DIGITAL SIGNAL PROCESSING USING GENERAL PURPOSE MICROPROCESSORS

Matjaz Vidmar, YT3MV (ex YU3UMV)

1. Introduction

One of the main problems of amateur satellite experimenters is the large number of different communications standards used: virtually any satellite uses its own modulation and data encoding standard and requires its own, matched data demodulation hardware.

This is probably the reason why digital signal processing (DSP) techniques are becoming so popular among satellite experimenter: a single microprocessor can be used that can replace many analog circuits. Different circuits can be implemented by simply changing the software being executed by the microprocessor: amateurs do not need all the demodulators at the same time.

DSP circuits usually include A/D and D/A converters to interface with the analog world and a powerful computer to solve numerically in real time the mathematical equations describing the analog circuit it replaces. Usually a very large number of computations per second is required and specialized, expensive dedicated DSP microprocessors have to be used.

Since amateur receivers and transmitters only provide or require voice bandwidth and dynamic range signals, many DSP circuits can be built with general purpose microprocessors, in particular with 16 bit microprocessors. Modern 16 bit microprocessors like the Intel 80286, NEC V30 or Motorola MC68010, are able to compute a 16 by 16 bit product in 3 to 5 microseconds. This is sufficiently fast to build DSP filters up to 10 to 15th order or similar complexity DSP circuits, in practice all the demodulators and modems for voice bandwidth signals.

A prototype DSP computer has been built based on a 16 bit general purpose microprocessor MC68010. Several practically very useful programs have been prepared and tested that are mentioned later in this text. Finally, a detailed description of the most interesting part of one of these is given: a DSP 1200 bps PSK demodulator.

The following description is a part of a much more detailed article which is actually being published in the radio amateur magazine UKW-Berichte / VHF-Communications.

2. Hardware developed and tested

The block diagram of the prototype DSP computer is shown on Fig. 1. The computer is built on several printed circuit boards connected together with a mother board with 64 pole "eurocard" 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 calender clock.

B) Video board including 128 kbytes of dual port dynamic video RAM, video D/A converter and all the timing circuitry,

generatig a dispaly made of 256 lines of 512 pixels of 256 grey levels each.
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 "eurocard" 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.

3. 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 avilable 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 retriving 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, it was decided to use CMOS static RAMs.
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 multitask 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 developement.

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 stadard 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. 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 grey 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 multitask so that only one computer is required to track a satellite and process the received data at the same time.

4. A DSP 1200 bps PSK demodulator

JAS1, renamed FUJI-OSCAR-12 after its succesful 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 onboard computer. The latter is usually programmed to work as a multi-user mailbox to support store-and-foreward type communications between radio-amateurs. The downlink 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. 2.

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. 2).

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 sinewave 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. 2 was practically built in the form of a machine code program for a MC68010 microprocessor (shown on Fig. 3) 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 since the whole AX25 PSK program, when compiled into 68010 machine code, takes over 70 kbytes!

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 uplink performance. With such a high performance demodulator and a higher power transmitter (around 200W) operation with the JAS1 mailbox using omnidirectional antennas should be possible.

##

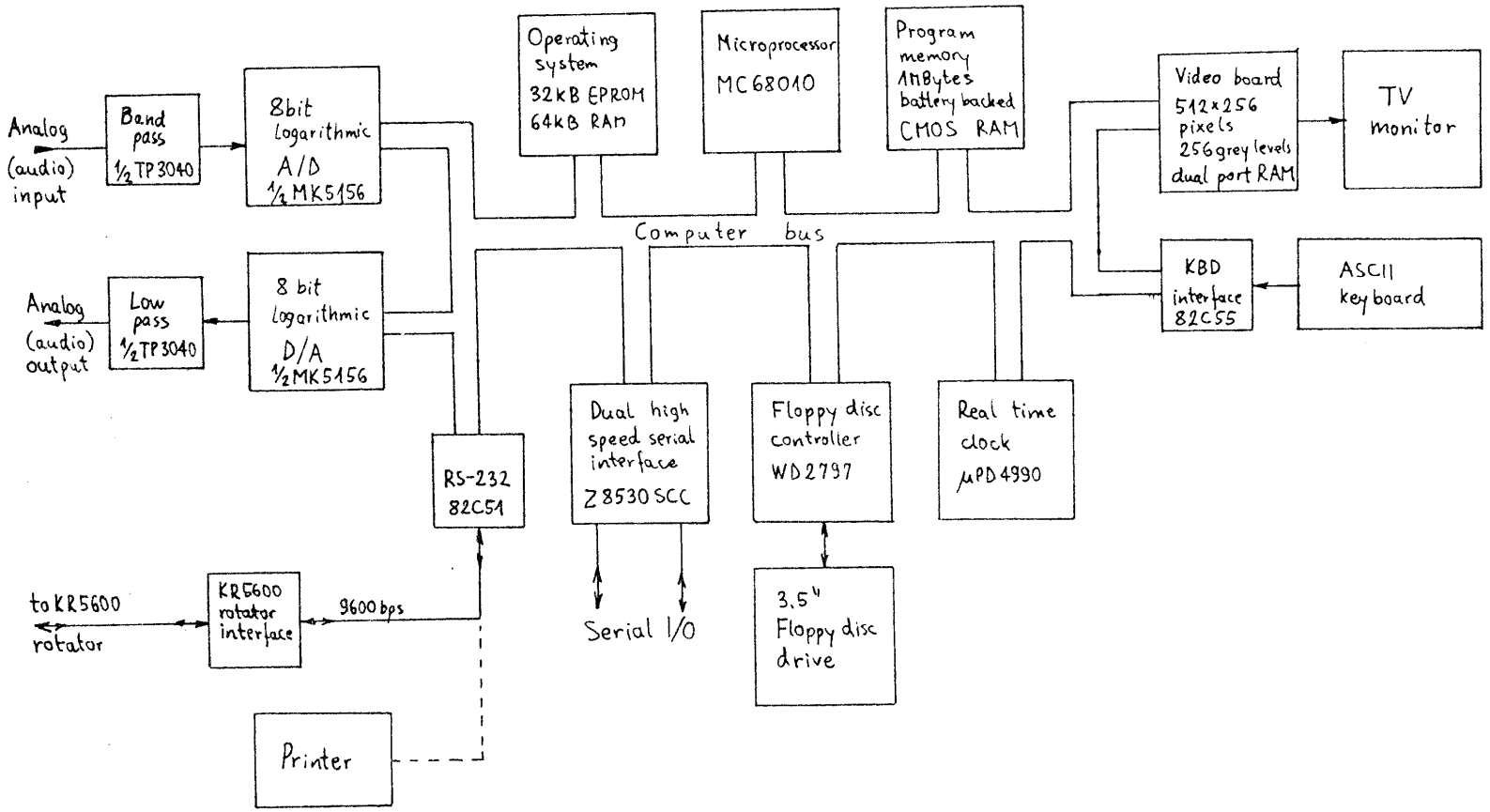

Fig. 1.- Block diagram of the prototype DSP computer. YT3MV

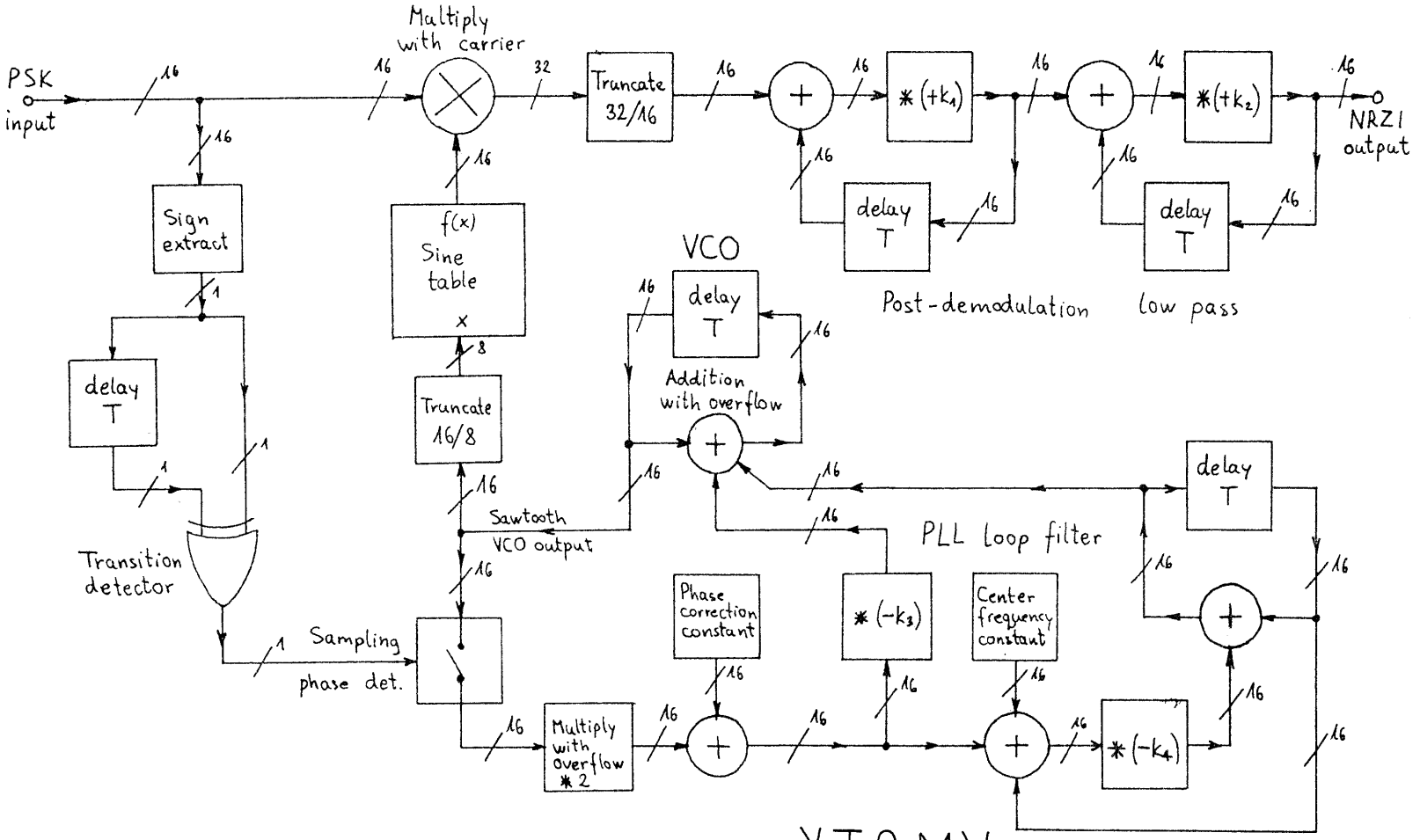

Fig. 2.- 1200 bps PSK demodulator. YT3MV

```
*** 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 ofset
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. 3 - PSK demodulator program example YT3MV