Technical report about the digital part of the DSP-circuit for the Tactile Hearing Aid (THA.I)

Eindhoven University of Technology
October, 1988

Technical Report about the Digital part of the DSP-circuit for the Tactile Hearing Aid (THA.I).

R.W.M. Mathijssen

Intern rapport 88EME02

Abstract

An experimental system for a tactile hearing aid using a Digital Signal Processor (DSP) has been developed at the division of Medical Electrical Engineering at the Eindhoven University of Technology. Not only the familiar techniques for a tactile hearing aid (such as energy level display, filterbank analysis, etc.), but novel techniques too, can be applied, tested and evaluated on this system.
The system has been developed especially to try out new recognition strategies, since the current available strategies are still not satisfactory enough. A portable tactile hearing aid that can recognize certain environmental sounds (alarm sounds) and certain features from the speech signal (such as pitch, voiced/voiceless, or even complete phonemes), being a good support for lipreading, should be the final result of the experiments.

Contents

Introduction

Tactile hearing aids (T.H.A.) have been developed and tested for several decades, e.g. [Sherrick, 1984]. However none of these aids proved to be very useful in supporting lipreading or recognizing sounds in the surroundings. Some tactile aids showed, that it is possible to discern a lot of words after a short period of training [Brooks, 1983, 1987]. None of them however, was so succesful, that deaf people used them over a longer period of time.

The problem that most of these aids had, was that they gave either too much information (during a certain time), so that the tactile sense could not keep up with the speed of the offered information, or that they offered the wrong information, e.g. information that can be obtained by lip-reading without the aid too.

Trying to make a tactile hearing aid, that gives just enough useful information, is what we try to do in our group. Of course, this is not as simple as it is stated here. First of all, we have to know what kind of information is useful. We hope to obtain (part of) this information during talks with deaf people (if possible, but not necessarily, people who have once used a tactile aid) and their speech-teachers. We expect that these interviews will not only give us the information we ask for, but also other ideas that can be very useful; ideas that a normal hearing person usually does not think of.

When we have found out what is useful information, we have to try to extract this useful information from the speech- and sound-signals. This is why we have made a signal processing system, that will be discussed in this report. With this system, we can examine what can be extracted and what can not (yet) be extracted in real time.

Finally, when the information is extracted, we enter the phase, where we have to test whether the information is indeed useful. It might be, that some information only seemed to be useful, but, once offered to the deaf, happens to be useless. In order to test this too, a real-time system is needed, being discussed in this paper.

Design rules for the system

The System that processes the signals, has to be designed according to some rules. These rules are not 'hard' rules, telling exactly what the performance of the system should be, but they are 'soft', i.e. they are some sort of vague description of the final system. Hard rules can only be expectetd after a thorough evaluation of the experimental set-up. Several short-comings of the system can only be found by working on it (while writing the software) and using it (when deaf people use the system).

The rules that are most important to make an experimental system are the following:

* the system should be fast enough, since it has to work in 'real-time'

* the system should be universal enough, to try out several approaches; it should be dedicated enough, in order to keep it simple (in using and programming)

* the system must be portable and battery-powered, i.e. low power consumption. (see below).

These rules are more than enough to start with. Further rules will definitely appear during writing the software etc., but it would take too much time to think about them beforehand (before writing the software and testing it). In order to design a perfect system, one should write the software for the complete system first (the system in between the microphone and the exitators). However, this might result in a hardware design, that is far too complex. Also, to write the software for a specified system, one has to know the limitations of the system. Therefore, the easiest thing to do, is make the hardware first, as complete as possible, and then write the software.

The system should be battery-operatable, so it can be used everywhere. Another reason for battery-operation, is the safety. At this moment we use loudspeakers as vibrators. Should we ever use stimulation by electric impulses, we have to have a **safe** system that is in no way connected to the mains.

Global description of the system

With the above mentioned rules, we came to the following system.

The Digital Signal Processor.

The heart of the hardware for our T.H.A. will be a Digital Signal Processor (D.S.P.) that processes the sounds into LPC- or FFT-parameters. Although e.g. special purpose LPC-analysing chips are available, a DSP offers a lot of advantages compared with dedicated IC's. The first advantage usually is the power consumption. DSP's can be obtained in CMOS-technology. Together with supporting hardware (also in CMOS) such a DSP still uses less energy than most speech-analysis IC's such as the SP1000 - a dedicated LPC-analyser. Another great advantage is that a DSP can be programmed to do all kind of different sorts of analysis. So, if one might decide trying another kind of analysis, it is quite easy to change it; with a dedicated chip this is not possible at all: you have either LPC-parameters (SP-1000) or filterbank-output (NEC 7763) etc.

The reason for using a DSP instead of a (single-chip) micro-processor, is that the latter one does not offer us enough computational power.

Finally, one can implement the recognition software and coding software in the DSP too. This gives the advantage, that no power is being used by a (single-chip) microprocessor and its surrounding hardware. Later however, we shall see, that adding a single chip microprocessor may have its advantages too.

Choice of the Digital Signal Processor.

When choosing the DSP one has to take several things into account.

* The DSP must be in CMOS technology (one of the design rules).
* It should be fast enough to do LPC-analysis and FFT together with recognition and coding in real-time.
* During development the program memory must be external.
* If possible, there should be enough support from the dealer of the DSP.

A lot of DSP's have been reviewed, and the Signal Processor from Analog Devices, the ADSP-2100 seemed to fit to our wishes quite well.

This DSP had, however, one disadvantage - though not for the very near future. The ADSP-2100 is not (yet) available with built-in program- and data-memory. This is disadvantageous, because there is no possibility to reduce space and powerconsumption by moving memory into the DSP; something that is possible with a lot of other DSP's. However, should it prove worthwhile to move memory in the DSP and should Analog Devices not come up with an ADSP-2100 with internal memory, we can always choose another DSP - perhaps an even better one than is available on this moment. (Note: at this moment, Analog Devices is developing a DSP **with** internal RAM-memory).

Analog to Digital Converter.

The Analog to Digital Converter (ADC) is the AD7580, also from Analog Devices. This is a ten bit ADC with internal Sample and Hold Amplifier. This means, that we only have to provide this ADC with an analog signal with 5 V_{pp} (of course low-pass filtered to prevent aliasing); an external sample and hold amplifier is not required. The ADC is a CMOS device, that only needs a single 5 volt supply.

Memory.

The program memory consists of three 32KByte CMOS EPROM's, to make a program memory of 16KWord of 24 bits, and a (fixed-)data memory of 16KWord of 16 bits.

Data memory is made of two 32KByte CMOS RAM's, making a (variable-)data memory of 14KWord of 16 bit. The DSP can access only 16KWord data memory direct. Since part of this data memory is used for I/O (Memory-mapped I/O), only 14KWord RAM can be used, using only little external hardware. Further, there are no 16 KByte RAM's available, so we have chosen to use 32KByte components. (At the end of this paper the use of twice as much RAM memory - i.e. 28 KWord - will be discussed).

Clock frequency.

In order to make the system battery-operatable, only CMOS components will be used. The next step to reduce power consumption, is to use a clockfrequency as low as possible. Most components can work on a clock with a frequency that is at least twice as high as the current clock. The Analog to Digital Converter operates on 1.5 MHz (2.5 MHz is allowed), giving samples at a speed of about 12 KHz (50 KHz maximum). The Digital Signal Processor operates on 12 MHz, while 32 MHz is its maximum. The memory components operate on a variable speed, depending on the instructions the DSP executes. The maximum clock frequency they are accessed with, is 3 MHz. The EPROM can operate on 8MHz; the RAM on 10MHz.

We have chosen for a clock frequency of 12 MHz for the DSP. On this frequency, the DSP can do FFT analysis and LPC analysis real-time, using approximately less than half of the total time available. The other part of the time can be used e.g. for recognition. For the A/D Converter, we came on a sample rate of about 12 KHz, being 12 MHz divided by 2^{10}. Also, 12 KHz is a frequency that is used quite often in speech recognition systems.

As a start the 12 MHz appears to be high enough to do the necessary operations, and low enough to have an economic system regarding powerconsumption (see Electrical specifications, p.16).

The system has memory-mapped I/O. We have used the upper 2 KWord memory space for I/O. Not because we need 2K I/O space, but because of the ease of making a decoder, dividing the memory in a 14K and a 2K part. Our system actually uses 2 I/O ports: the A/D converter and a 16-bit latch (e.g. for driving the vibrators). Expansion to 8 I/O ports is possible with only minor hardware adjustments.

The Digital Signal Processor

Here we shall briefly discuss the ADSP-2100 Digital Signal Processor, to give an impression of the possibilities of the device. (see also: ADSP-1000 User's Manual)

The ADSP-2100 is a programmable single-chip micro-processor, optimized for Digital Signal Processing (DSP) and other high-speed numeric processing applications. It contains three full-function and independant computational units: an Arithmatic/logic unit (as in most micro-processors), a multiplier/accumulator and a barrel shifter. The computational units process 16-bit data directly.

As can be seen in figure 1, the ADSP does not have any internal program memory nor any data memory (the cache memory might be seen as a temporal program memory, storing a short history of up to 16 previously executed instructions); both program memory and data memory must be externally provided for. Together with its peripherals (such as an A/D converter and - in our case - control logic for the vibrators) the ADSP-2100 system can be seen in figure 2.

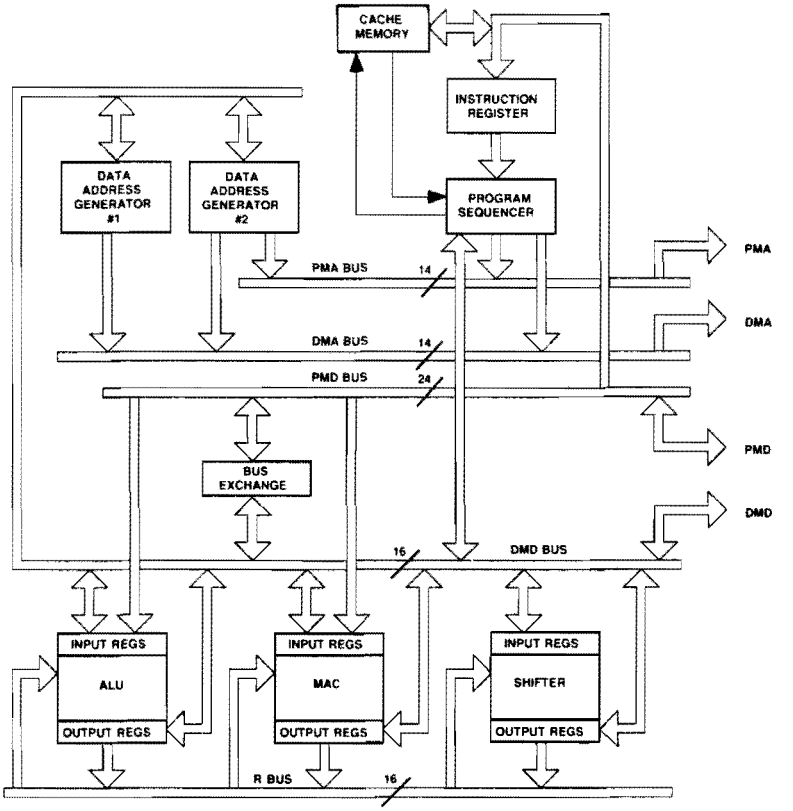

Figure 1. ADSP-2100 Internal Architecture (ADSP-2100 User's Manual)

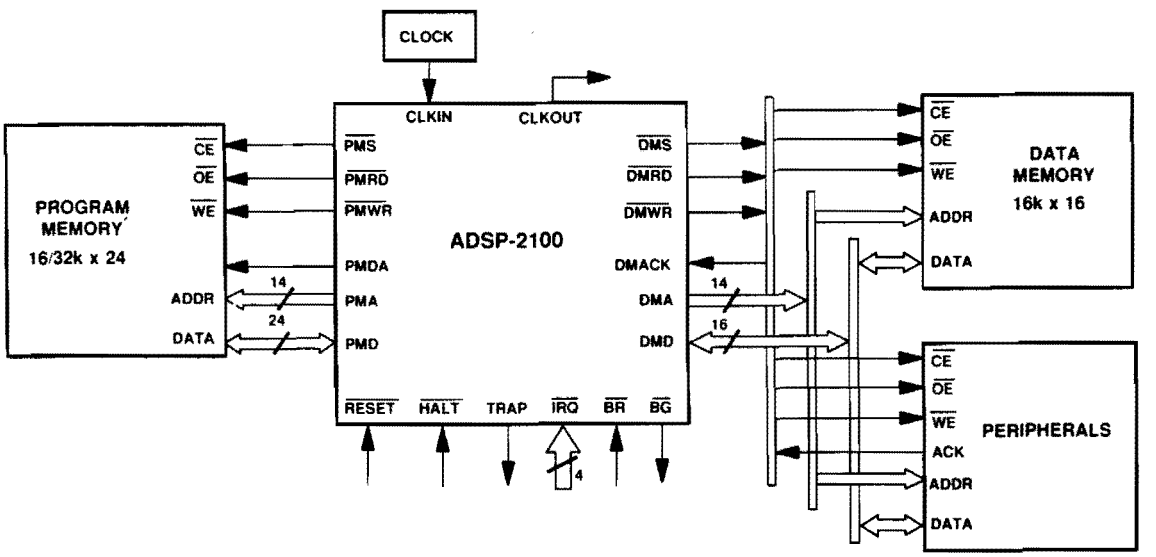

Figure 2. Minimal ADSP-2100 system (from: ADSP-2100 User's Manual)

The ADSP-2100 can operate on a clock frequency of 32 MHz, which equals an instruction cycle time of 125 nsec. However, to reduce power consumption, the clock frequency is reduced to 12 MHz (a rough calculation shows, that the DSP operating on this frequency needs less than half of its available time to do an LPC- or an FFT-analysis). Next to the power advantage, this lower clock frequency also enables the use of less fast memory- and supporting components.

Description of the Hardware

The complete circuit of the digital part of the T.H.A. can be seen in figure 3. As this figure shows, hardly any additional hardware - next to the processor, the memory, the ADC and the latches - is used to make the system operate. Here we shall discuss the function of the additional hardware.

The CLOCK circuit.
To generate a 12 MHz clock signal, we have used the circuit from figure 4. This configuration operates quite well, especially for relatively high frequencies. The leftmost inverter (G1) is made very instable by means of the feedback resistor of 3.3 KΩ. The middle inverter (G2) 'polishes' the output signal from the first inverter (G1) and feeds it back to this inverter (G1) through the crystal (the capacitor in series with the crystal has so high a value, that it hardly effects the signal at all). This crystal 'filters' out every frequency other than 12MHz, so the first inverter can only produce a 12 MHz signal (in this configuration). The rightmost inverter (G3)is used to buffer the 12 MHz signal. The binary divider is used to generate the required frequencies (1.5 MHz and 11718.75 Hz) for the A/D converter.

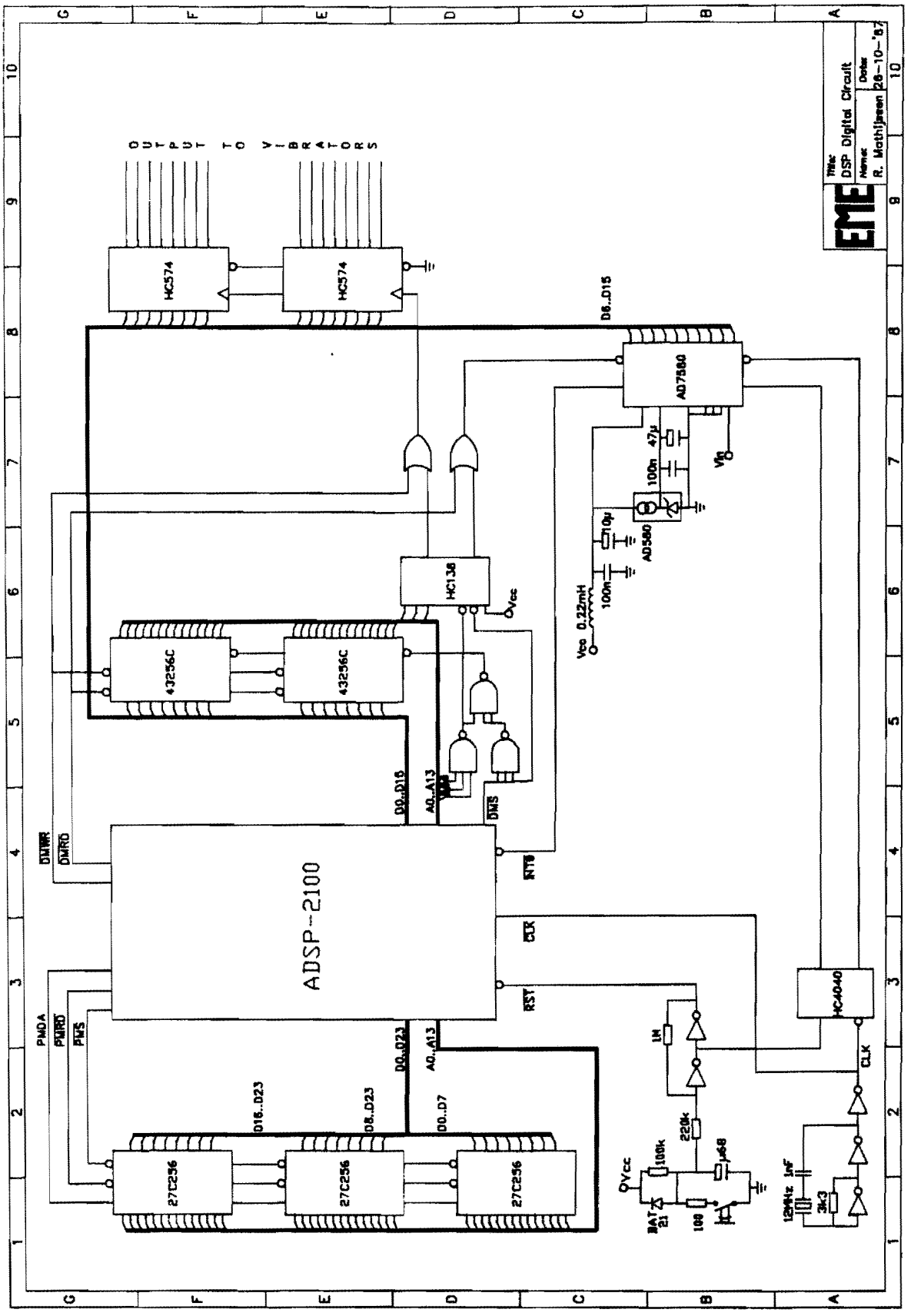

Figure 3. Complete hardware design.

The RESET circuit.

To start the DSP correctly, it needs to be reset when power comes up. This is accomplished by the circuit from figure 5. The RC combination (110KΩ and 0.68μF) takes care of a reset-pulse that is long enough to reset the DSP. The two inverters (G4 & G5) are to buffer the reset pulse (by means of a schmitt-trigger configuration: resistors of 220kΩ and 1MΩ) and to invert it for the binairy divider in the CLOCK circuit. Resetting the divider is done to 'synchronize' the ADC with the DSP. Thus the first time a conversion is finished, is after a fixed number of processor cycles. To enable a manual reset, a push button is added parallel to the capacitor (with a 100Ω resistor, to protect the capacitor and the switch). The diode is added to prevent negative pulses at the inverter input when the power is switched off.

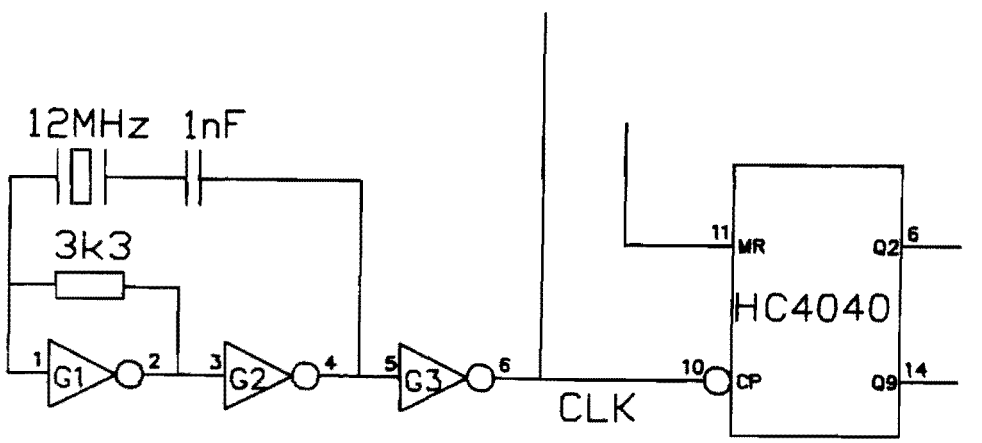

Figure 4. The CLOCK circuit.

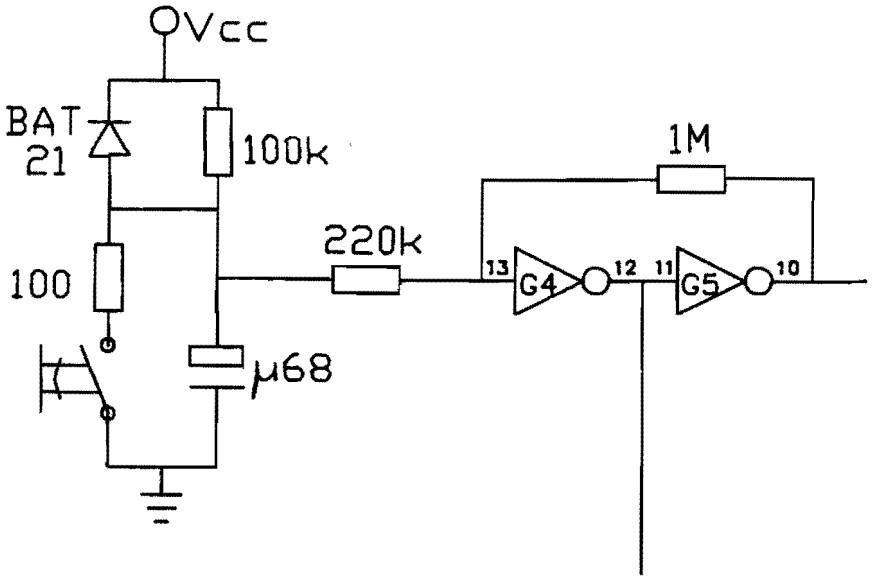

Figure 5. The RESET circuit.

Address decoder.

The address decoding for the data RAM is performed by three 3-input NAND-gates. The Data Memory Select ($\overline{\text{DMS}}$) line is gated with the 3 most significant bits of the addressbus, in such a way, that only when those 3 bits are not all at a high level, the $\overline{\text{CS}}$ for the RAM becomes active.

The output from the NAND, connected to the address bus, also controls the 1 to 8 decoder (HC138), which can select either the latches or the A/D converter. The output signal from the 1 to 8 decoder is OR-ed with the $\overline{\text{DMWR}}$ or the $\overline{\text{DMRD}}$ line to control the previously mentioned devices. The address decoder circuit can be seen in figure 6.

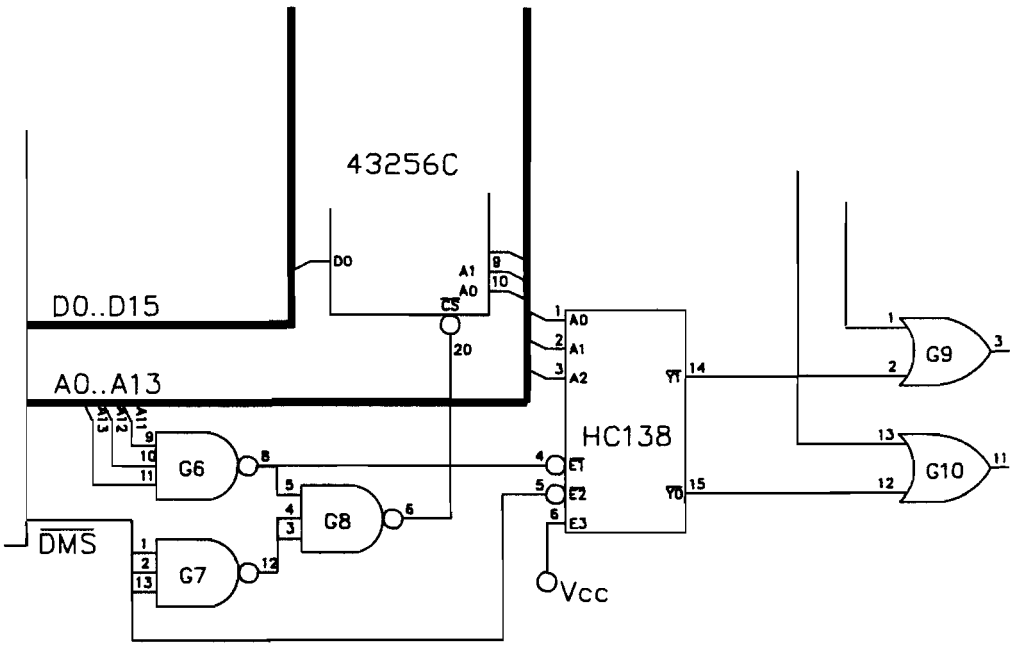

Figure 6. Address decoder.

When more peripherals are needed, the other outputs from this decoder can be used directly. If necessary, the output can be OR-ed with a read- or write puls with the two spare OR-gates.

VLSI-chips.

Not much can be said about the connection of the DSP, Memory and ADC. Since Program memory is 24 bits wide, three EPROMs are required. The Data memory is 16 bits wide, so we need two 8 bit RAMs. The A/D-converter is connected to the Data memory databus on the 10 MSBs. Thus when input is read from the ADC, it needs to be masked, to zero the 6 LSBs. The ADC tells the DSP that data is available by means of an interrupt.

Electrical specifications of the Hardware

Since one of the requirements of the design was an energy economical system, some measurements on power consumption were done. Table 1 shows the current each sub-system consumed during a specific task. Ten different actions were performed to measure the various currents:

0) DSP in a loop, without DM or PM action
1) DSP in a loop, without DM action, with PM action
2) DSP in a loop, without DM action, with PM-data read.
3) DSP in a loop, with DM write, no PM action
4) DSP in a loop, with DM read, no PM action
5) DSP in a loop, with both PM (data) and DM read
6) DSP in a loop, with PM read (program), DM write
7) DSP in a loop, without DM or PM action, with reading ADC
8) DSP in a loop, reading ADC, acquisition of data, write to DM
9) DSP in HALT-mode

During these actions, no power was used by LED's etc. The sub-devices measured were:

A) DSP alone
B) RAM alone
C) EPROM alone
D) ADC alone
E) gates alone
F) total circuit

Since only one system was available, the measured values can differ when various components are used. Further, when performing e.g. an F.F.T. all those actions mentioned above will occur in some kind of mixture. Therefore the measured values are some sort of worst-case values for this system.

Table 1. Power consumption of the various subsystems (in mAmps).

	(A) DSP	(B) RAM	(C) EPROM	(D) ADC	(E) gates	(F) total
(0) no DM&PM	7.40	0.01	7.97	4.50	8.65	28.7
(1) only PM	8.56	0.01	8.95	4.47	8.65	30.9
(2) PM data r	3.50	0.01	12.23	4.47	8.65	29.5
(3) DM data w	7.49	45.9	8.70	4.45	8.63	79.4
(4) DM data r	7.17	0.72	8.75	4.47	8.66	30.1
(5) PM&DMread	12.17	0.69	12.13	4.46	8.65	38.7
(6) PMrd&DMwr	8.74	46.3	9.82	4.45	8.63	82.0
(7) ADC read	7.72	0.01	8.26	4.45	8.66	29.5
(8) ADCrd,acq	7.95	0.32	8.35	4.46	8.72	30.1
(9) DSP HALT	0.48	0.01	0.01	4.55	8.66	13.6

As can been seen in table 1, the power consumption varies for different operation modes. The power consumption of the RAM and the EPROM is measured in both its maximum and mimimum consuming state. The power consumption of both the ADC and the Gates varies hardly in the different operation modes. The only component which is not yet fully examined, is the DSP. During the test program, it only performed 'minor' instructions, such as No Operation (NOP) and memory moves. It might be possible, that arithmetic instructions (such as multiply & accumulate, combined with data move) are much more power consuming.

In order to reduce power consumption even further than by merely using CMOS components, it is possible to 'switch-off' the DSP, by means of a HALT instruction. To restart the DSP, however, some sort of intelligent circuit is needed, which decides when DSP action is really useful and when it is not useful. In the paragraph 'suggestions' this idea will be explained further.

Another means to reduce power consumption, is making the Gate-part and the ADC more economic. This, however, needs further investigation, since this circuitry is always required.

Finally, figure 7 shows some control signals for the EPROM and the RAM during several operation modes. This shows e.g. that the EPROM is always selected, even when the DSP does not read the Program memory, e.g. during cache operation. Relatively a lot of current could be saved, if the EPROM would not be selected. Unfortunately, this can not be accomplished by means of extra hardware. Selecting the EPROM is done in the DSP and no control signals from the DSP can be used to suppres the Program Memory Select line.

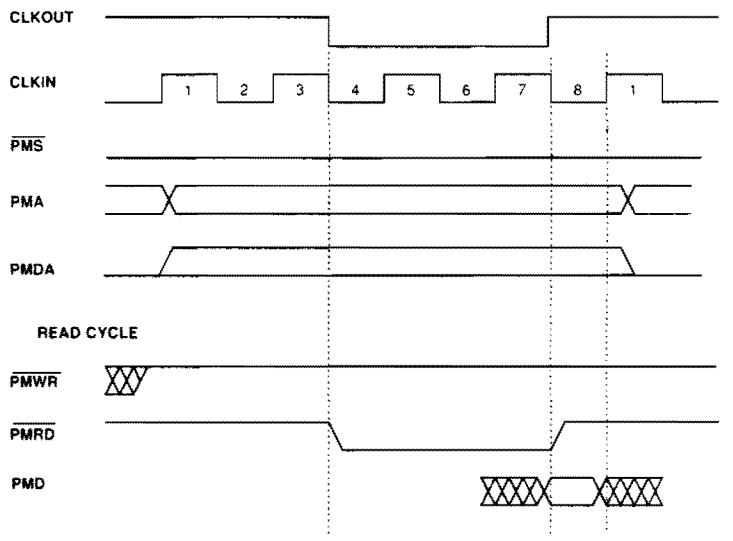

Figure 7[a]. Program Memory Read

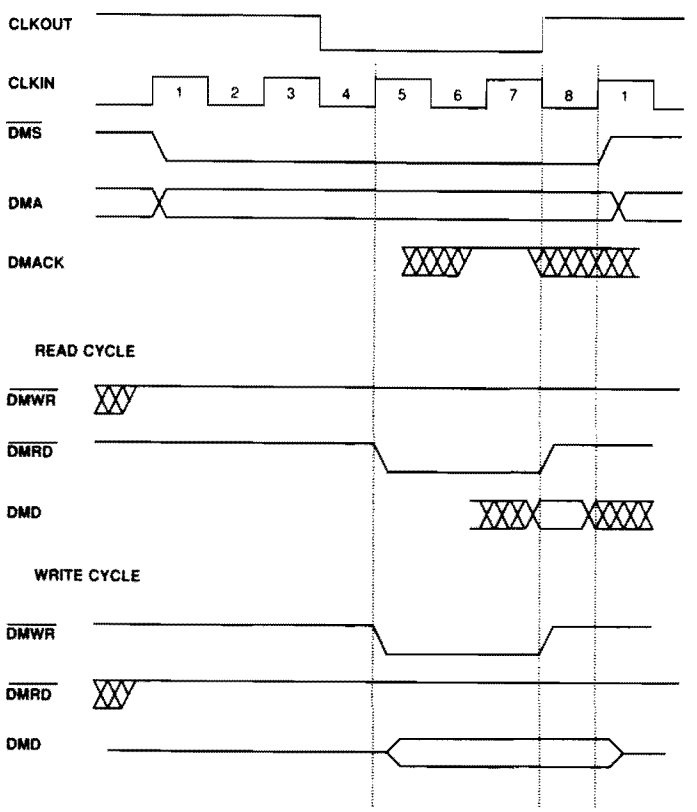

Figure 7[b]. Data Memory Read/Write

Placement of the components.

The components of the system on the experimenting board are placed according to figure 8. When a printed circuit board should be made, the board can be fitted directly in a standard plastic moulded case.

On the experimenting board the components are spaced a bit more widely, to facilitate measuments on the various devices.

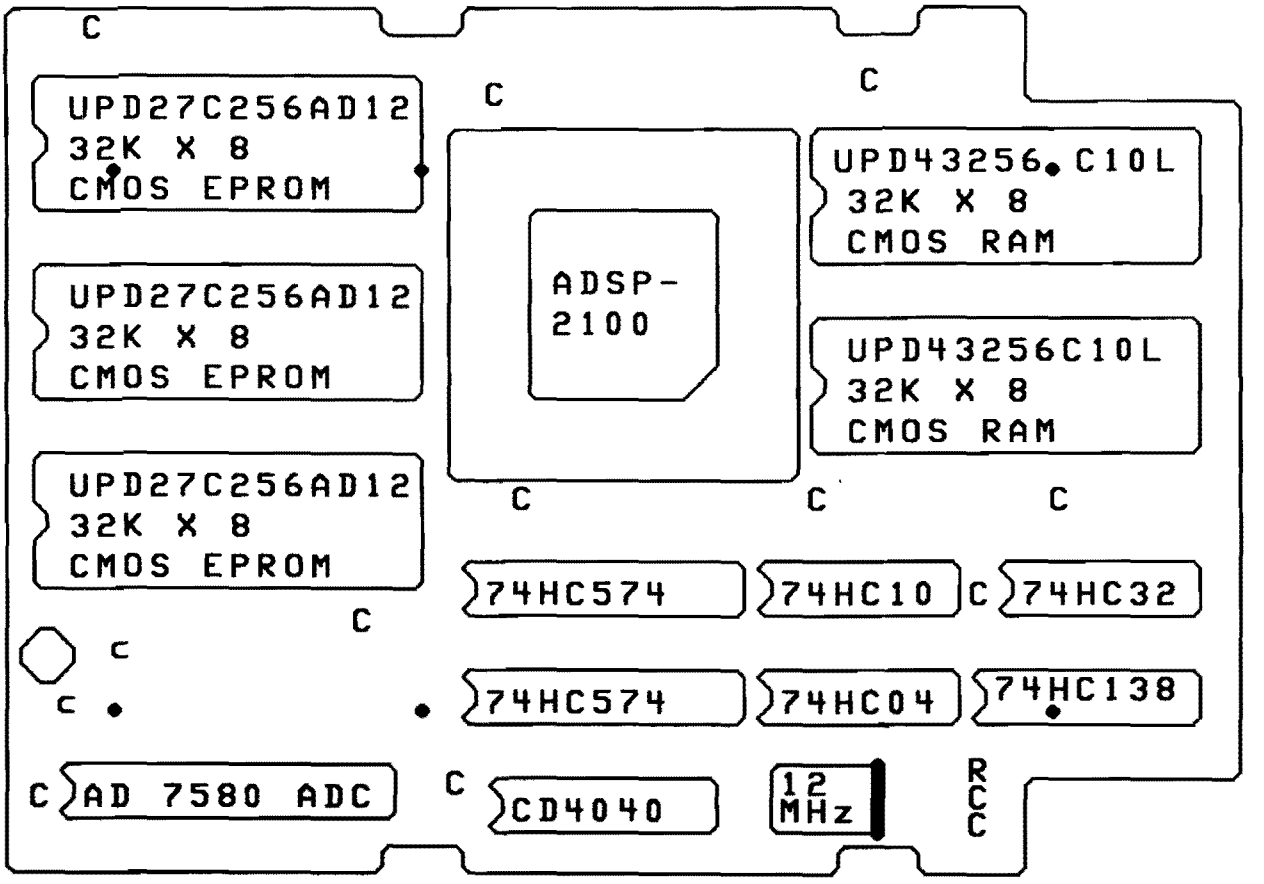

Figure 8. Placement of the components.

Speed of operation.

The Digital Signal Processor has to perform some sort of signal processing (pre-processing), before e.g. recognition or parameter extraction (post-processing) can be done. The processing techniques most commonly used are Fast Fourier Transform and Linear Predictive Coding [Rabiner, 1978]. Since the DSP has to do both the pre-processing and the post-processing, one has to assure that the pre-processing does not use all of its processing time. The same can be said about the post-processing, although this will probably give less serious problems: one can't start the post-processing if the pre-processing is not ready yet, so pre-processing usually simply takes what it needs, concerning processing time.

To get some idea about the amount of processing time needed by a Fourier transform or an LPC analysis, some estimated values are presented. We assume that we always perform an analysis on 256 input samples, which equals a period of about 22 milliseconds.

From [Analog, 1987] we get that a 256 point FFT - using the most accurate routines and a DSP on 8MHz - takes 2.57 milliseconds using radix-2 DIF, or 1.55 milliseconds using radix-4 DIF. However, we use a 3MHz DSP, so we get the following times:

256 point radix-2 DIF : 6.86 msec (equals 20593 cycles)
256 point radix-4 DIF : 4.13 msec (equals 12398 cycles)

Processing time needed for scrambling and data-input using a relatively short time, is not included in these calculations.

LPC analysis, computing 10 parameters and the pitch of the signal, takes approximately 30,000 cycles, which equates 10 msec (mainly due to the correlation routine, using over 80% of this time).

Should one wish to perform both a FFT and an LPC analysis, the pre-processing uses about $^{3}/_{4}$ of the total processing time available.

Implementing various recognition and display techniques.

This system with the DSP can perform various recognition and display tasks. At this moment, the T.H.A. has a 16 bit output port, to activate at most 16 vibrators on a single frequency and amplitude. The circuit to drive these vibrators is made X. Wang and can drive 16 vibrators at 200 Hz and with 2 defined amplitudes (i.e. on/off), or 8 vibrators with 4 amplitude levels (for further information, see X. Wang).

Should one wish to drive more vibrators, it is possible to extend the output port. Using only latches as extra hardware, one can extend it with up to 32 bits. Using an extra quadruple 2-input OR-gate too, it can be extended to up to 96 bits.

Should one wish to use a variable frequency, a Digital to Analog converter can be added as output extension. When a limited number of different frequencies will do too, it is possible, to select the frequencies with the latches.

At this moment however, 16 vibrators driven with only one frequency, switched on and off, seems to be sufficient to start with.

Since only the digitizing of the signal is a hardware matter, every other processing of the signal - being a software matter - can be changed freely. This means that e.g. filterbank analysis can be performed, or pitch detection with amplitude detection. One can detect the 'major' formants and use these as output; or the distiction between voiced and unvoiced speech can be made.

However, should one wish to implement some sort of signal (speech) recognition, this is possible too. When using e.g. Fast Fourier Transform as a base for recognition, about three quarters of the total processing time is available for recognition purposes: FFT combined with ADC input requires only about 1 quarter processing time (256 point Radix-4 DIF FFT, Conditional Block Floating-Point Scaling [Analog, 1987]). When performing Linear Predictive Coding, about one half of the processing time (but perhaps even more, when the algoritm is sufficiently adapted) is still available for further recognition [Analog, 1987].

Suggestions.

The system as it is built works rather well. There are however still a few matters that can be altered, to make the system operate even better, to make the system suitable for more kinds of recognition or display, or to reduce power consumption even further. Some suggestions to improve the system concerning display and sound processing techniques were mentioned before. Other suggestions, concerning the overall performance of the system will be mentioned below.

First of all, power consumption should be diminished further, when the system must be portable. Table 1 shows that the system uses approximately 30 to 40 milliamperes in normal operation. When using normal NiCad penlight batteries, this means that the system can operate for about 12 to 15 hours without recharging. However, not only this part has to be fed by the batteries; the analog input circuit (amplifier, filter) and especially the output circuit (the combination of driver and vibrator) will use a lot of energy too (how much will depend on the number of vibrators and the kind of vibrator, such as electromechanical or piëzo electrical). Therefore, the less energy this DSP-system uses, the longer the complete system can run on one set of batteries.

About 25 % of the energy consumption is used by the ADC and the gates. Perhaps this consumption can be reduced when these components are examined more closely. Another way to reduce power consumption, is switching of parts of the circuit when it is not needed, e.g. in a silent surrounding. This might be possible by means of adding a single chip micro processor, controlling the circuit.

Adding a single chip micro processor also brings another possible advantage. Now the Analog to Digital converter is 'clocked' by means of a hardware design. This means, that we have a fixed sample rate. It can be useful, however, to be able to change the sample rate, e.g. when trying out several recognition algorithms. A variable sample rate is possible, when a micro controller is added, that can control the AD-converter.

Should one wish to alter the sample rate, only the software for this micro controller has to be changed.

At this moment, the AD-converter notifies the DSP that conversion is completed by means of an interrupt on INT0 of the DSP. However, INT0 is the interrupt with lowest priority, so should one wish to use other interrupts too, it might be wise to connect the ADC to an interrupt line with higher priority.

When a printed circuit board is made for the system, the placement of the components should be reviewed, in order to get shorter wiring. Especially concerning the AD-converter, shorter wiring is desirable (to diminish cross-talk). Also, the input from the ADC, together with the voltage reference should be placed in such a way, that control- and databus lines do not mess up these analog signals (cross-talk). This means: reference and input signal on the border of the printed circuit board; digital lines, on the inside of the board.

Further, connecting the DSP might give some problems, since it has 100 pins in a rather small area. It is possible, that a double layer printed cicuit board does not offer enough room; perhaps a multi layered board is required, or a partially wired printed circuit board.

As was mentioned before, it is quite easy to extend the number of peripherals. However, peripherals (i.e. latches) can be used to extend the data RAM address range too. Since 32 KByte RAMs are used, and only 14KByte is used by the DSP, we can think about increasing the address range to e.g. 28KWord. This can be done by means of bank switching. Using one bit of a latch, one can control the most significant bit of the RAM, now put on a fixed level. This means that a software switch can control what part of the RAM will be used: the lower part or the upper part.

In practice the lower part can be used e.g. as a working memory, while the upper part contains e.g. statistical information about the occurence of several states or statistical values of the input signal. The upper part of the memory can even be used as an extra working memory (perhaps necessary when performing a 2048 or a 4096 point fast Fourier transform).

Conclusions.

At this moment we have a working system, performing quite well, for testing various algorithms. Every part of the system is tested on hardware errors; none were found.

The Analog to Digital Converter operates well too. No disturbance due to digital signals was found on the output.

Instead of vibrators, it is also possible to drive 16 LEDs, indicating the status of the output port. Checking the output ports is easier by means of visual inspection than by means of tactile inspection.

The power consumption should be lowered, when we want a portable device. This seems to be possible by using an extra micro controller.

Adding peripherals or increasing the DATA memory is possible by adding only a few extra components.

References.

Brooks, P.L. Frost, B.J. Evaluation of a Tactile Vocoder for Word Recognition.
Journal of the Acoustical Society of America. Vol 74, nr. 1.
July 1983. pp. 34-39.

Brooks, P.L. Frost, B.J. Mason, J.L. Gibson, D.M. Word and Feature Identification by Profoundly Deaf Teenagers Using the Queen's University Tactile Vocoder.
Journal of Speech and Hearing Research, Vol. 30, Mar. 1987.
pp. 137-141.

Rabiner, L.R. Schafer, R.W. Digital Processing of Speech Signals.
Prentice-Hall,Inc.,Englewood Cliffs,New Jersey.ISBN 0-13-213603-1

Sherrick, Carl. E. Basic and Applied Research on Tactile Aids for Deaf People: Progress and Prospects.
Journal of the Acoustical Society of America. Vol 75, nr. 5.
May 1984. pp. 1325-1342.

Analog Devices ADSP-2100 User's Manual. Second Edition. 1986

Analog Devices ADSP-2100 Applications Handbook. Volume 1. 1987

* * * * * * * * * * * * * *