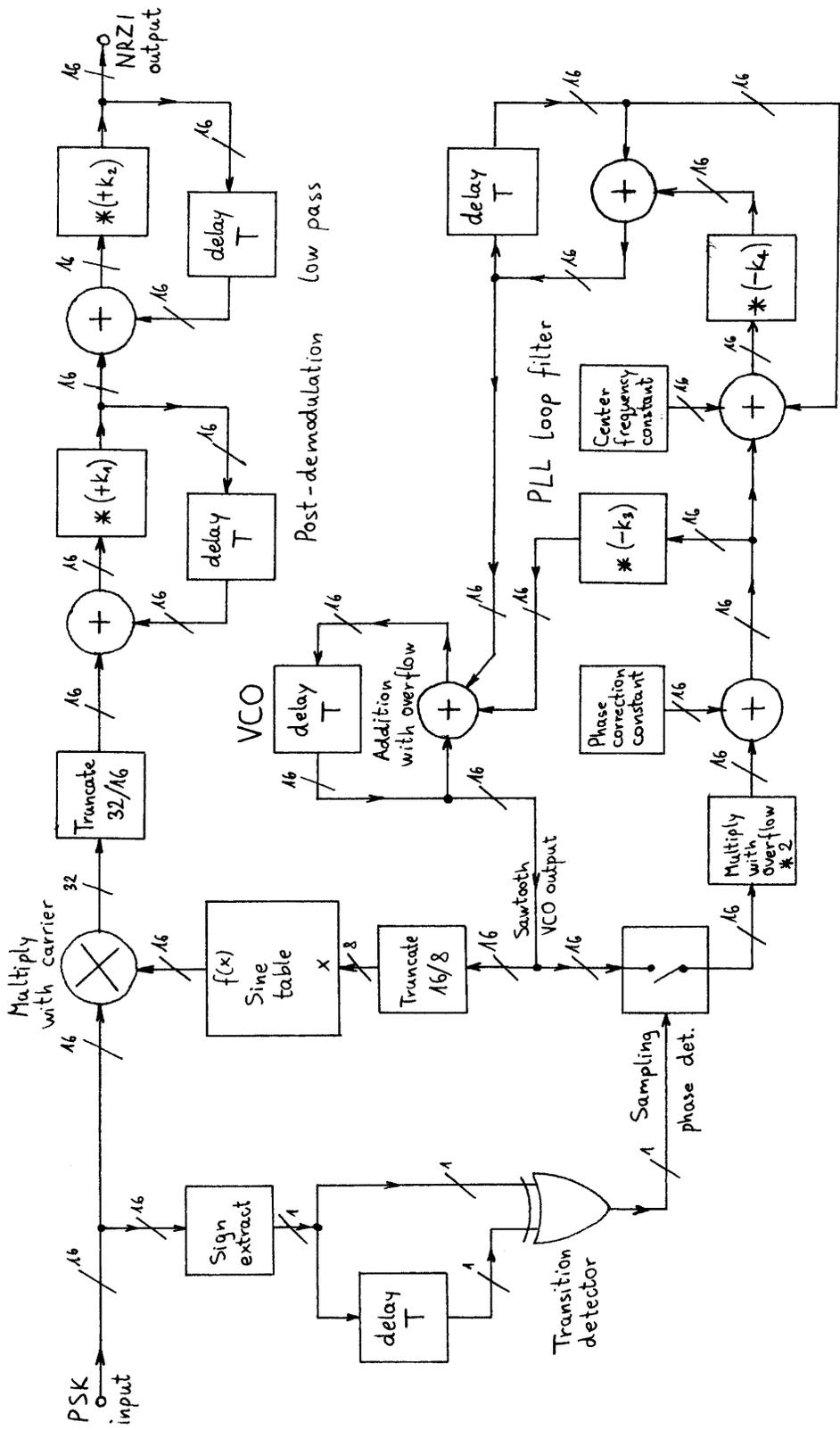


Fig. 8 - 1200 bps PSK demodulator.

*** 1200 bps PSK demodulator program for MC68010 ***

Input: D1.W

Output: D1.W

Address reference: A2

Other registers used: D0, D2

```
3401  MOVE.W D1,D2          ! Save input sample to D2
3012  MOVE.W (A2),D0        ! VCO output to D0
E048  LSR.W #8,D0          ! Truncate D0 content to 8 bits
E348  LSL.W #1,D0         ! Even address offset
0640  ADDI.W #0250,D0      ! Offset to start of sine table
0250  !
C3F2  MULS #00(A2,D0.W),D1 ! Multiply with carrier
0000  !
301A  MOVE.W (A2)+,D0      ! VCO output to D0
3212  MOVE.W (A2),D1      ! Previous sign to D1
0242  ANDI.W #8000,D2     ! Extract sign in D2
8000  !
34C2  MOVE.W D2,(A2)+     ! New sign to (A2)+
9441  SUB.W D1,D2         ! Any sign difference?
670E  BEQ #0E            ! Branch if signs equal >>-----\
3400  MOVE.W D0,D2       ! VCO output to D2
E34A  LSL.W #1,D2        ! Multiply by 2 (overflow!)
0442  SUBI.W #2000,D2    ! Phase correction
2000  !
3202  MOVE D2,D1         ! Phase difference to D1
E841  ASR.W #4,D1        ! Multiply by 1/16
9041  SUB.W D1,D0        ! Correction to VCO phase
321A  MOVE.W (A2)+,D1    ! VCO frequency to D1 <<-----/
D441  ADD.W D1,D2        ! Get new VCO frequency
945A  SUB.W (A2)+,D2     ! Add center frequency constant
E042  ASR.W #8H,D2      ! Multiply by 1/256
9242  SUB.W D2,D1       ! Get new VCO frequency
3541  MOVE.W D1,#FFFC(A2) ! Save VCO frequency
FFFC  !
D041  ADD.W D1,D0       ! Get new VCO phase
3540  MOVE.W D0,#FFF8(A2) ! Save VCO phase
FFF8  !
4841  SWAP D1           ! Result to lower 16 bits
D252  ADD.W (A2),D1     ! Lowpass #1
E241  ASR.W #1,D1      ! Multiply by 1/2
34C1  MOVE.W D1,(A2)+   !
D252  ADD.W (A2),D1     ! Lowpass #2
E241  ASR.W #1,D1      ! Multiply by 1/2
34C1  MOVE.W D1,(A2)+   !
```

Fig. 9 - PSK demodulator program example

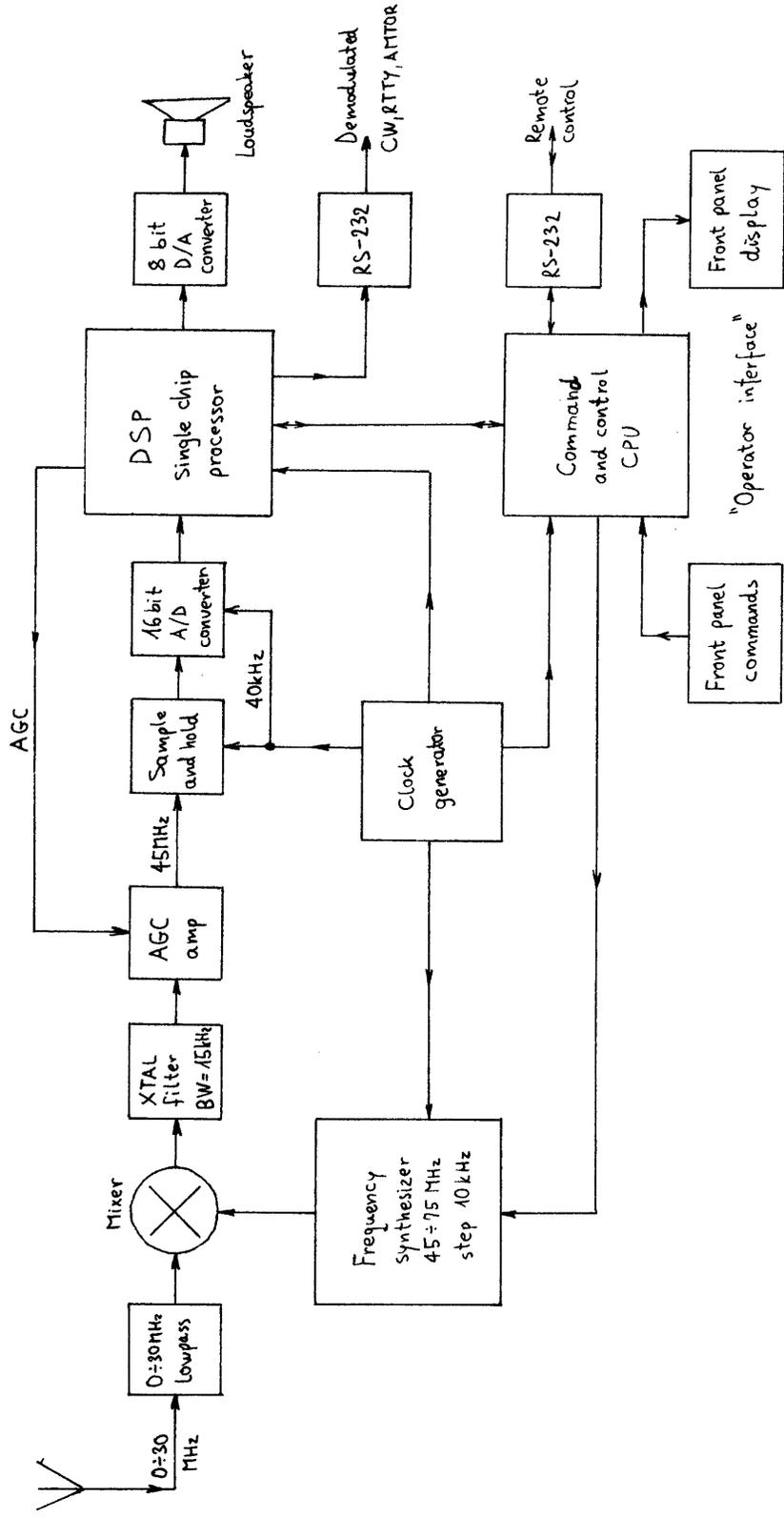


Fig. 10 - An "all mode, general coverage" DSP oriented HF receiver.

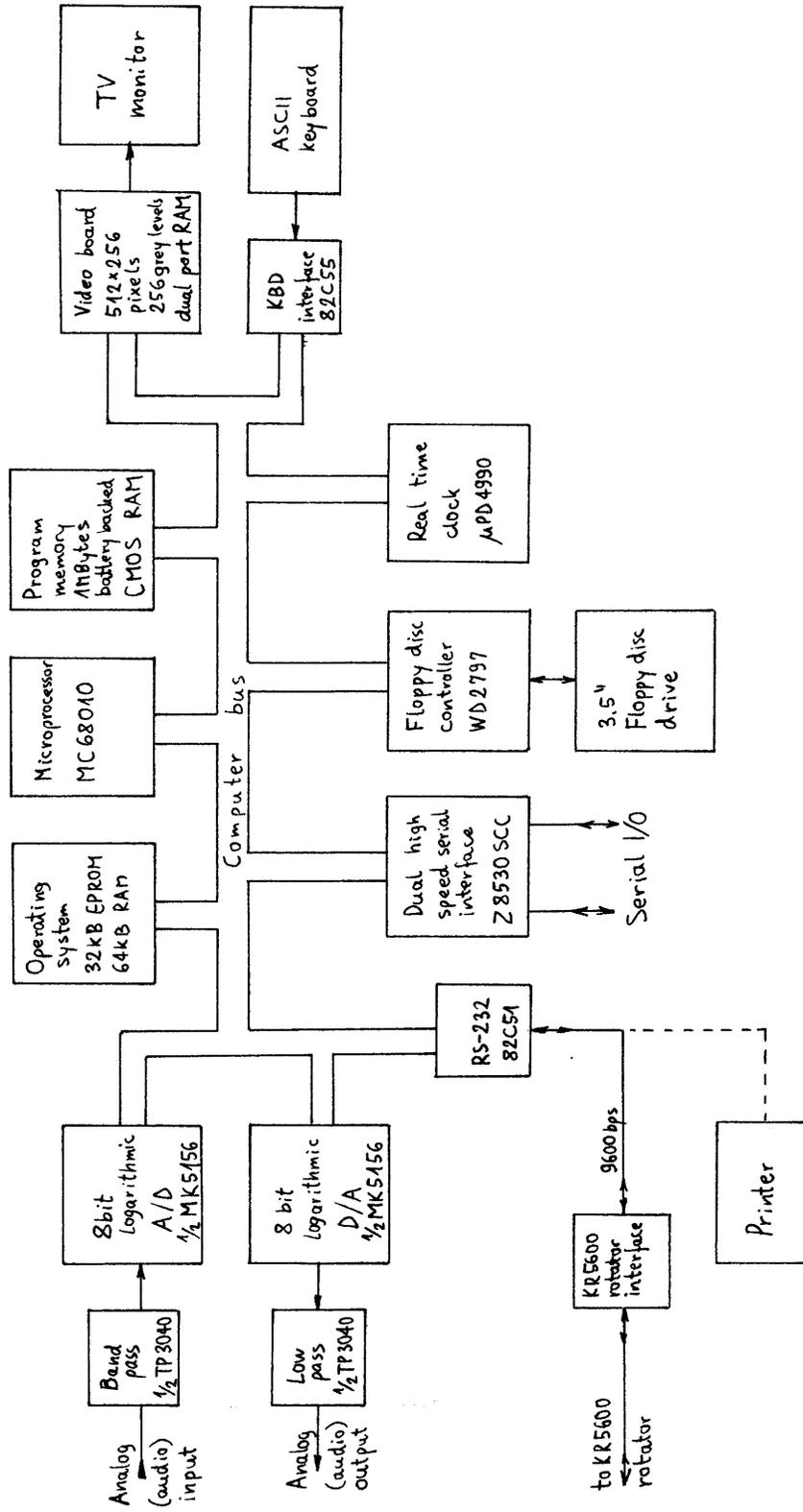


Fig. 11 - Block diagram of the prototype DSP computer.