# A Fast Microprogrammed Digital Filter Supporting Early Signal Processing Research

Olli Simula, Yrjö Neuvo



# A Fast Microprogrammed Digital Filter Supporting Early Signal Processing Research

Olli Simula and Yrjö Neuvo

Aalto University, Espoo, Finland
{olli.simula, yrjo.neuvo}@aalto.fi

**Abstract.** In the 1970s research and teaching of digital signal processing was started in several universities in Scandinavia. Special emphasis in the research was on digital filter structures implementable on emerging digital hardware. On the theoretical side, the development of computationally efficient, low-sensitivity digital filter structures was widely investigated. Realization of these novel transfer functions was also important in order to verify the efficiency of the filters in practice. The latest commercial microprocessors indicated that real time signal processing covering the audio range of signals was becoming realistic. A microprogrammable digital signal processor based on high speed bit-sliced architecture was designed at Helsinki University of Technology. With this hardware, the performance of computationally efficient, low-sensitivity digital filter structures was tested. Algorithm research with emphasis on implementation aspects had a great impact on the readiness of Finnish electronics and telecommunications industry to start using programmable digital hardware. Early DSP applications in telecommunications industry were released already in late 1970s. Ten years later, when the first GSM phones were under development, Finnish industry was on the leading edge of applying digital signal processors in a number of applications.

**Keywords:** Digital filters, low-sensitivity realizations, computational efficiency, digital signal processors, bit-sliced architecture, microprogramming

## 1 Introduction

In the late 1970s, digital signal processing (DSP) had been a rapidly growing and dynamic field of research for more than a decade. The first text book in DSP was *Digital Processing of Signals* by Gold and Rader in 1969 [1]. It was followed the widely used *Digital Signal Processing* by Oppenheim and Schafer in 1975 [2].

In the 1970s, DSP research was concentrating on the development of computationally efficient digital filters transfer functions. Due to the limited computational resources, special attention was paid to the coefficient sensitivities of the filter structures. It was, however, important to verify the theoretical results in practice and to demonstrate the advantages of DSP using real applications. Audio signal processing was an excellent example field. These early implementations formed the background for the utilization of advanced DSP algorithms later in the communications applications, e.g., in mobile phones as well as in various image processing tasks.

Among the most active universities in Scandinavia were Linköping University in Sweden, Norwegian Institute of Technology in Trondheim and Helsinki University of Technology (TKK) in Finland. Professor Tore Fjällbrant's group in Linköping did research on wave digital filters and their implementations. The group members, Professor Lars Wanhammar and Dr. Sven Eriksson even wrote a Swedish textbook entitled *Tidsdiskreta filter* (Discrete-time filters) in 1978 [3]. The first DSP course at TKK in Helsinki was based on the manuscript of Oppenheim's and Schafer's textbook in 1974, one year before the publication of the book. In mid 1970s, the DSP research group at TKK consisted of the authors of this paper.

Nordic cooperation in research and education was strong in the 1970s and 1980s. The first DSP conference in the Nordic countries was probably the Symposium on Digital Filtering and Related Algorithmic Problems held in Stockholm, Sweden, April 2-4, 1974. The organizer of the symposium was Professor Lars Zetterberg from Royal Institute of Technology (KTH). The highlights of the symposium were the famous speakers: Dr. James W. Cooley, the co-inventor of the Fast Fourier Transform (FFT) algorithm[1], Dr. Jorma Rissanen one of the pioneers of information theory, and Prof. Teuvo Kohonen, the developer of neural networks algorithms. This conference was followed by several Nordic symposia in the 1980s in DSP and related fields, e.g., in image processing and VLSI implementations of DSP algorithms. The greatest success was the 1988 IEEE International Symposium on Circuits and Systems held in Espoo in June 1988. It was organized as a joint Nordic effort. With its 1100 participants it was by then the largest ISCAS ever. In addition, two Nordic intensive courses on DSP algorithms were organized in Trondheim in 1986 for doctoral students. Professor Tor Ramstad was the responsible organizer of these quite successful events.

The early DSP research in Finland was focused on the development of low-sensitivity digital filter transfer functions with small number of multiplications. Main theoretical results were the so-called MUCRO filters which were based on MUltiple CRitical ROot polynomials [4]. These polynomials were originally intended for analog filter design by Dr. A. Premoli [5], [6]. In addition, low-sensitivity digital filter transfer functions were developed based on the ladder and lattice structures [7]. It was proven that the coefficient sensitivity on the lattice structure was equal to zero at zero frequency [8]. Reducing the number of zeroes in the transfer function resulted in computationally efficient realizations, the so-called one-zero lattice structures [9].

In order to support DSP research and education but also to create industrial interest on DSP applications, a real time hardware implementation was desirable. First microprocessor circuits suitable for signal processor implementations came into the market in the 1970s. The Am2900 family of integrated circuits (ICs) was released by Advanced Micro Devices (AMD) in 1975 [10]. It was constructed using bipolar devices, in a bit-slice topology. This was the fastest bit-slice TTL microprocessor in the market. Bit-slicing techniques allowed the design of a processor with data, addresses,

---

[1] Fast Fourier Transform (FFT) algorithm reduces the arithmetic operations from $N^2$ to $N\log_2 N$, e.g., for N=1000 from $10^6$ to $10^4$. Invention of FFT is often considered as one of the starting points of digital signal processing as a discipline of its own.

and instructions to be any multiple of 4 bits by multiplying the number of ICs. The Am2901 chip was the arithmetic-logic unit (ALU), and the "core" of the Am2900 series. It performed arithmetic and binary operations with 4 bit accuracy and could also implement various bit-shifting operations.

Based on this chip set a microprogrammable digital signal processor, named NRS 290, was designed. The design objective was to have a very flexible system in order to test and demonstrate the capabilities and limitations of signal processing algorithms and their hardware implementations. The letters NRS stand for the original design team: Neuvo, Ropponen, and Simula while numbers 290 stand for Am2900 circuit family.

## 2 NRS 290 Microprogrammable Digital Filter

The key design objectives of NRS 290 were high performance and ease of programming a variety of modular DSP algorithms. Building this kind of leading edge hardware in a university environment was expected to be challenging but also educational.

The NRS 290 hardware was able to implement a basic digital filter module which then could be multiplexed up to 16 parallel channels. The basic filter module was realized by a microprogram block which could be looped to implement, e.g., a higher order filter or a filter bank. The structure of the digital filter could be changed by changing the microprogram defining the realization form of the basic module. The filter parameters as well as the order and number of filters were stored in the data memory and registers, and could be externally controlled. The performance of NRS 290 was well comparable to the high-speed digital filters of that time.

### 2.1 Hardware Organization of NRS 290

The hardware organization and register configuration of NRS 290 digital filter are shown in Fig. 1. It followed the Harvard architecture principle where the instructions and data were physically separated. The microprogram memory and the data RAM were connected to separate control and data buses, respectively. The maximum number of input/output channels in the filter bank was 16. The word length of the filter was chosen to be 12 bits. This was a compromise between the computational speed and accuracy. However, for low-sensitivity filter structures the word length was satisfactory.

According to Fig. 1, the processor hardware was divided into four parts: Central processing element (CPE), memory (DATA RAM), multiplier, and external control unit. Internal communications as well as connections to AD- (ADC) and DA- (DAC) converters were performed via the 12 bit data bus. Arithmetic operations were carried out in the CPE ALU and in the multiplier. Intermediate results and constants were stored in the CPE RAM and in the DATA RAM.

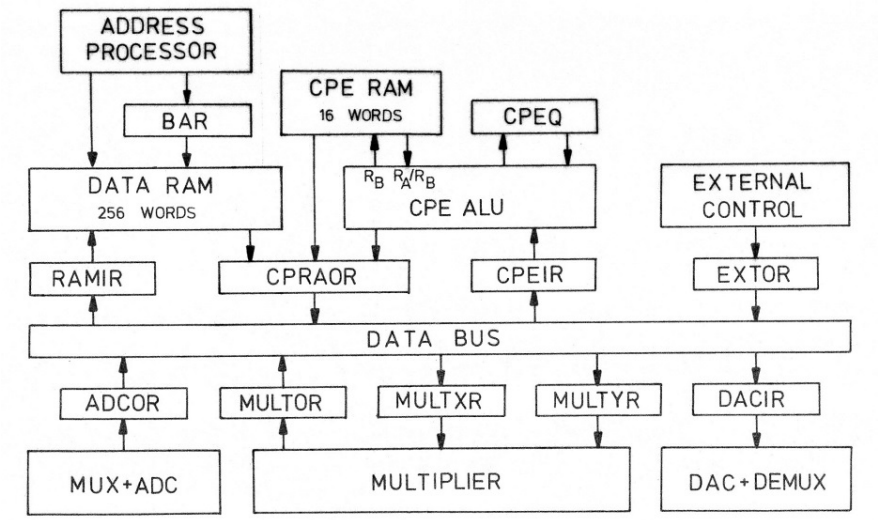


**Fig. 1.** Hardware organization and register configuration of NRS 290

The Am2901 microprocessor formed the CPE. The 12-bit word length required three 4-bit slices. The estimated maximum microinstruction cycle time was 230 ns due to the microprogram sequencer and control circuitry throughput. For computationally efficient multiplication a separate hardware array multiplier was included. The multiplier unit was built using Am25S05 4-by-2 bit two's complement multiplier circuits. The maximum multiply time for 12 bit numbers was about 200 ns which was roughly equal to the microinstruction cycle time.

The size of the data RAM was 256 words divided into 32 blocks of 8 words. Each block contained the parameters, including fixed filter coefficients, of one filter module, e.g., a second order block. The RAM addressing was performed directly under the microprogram control. The block address register, BAR, shown in Fig. 1 was implemented as a five bit up/down counter containing the address of one memory block. BAR could be incremented and decremented under the microprogram control. The memory location within a block was addressed directly from the microprogram with three bits of the microinstruction word. More details about the NRS 290 hardware can be found in [11].

The NRS 290 hardware consisted of six circuit boards of size 234 mm x 160 mm. They were: (1) Microprogram memory and sequencer, (2) CPE ALU and DATA RAM, (3) external control, (4) MUX + ADC, (5) DAC + DEMUX, and (6) multiplier. The circuit boards were constructed using the wire-wrap techniques with all circuits installed into sockets. The NRS 290 front panel and the multiplier circuit board are shown in Fig. 2.

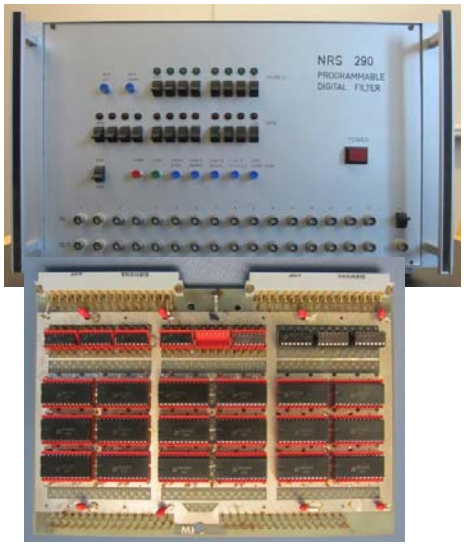


**Fig. 2.** NRS 290 front panel and the multiplier circuit board

### 2.2 Microinstruction word

The microinstruction word length of NRS 290 was 40 bits (including 2 free bits) shown in Fig. 3. All control signals required for the external devices were generated in the microprogram. These signals were completely independent of each other allowing the external devices operate simultaneously, if desired. The only restriction in the operation was that only one data transfer in the bus was possible during one microinstruction cycle. Therefore, the only encoded parts of the microinstruction word were the source and destination of the data bus. This allowed the control of all hardware modules independently, resulting in flexible implementations of filter structures.

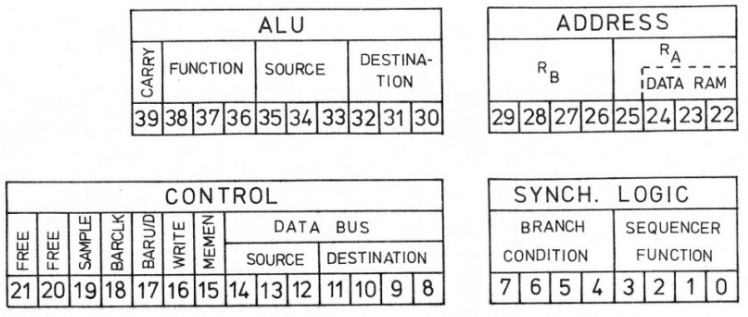


**Fig. 3.** Microinstruction word

The CPE RAM addresses ($R_A$ and $R_B$) as well as the DATA RAM address were also defined in the microprogram. To reduce the microinstruction word length the same bits were used to address the DATA RAM and to read from the CPE RAM ($R_A$). This did not affect the processing time because no register-to-register operations were normally needed simultaneously with the DATA RAM operations.

## 3 Programming

A special register transfer language (RTL) was developed to describe the operations executed by the microprogram [11]. The instruction set is defined in Table 1. RTL instructions considerably simplified the microprogramming which is shown in the examples below.

**Table 1.** Instruction set of the RTL language.

| Instruction | Operation |
|---|---|
| RMEM(N) | Read from N:th location of data RAM into CPRAOR |
| WMEM(N) | Write into N:th location of data RAM from RAMIR |
| DBTR(U→V,W) | Data bus transfer between registers specified by U, V and W |
| ADD(U,V→W,T) | Addition in CPE with source and destination registers specified by U, V, W, and T |
| SUB(U,V→W,T) | Subtraction in CPE with U, V, W, and T as above |
| MOV(U→V,W) | Move data between CPE registers and/or CPE RAM |
| SHRRB(N) | Right shift at CPE RAM through port $R_B$ (division by 2) |
| SHLRB(N) | Left shift at CPE RAM through port $R_B$ (multiplication by 2) |
| INC(U) | Increment register specified by U |
| DEC(U) | Decrement register specified by U |
| SAMPLE | Start AD- and DA-converters |
| JUMP I | Jump to I:th instruction |
| IF...ELSE... | Test and conditional jump |
| CONTINUE | Continue (no operation) |

The performance of the digital filter bank can be exemplified by the following RTL program. The program implements a cascade form realization of $2^{nd}$ order blocks in direct form II [2, p. 150]. The block diagram of the program is shown in Fig. 4. Cascading of $2^{nd}$ order blocks is realized in the loop from step 9 to 29. The instructions from 0 to 8 and 21 to 29 realize the same computations, the only difference being the source of the input signal, i.e., whether it is the AD-converter or the preceding filter block. Scaling is performed with one scaling coefficient and a shift operation. The parameters of a $2^{nd}$ order digital filter block in direct form II are shown in Fig. 5.

The RTL program corresponding to the block diagram of Fig. 4 is given in Fig. 6 below. From the program it can be seen that the data bus really is busy most of the computation time. About 77% of the microinstructions are performing data bus transfers. The corresponding load percentages for the CPE and DATA RAM are only 43% and 47%, respectively.

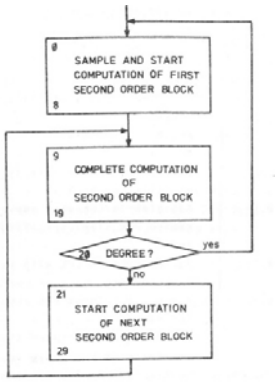


**Fig. 4.** Block diagram of the program

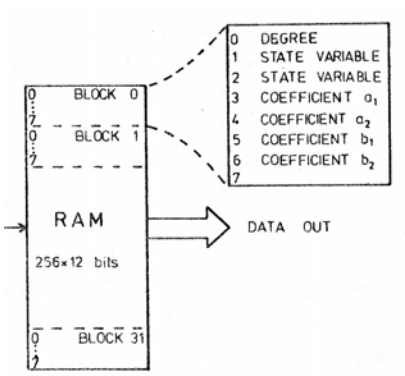


**Fig. 5.** Contents of the memory block

```
Ø   DBTR (ADCOR → MULTXR), RMEM (3)
1   DBTR (CPRAOR → MULTYR), MOV (CPEQ → CPRAOR)
2   DBTR (CPRAOR → DACIR)
3   DBTR (MULTOR → CPEIR), RMEM (2)
4   DBTR (CPRAOR → MULTXR) MOV (CPEIR → CPEQ), RMEM (7)
5   DBTR (CPRAOR → MULTYR), RMEM (Ø)
6   DBTR (CPRAOR → CPEIR)
7   DBTR (MULTOR → CPEIR), MOV (CPEIR → RB (Ø)), RMEM (5)
8   DBTR (CPRAOR → MULTYR), ADD (CPEIR, CPEQ → CPEQ), SAMPLE
9   CONTINUE
1Ø  DBTR (MULTOR → CPEIR), RMEM (1)
11  DBTR (CPRAOR → MULTXR, RAMIR), MOV (CPEIR → RB (1)), RMEM (6)
12  DBTR (CPRAOR → MULTYR), WMEM (2)
13  CONTINUE
14  DBTR (MULTOR → CPEIR), RMEM (4)
15  DBTR (CPRAOR → MULTYR), ADD (CPEIR, CPEQ → RB (2)), SHLRB (2)
16  ADD (CPEIR, CPEQ → CPEQ)
17  DBTR (MULTOR → CPEIR), MOV (RB (2) → CPRAOR)
18  DBTR (CPRAOR → RAMIR), ADD (CPEIR, RA (1) → RB (1)), WMEM (1)
19  DEC (RB (Ø)), DEC (BAR)
2Ø  ADD (RA (1), CPEQ → CPEQ, CPRAOR), IF (RB (Ø)= Ø) JUMP Ø
21  DBTR (CPRAOR → MULTXR), RMEM (3)
22  DBTR (CPRAOR → MULTYR)
23  CONTINUE
24  DBTR (MULTOR → CPEIR), RMEM (2)
25  DBTR (CPRAOR → MULTXR), MOV (CPEIR → CPEQ), RMEM (7)
26  DBTR (CPRAOR → MULTYR)
27  CONTINUE
28  DBTR (MULTOR → CPEIR), RMEM (5)
29  DBTR (CPRAOR → MULTYR), ADD (CPEIR, CPEQ → CPEQ), JUMP 9
```

**Fig. 6.** RTL program of an example filter

## 4 Performance Measurements

The performance measurements clearly showed that the computation speed of NRS 290 was on the level of fastest digital filter implementations of its time. The initial design specifications of NRS 290 were fully reached and even in some respect exceeded. The measured microinstruction execution time was 195 ns which was 35 ns less than originally estimated. The performance of different digital filter implementations was compared on the basis of attenuation and computation time. However, the computation time was strongly dependent of the filter type, degree, and realization form.

For comparison, the number of microinstructions as a function of degree is shown in Table 2 for different filter structures. The traditional 2nd order block realizations and low-sensitivity digital lattice forms were compared. The computation times and the corresponding maximum sampling rates can be calculated using the measured microinstruction cycle time 195 ns. Details of all comparisons can be found in [12].

Using the microinstruction cycle time of 195 ns, the computation times of 4.1 μs for a standard 2nd order digital filter block (row 1 in Table 2) was obtained. Similarly, for a full filter bank of 16 4th order filters the computation time of 131 μs was achieved. The corresponding sampling rates were 244 kHz and 7.63 kHz, respectively.

**Table 2.** The number of microinstructions as a function of degree in different filter programs.

| transfer function | filter structure | number of micro-instructions |
|---|---|---|
| general $\frac{N(z)}{D(z)}$ | 2nd order block<br>digital lattice | $21\frac{n}{2}$ (n even)<br>11n + 8 |
| all-pole $\frac{K}{D(z)}$ | 2nd order block<br>digital lattice | 8 n (n even)<br>10n + 1 |
| one-zero lowpass $K\frac{(z+1)}{D(z)}$ | 2nd order block<br>digital lattice | 8 n (n even)<br>10n + 2 |
| two-zero lowpass $K\frac{(z^2+1)}{D(z)}$ | 2nd order block<br>digital lattice | 8 n (n even)<br>10n + 3 |
| two-zero bandpass $K\frac{z^2-1}{D(z)}$ | 2nd order block<br>digital lattice | 8 n (n even)<br>10n + 3 |

To make comparisons easier, a figure of merit was introduced to measure the attenuation efficiency, $\eta_A$

$$\eta_A = \frac{\text{Attenuation in dB}}{\text{Number of microinstructions}} \tag{1}$$

In the above equation, attenuation was measured at the stopband edge frequency. For bandpass filters the minimum attenuation at the edge frequencies was used. Table 3 shows the measured computation times and attenuation efficiencies of different filter structures. A 4th order elliptic filter was used as a reference structure. It was compared with both 4th and 6th order filters with reduced number of zeroes but equal passband specifications. It can be seen that the attenuation is slightly increased in the realizations with reduced number of zeroes with some increase in computation time, too. However, the attenuation efficiency remains about the same.

**Table 3.** Comparison of filter implementations.

| Transfer function | Degree | Stopband edge attenuation (dB) $\omega_{1s} = 0.2\ \pi$ | $\omega_{2s} = 0.54\ \pi$ | Filter structure | Computation time (μs) | Attenuation efficiency |
|---|---|---|---|---|---|---|
| $\frac{N(z)}{D(z)}$ | 4 | 28.8 | 28.8 | 2nd order block<br>digital lattice | 8.19<br>10.14 | 0.69<br>0.55 |
| $\frac{K}{D(z)}$ | 6 | 29.9 | 42.7 | 2nd order block<br>digital lattice | 9.36<br>11.90 | 0.63<br>0.49 |
| $\frac{K(z^2-1)}{D(z)}$ | 6 | 33.3 | 41.7 | 2nd order block<br>digital lattice | 9.36<br>12.29 | 0.69<br>0.53 |

In Fig. 7, the measured amplitude response of a 4th order elliptic bandpass filter realized using all-pole 2nd order blocks is shown. The passband ripple is 1 dB and the center frequency 13.2 kHz with lower and upper cutoff frequencies 12.7 kHz and 13.6 kHz, respectively. The sampling frequency of 75.2 kHz was used.

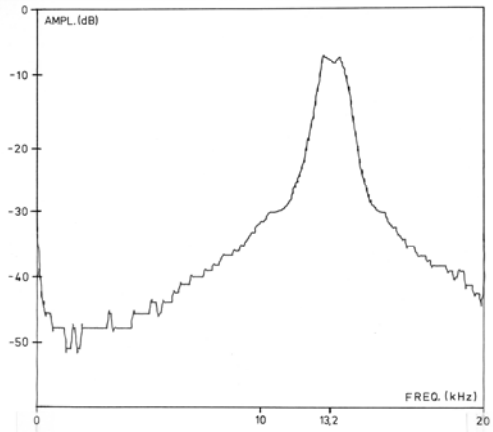


**Fig. 7.** Measured amplitude response of a 4th order all-pole bandpass filter

The computation speed of NRS 290 can be compared with similar hardware architectures of late 1970s. In 1980 a signal processor architecture based on distributed arithmetic (SDA) was reported in [13]. The SDA hardware built at ETH Zurich was also based on Am2900 circuit family. It was microprogrammable and had 16 bit word length using, thus, 4 Am2901 bit-slices. The basic cycle time of the SDA architecture was 300 ns. Another TTL based implementation of a 4800 bit/s microprocessor data modem was reported already in 1978 in [14]. The microprogrammable hardware was also using bit-sliced microprocessors of 4 bits. With 12 bit word length the cycle time of 230 ns was achieved. Thus, it can be seen that the computation speed of NRS 290 hardware was very well comparable, or even better than that of similar implementations based on the latest microprocessor technology of 1970s.

## 5 Conclusions

In the 1970s, digital signal processing was rapidly maturing for real time applications. Although the design of filters was laborious due to the lack of CAD programs, digital filters were designed for practical applications. Also the implementation was difficult because there were no chips or chip sets tailored for DSP. The development was, however, fast. Ironically, as soon as the NRS 290 hardware multiplier unit had been built and tested, TRW released a 16 bit multiplier on a single chip. Similar progress was also seen in the DSP software development tools in the 1980s.

In Finland, the pioneering work in DSP was very successful. The telecommunications industry was especially enthusiastic of modern DSP implementations. One of the first industrial applications was started in the middle of 1970s. A state-owned telecommunications company Televa, later part of Nokia Telecommunications, was investigating digital filter implementations for detecting touch tone dialing signals. The filter bank was designed as a master's thesis project at TKK. At that time digital filter design tools did not exist. However, the designed filters worked properly and were computationally extremely efficient.

This kind of early research cooperation with industry was very important for the future development of the telecommunications and mobile phone industry. Nokia introduced the world's first GSM handset Nokia 1011 in early 1990's. At that time commercial 16 bit signal processors with hardware multiplier were available. In the beginning the GSM speech coder needed two signal processor chips in parallel in order to meet the real time processing requirements. Nokia in partnership with Texas Instruments had a strong role in pushing digital signal processing first to speech coding and later to many other demanding mobile applications including image and video compression. Nokia was thus one of the leading companies utilizing signal processors in the mobile phones starting from 2G. Much of the signal processing expertise required can be traced back to the research and education activities where NRS 290 had played an important role.

## References


1. Bernard Gold and Charles M. Rader, Digital Processing of Signals, McGraw-Hill, 1969
2. Alan V. Oppenheim and Ronald W. Schafer, Digital Signal Processing, Prentice-Hall, 1970
3. Sven Eriksson and Lars Wanhammar*:* Tidsdiskreta filter, del 1 - 3, 1978
4. Y. Neuvo, O.Simula, and L. Vainio, MUCROMAF and MUCROER Polynomials in Digital Filter Design, IEEE Trans. on Circuit Theory, vol. CAS-24, pp. 151-154, March 1977
5. A. Premoli, The MUCROMAF Polynomials: An Approach to the Maximally Flat Approximation of RC Active Filters with Low Sensitivity, IEEE Trans. on Circuits and Systems, vol. CT-20, pp. 77-80, Jan. 1973
6. A. Premoli, A New Class of Equal-Ripple Filtering Functions with Low Q-Factors: The MUCROER Polynomials, IEEE Trans. on Circuits and Systems, vol. CAS 21, pp. 609-613, Sept. 1974
7. A.H. Gray and J.D. Markel, Digital Lattice and Ladder Filter Synthesis, IEEE Trans. on Audio and Electroacoustics, vol. AU-21, pp. 491-500, Dec. 1973
8. Y. Neuvo and O. Simula, Coefficient Sensitivities of Cascaded Lattices at Zero Frequency, IEEE Trans. on Acoustics, Speech, and Signal Processing, vol. ASSP-24, pp. 336-337, Aug. 1976
9. Y. Neuvo and O. Simula, An Iterative Method for Designing Digital Filters with Reduced Number of Zeroes, Proc. 1976 European Conference on Circuit Theory and Design, Genova, Italy, Sept. 7-10, 1976, pp. 332-339
10. "The Am2900 Family Data Book with Related Support Circuits". *AM-PUB003*. Advanced Micro Devices. 1979. Archived from the original on 16 July 2011. Retrieved July 11, 2011.
11. Y. Neuvo, K. Ropponen, and O. Simula, A Fast Microprogrammable Digital Filter Design, Proc. 1977 IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP-77), Hartford, May 1977, pp. 523-526
12. P. Bärling, Y. Neuvo, and O. Simula, Microprogrammable Digital Filter, Programming and Performance, Proc. Fourth International Symposium on Network Theory, Ljubljana, Yugoslavia, Sept. 4-7, 1979, pp. 391-396
13. J.Zeman and H.T. Nagle, A High-Speed Microprogrammable Digital Signal Processor Employing Distributed Arithmetic, IEEE Trans. on Computers, vol. C-29, pp. 134-144, Feb. 1980
14. K. Watanabe, K. Inoue, and T. Sato, A 4800 bit/s Microprocessor Data Modem, IEEE Trans. on Communications, vol. COM-26, pp. 493-498, May 1978