Andrej Magyar; Arild Lacroix

Digital signal processors

Kybernetika, Vol. 26 (1990), No. 3, 206--220

Persistent URL: http://dml.cz/dmlcz/125573

DIGITAL SIGNAL PROCESSORS

ANDREJ MAGYAR, ARILD LACROIX

The paper deals with single chip programmable digital signal processors, which are widely used for real time implementation of various signal processing algorithms. After short introduction a classification of signal processors into three generations is provided and some typical architectures presented. Then the processors are compared according to common signal processing benchmarks. It is followed with a discussion on the future trends in the development of signal processors.

1. INTRODUCTION

About 20 years have elapsed since the introduction of the first microprocessor (the i4004) and roughly a decade since the appearance of the first digital signal processor (the i2920). Both chips were introduced by Intel. Although comparing them by present standards, they may seem quite primitive, each marked the beginning of an evolutionary cycle.

The four bit i4004 started the era of general purpose microprocessors while the i2920 gave rise to a class of so called general purpose programmable single-chip digital signal processors (DSPs). The latter group is of a primary interest of this paper and will be discussed in more detail.

DSPs have specialized architecture fitting the requirements of signal processing algorithms. In fact they are reduced-instruction-set computers optimized for the fastest possible execution of addition, subtraction, multiplication and shifting instructions. These operations constitute the base of most signal processing algorithms. The need of these operations is evident e.g. from a finite impulse response digital filter (FIR), whose structure is shown in Fig. 1.

The implementation of this filter in each tap requires: (1) fetching the instruction, (2) fetching two operands from memory, (3) multiplying, (4) accumulating, and (5) shifting data in the delay line. All modern DSPs can implement these operations in a single instruction cycle which results in one FIR tap per cycle.

DSPs possess parallel architecture, fast ALU, relatively large on-chip memory, reduced but powerful instruction set, bus oriented I/O, low power consumption and short instruction cycle. They allow cost-effective implementation of even complex signal processing algorithms with minimum number of external components. Because of these features, DSPs offer a good base for real-time application of signal processing algorithms. Some application examples include digital filtering, fast Fourier transforms, digital audio, speech and image processing etc. [1].

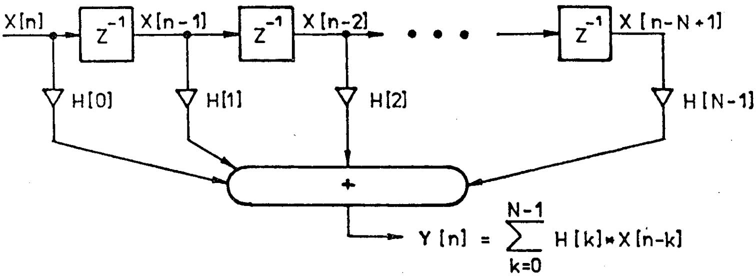

Fig. 1. The structure of FIR filter.

On the other hand DSPs have some common features which make them different from the general purpose microprocessors. First, their architecture fits the requirements of signal processing algorithms. Second, DSPs have powerful 16 to 32 bits wide ALU supported with high speed single-cycle hardware multiplier. Third, they have reduced but powerful instruction set. In one instruction cycle they can execute several operations e.g. multiply two operands, accumulate the previous product, move operands to registers or memory, modify counters etc. Fourth, they have short instruction cycle with period less than 100 ns for the newer generation of DSPs.

2. CLASSIFICATION OF DSPs

Since the introduction of the first DSP, the i2920 in 1979 by Intel, a rapid development has occurred in this field. Their complexity and speed has increased substantially. The present DSPs have achieved the complexity of 700 000 transistors on a chip compared to 18 000 for the i2920. The computational power of newer DSPs is comparable with that of array processors in 1979 [2]. However, the power consumption and size are incomparable.

Todays more than 20 DSPs are announced by many manufacturers all over the world. They feature different parameters from which the most important are the following:

– arithmetic type (fixed/floating point),

– memory capacity,

– instruction cycle time,
– complexity (number of transistors and pins).

The first parameter, arithmetic type, is very important from the viewpoint of precision and dynamic range. First and second generation DSPs usually included a fixed-point arithmetic-logic unit (ALU) working with 16 or 24 bits wide operands. Besides the low dynamic range of fixed-point DSPs it is necessary to carefully scale the signals in order to avoid the overflow problems. For this a thorough simulation of the algorithm should be done before its implementation on the selected DSP. The overflow problem is not so severe on the third generation DSPs, which exclusively include a floating-point ALU. They are usually working in 32 bit format where 24 bits are assigned for the mantissa and 8 bits for the exponent. This format essentially simplifies the programming of DSPs and gives enough precision and dynamic range.

The second parameter, memory capacity, in fact limits the implementation of memory intensive algorithms like e.g. discrete transforms. Most DSPs use at least three on-chip memory blocks: instruction, data and coefficients. All of them can cause limitations. If we consider e.g. an often used 1024 points complex FFT, large data block is to be stored in data memory. However for better performance it is preferred also to store the required coefficients. As a result the FFT size N is limited by on chip-data and coefficient memories. On the other hand the Winograd-Fourier Transform Algorithm, for its complicated internal structure, needs a lot of instructions and therefore a large program memory. The memory limitation problem is effectively solved by connecting additional off-chip memories.

The third parameter, instruction cycle time, depends on the technology used and the processor architecture, especially with respect to pipelining. Apart from the complexity the newer third generation DSPs have instruction cycle times from 60 to 80 ns compared to 400 ns for the first DSP. The instruction cycle time essentially influences the achieved sampling rate.

Table 1. Range of DSP parameters.

Selected DSP parameters		DSP generation		
		First	Second	Third
Arithmetic type		INT	INT	FLP
Memory	Internal	1·5k	4k	4k
	External	4k	16k–64k	>64k
Instruction cycle [ns]		>200	100–200	<150
Complexity	No. of transistors	<70 000	150 000	>300 000
	Number of pins	28–40	40–68	>68

Note: INT = fixed-point, FLP = floating-point arithmetic

The final parameter, complexity, expressed in number of on-chip transistors and pins gives an idea of the processor's architecture and the solution of I/O interfaces. The number of transistors is a result of ALU's complexity, the internal memory capacity etc. On the other hand, the number of pins can be considered as a measure of the DSP's capability for connections to the outside world. It is in close relation to the multiplexing of data and address busses for off-chip memories.

According to the above discussed parameters, the range of which is given in Table 1, it is possible to classify the available DSPs into three generations, the discussion of which follows.

a) First generation DSPs

The era of first generation DSPs is roughly from 1979 to 1982. Table 2 gives a review of some typical parameters of representative DSPs.

The i2920 from Intel, announced in 1979 [3], was designed for digital processing

Table 2. Parameters of first generation DSPs.

Parameter			Company/Processor			
			Intel	NEC	NEC	TI
			i2920/21	µPD7720	µPD77C25	TMS32010
Announced in			1979	1980	1987	1982
Instruction cycle [ns]			400	250	122	200
Data path:	Integer/Fl. point		INT	INT	INT	INT
	wordlength [bits]		25	16	16	16
Multiplier organization			—	16×16→31	16×16→31	16×16→32
Internal	Program		192 × 24	512 × 23	2k × 24	1536 × 16
memory	Coeff.		—	512 × 13	1k × 16	
(bits)	Data		40	128 × 16	256 × 16	144 × 16
External	Program		—	—	—	4k × 16
memory	Data		—	—	—	—
Interface	Parallel	Address	—	—	—	(1) 12-bit
	bus	Data	—	(1) 8-bit	(1) 8-bit	(1) 16-bit
	Serial	Input	4 Analog	1	1	—
	I/0	Output	8 Analog	1	1	—
Technology			6µ-NMOS	3µ-NMOS	1·6µ-CMOS	3µ-NMOS
Max. power consumption [mW]			800	900	200	900
Complexity	chip-area [mm^2]		30·38	28·44	—	43·87
	No. of elements		18 000	40 000	44 000	55 000
	Package		28-DIP	28-DIP	28-DIP	40-DIP

of analog signals. It included on-chip 9-bit A/D and D/A converters, but a fast hardware multiplier was missing. Because of limited on-chip RAM memory (40 words) and relatively long instruction cycle (400 ns) only simple algorithms could be implemented on it. Nevertheless this processor has become a pioneer in the development of DSPs.

The μPD7720 from NEC [4], was the first widely used DSP. Its parallel Harvard architecture is still being updated today (μPD77C25) making it the most durable single-chip DSP architecture. In contrary to the i 2920, by including the hardware multiplier into the data path, multiply/accumulate operations can be executed in single instruction cycle. The processor is designed both for stand-alone applications and as a signal processing interface to commercially available microprocessors. Two versions are available from the manufacturer: 1) mask-ROM version for high volume applications and 2) user programmable EPROM version for algorithm development purposes.

The μPD77C25 [5] is a slightly modified version of its predecessor the 7720. Both pin and functional compatibility of processors are maintained. As follows from Table 2, the 77C25's instruction cycle is halved and the on-chip memory capacity increased.

These processors have a closed architecture excluding the possibility to expand internal memories by connecting external one.

The TMS32010 from Texas Instruments [6], is a first member of very popular TMS 320 family of DSPs. Compared to 7720 it has 3 times larger on-chip program memory, which in addition can be expanded by off-chip RAM or EPROM. As a result, program development is simpler without a need of expensive in-circuit hardware emulator. The multiply/accumulate operations require two instruction cycles but the product can be accumulated in 32-bit wide ALU.

b) Second generation DSPs

The second generation DSPs in comparison with the first generation have a few new features:

- shorter instruction cycle (approx. 100 ns),
- larger internal memory capacity,
- possibility of off-chip memory expansion,
- low power consumption,
- simple and fast I/0 interface.

In the Table 3 parameters of four second generation DSPs are summarized.

The μPD77220 [7] is a continuation in NEC's family. It is designed with 24-bit fixed point ALU including a hardware multiplier and equipped with four on-chip memory blocks. If necessary, program and data memories can be expanded off-chip. The processor can use two data memory blocks simultaneously which are addressed by individual address generators.

Table 3. Parameters of second generation DSPs.

Parameter			Company/Processor			
			NEC µPD77220	TI TMS320C25	AT & T DSP16A	Motorola DSP56000
Announced in			1985	1985	1988	1987
Instruction cycle [ns]			100	100	33	98
Data path:	Integer/Fl. point		INT	INT	INT	INT
	wordlength [bits]		24	16	16	24
Multiplier organization			24×24→48	16×16→32	16×16→32	24×24→56
Internal memory (bits)	Program		2k × 32	4k × 16	4k × 16	2k × 24
	Coeff.		1k × 24	256 × 16	—	(2) 256 × 24
	Data		512 × 24	288 × 16	2k × 16	(2) 256 × 24
External memory	Program		4k × 24	64k × 16	64k × 16	64k × 24
	Data		4k × 24	64k × 16	—	128k × 24
Interface	Parallel bus	Address	(1) 13-bit	(1) 16-bit	(1) 16-bit	(2) 16-bit
		Data	(1) 32-bit	(1) 16-bit	(1) 16-bit	(1) 24-bit
	Serial I/0	Input	1	1	1	1
		Output	1	1	1	1
Technology			1·7µ-CMOS	1·8µ-CMOS	1µ-CMOS	1·5-CMOS
Max. power consumption [mW]			700	500	500	900
Complexity	chip-area [mm^2]		—	—	—	—
	No. of elements		200 000	150 000	—	—
	Package		68-PGA	68-PGA	84-PLCC	84-PGA

Another processor of this generation of DSPs is the TMS32C25 [8, 9], a very powerful and successful member of TMS320 family. In addition to single-cycle multiply/accumulate operation it offers 32 bit wide accumulation and external memory expansion up to 64 kwords.

Besides the traditional DSP suppliers like NEC and TI there are two newcomers: AT & T Bell and Motorola. They introduced two new DSPs, which according to their parameters, belong to the second generation.

The DSP16A from AT & T [10], has an extremely short multiply/accumulate cycle (only 33 ns) and uses a 36-bit wide accumulator.

The DSP56000, from Motorola [11], uses 24-bit fixed-point data format however the accumulation is executed in a 56-bits wide ALU. This provides high-precision arithmetic sufficient for most real-time applications. The processor can accumulate 256 successive full precision products without overflow. The processor uses five memories organized in three blocks: one program and two data. A special address generation unit is used for data memory blocks. It can implement linear, modulo and reverse-carry addressing.

c) Third generation DSPs

All third generation DSPs are using 32-bit floating-point data format with 24-bit mantissa and 8-bit exponent. In Table 4, the parameters of 4 representative DSPs are listed although more processors have already been announced.

The μPD77230 [12] was the first representative of this generation. Internally the processor is working with NEC's floating-point format which can be converted into the IEEE-754 format in a single instruction if necessary. The processor uses four memories: one for program, two for data and one for coefficients. By adding external

Table 4. Parameters of third generation DSPs.

Parameter			Company/Processor			
			NEC μPD77230	TI TMS320C30	AT & T DSP32C	Motorola DSP96002
Announced in			1986	1987	1988	1988
Instruction cycle [ns]			150	60	80	74
Data path:	Integer/Fl. point		FLP	FLP	FLP	FLP
	wordlength [bits]		32	32	32	32
Multiplier organization			32× 32→55	32× 32→40	32× 32→40	32× 32→56
Internal	Program		2k × 32	4k × 32	2k × 32	1k × 32
memory	Coeff.		1k × 32	–	–	(2) 512 × 32
(bits)	Data		(2) 512 × 32	(2) 1k × 32	(2) 512 × 32	(2) 512 × 32
External	Program		4k × 32	16M × 32	4M × 32	4G × 32
memory	Data		4k × 32	16M × 32	–	(2) 4G × 32
Interface	Parallel	Address	(1) 13-bit	(2) 24-bit	(1) 22-bit	(2) 32-bit
	bus	Data	(1) 32-bit	(2) 32-bit	(1) 32-bit	(2) 32-bit
	Serial	Input	1	1	1	1
	I/0	Output	1	1	1	1
Technology			1·7μ-CMOS	1μ-CMOS	·75μ-CMOS	1μ-CMOS
Max. power consumption [mW]			1700	500	400	900
Complexity	chip-area [mm^2]		130	400	88	–
	No. of elements		370 000	700 000	405 000	–
	Package		68-PGA	180-PGA	133-PGA	84-PGA

memories the program and data memories can be expanded. The products from the multiplier can be accumulated with full precision in the 55-bits wide floating-point ALU.

The TMS320C30 [13] is the most powerful DSP in the TMS320 family. In addition to the floating point ALU including multiplier, the processor has a few new features. First, it can operate either from the on-chip program ROM (microcomputer mode) or from the external memory by use of the program cache (microprocessor mode). The use of cache allows to connect slower external memories without degrading the processor's performance. Second, by use of seven internal buses, high throughput is achieved with instruction cycle time of 60 ns. Third, the processor can address 16-Mwords of external memory, a capacity sufficient even in image processing applications.

The remaining processors were announced in 1988 and they are qualitative upgrade of their second generation predecessors.

The DSP32C from AT & T [14] is an improved version of its predecessor the DSP32. It can perform either 24 bit integer operations or 32 bit floating point operations at a rate of 25 million per second. The processor uses very flexible memory configuration and provides simple interface to external devices. Three on-chip DMA controllers support direct, independent memory access via the serial input, serial output and parallel I/0 ports.

The DSP96002 from Motorola [16, 17] is the only DSP fully supporting the ANSI/IEEE 754-1985 Standard for Binary arithmetic [18]. It retains the hardware and software features available on its fixed-point counterpart, the DSP56000. The processor is designed for parallel processing in which multiple 96002's share buses and communicate directly with each other. Because of its dual memory expansion ports, two external memories can be accessed concurrently. A unique feature included in the 96002 is a dedicated serial port with on-chip circuit emulation and debugging capability. It provides simple program development and eliminates the need of an expensive hardware emulator.

3. PERFORMANCES AND LIMITS OF DSPs

In looking for a proper DSP it is necessary to have an idea about performances achieved by the available DSP generations.

a) DSP performances

Although the third generation floating-point DSPs can implement very complex algorithms in real-time, their main application area is still limited to audiofrequency range. Figure 2 summarizes the sample rate limits for three generations of the popular TMS320 DSP family.

The vertical axis gives the number of instruction cycles per sample period against the sample frequency on the horizontal axis. It is apparent that by introducing of each new DSP generation the sample rate is increased. However, even the presently existing most powerful DSPs cannot achieve the video sample rate. For these applications either multi-DSP structures or nonconventional architectures like the NEC's μPD7281 data-flow signal processor [19] designed for image processing will be used.

Figure 2 is complemented by Table 5. It gives execution times of some often used signal processing algorithms on TMS320 DSP family.

Many attempts have been made to compare the performances of existing DSPs

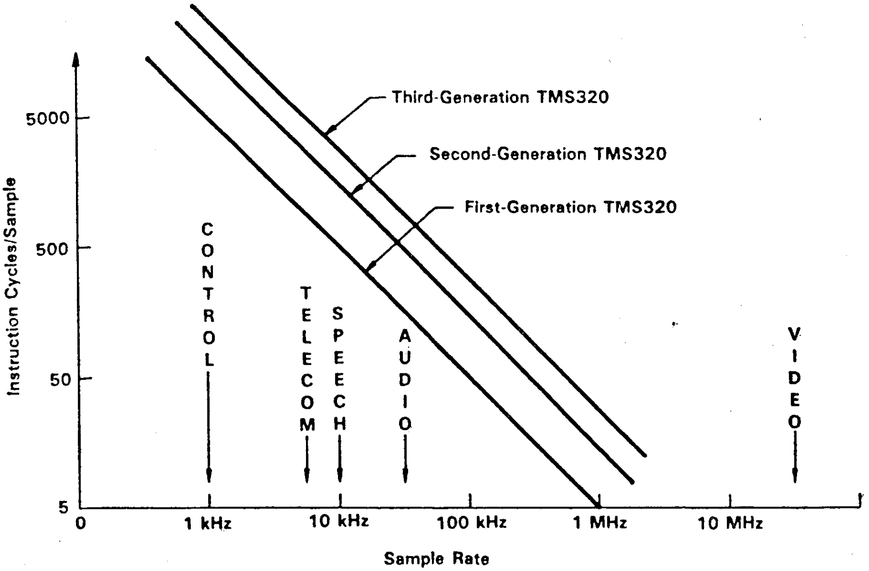

Fig. 2. Sample rate limits for DSP generations.

Table 5. Execution times of signal processing algorithms.

DSP Routines/Applications	DSP Generation		
	First	Second	Third
FFT radix 2, 1024 complex points	33 ms	9·1 ms	3·7 ms
FIR filter: 1-tap	400 ns	100 ns	60 ns
FIR filter: 256-taps (sample rate)	9·25 kHz	37 kHz	>60 kHz
LMS adaptive FIR: 1-tap	700 ns	400 ns	180 ns
LMS adaptive FIR: 256-taps (sample rate)	5·4 kHz	9·5 kHz	>20 kHz
IIR filter: 2-nd order	2 μs	1 μs	360 ns
IIR filter: 8-th order (sample rate)	140 kHz	285 kHz	660 kHz
Echo canceller (single chip)	8 ms	32 ms	>64 ms

[20–25]. It is not an easy task to qualify the available DSPs because of their architectural differences, circuit technology etc. The interested reader can find a thorough comparison of 18 DSPs on the base of 12 selected signal processing benchmarks in reference [26]. Even in this reference it is not declared which DSP is the best and which the worst. The reason is simple: it is application dependent. Therefore for a specific application it is important to select a proper DSP in accordance with the requirements of the implemented algorithm especially as to the precision, memory capacity, sampling rate, price, production volume etc.

b) Software limitations

The majority of first and second generation DSPs have to be programmed in their assembly languages. Although very efficient programs can be written in assembler a huge effort of an experienced programmer is necessary to do so. For this, it takes a relatively long time to convert the algorithm into the assembler program. This fact is one of the main reasons why the first and some second generation DSPs did not become an application standards.

Fortunately, the situation has been changed by introducing the C compilers for DSP families from Texas, AT & T and Motorola [27]. Apart from the efficiency of C compilers the algorithm translation into the selected DSP is fast and without the necessity of being familiar with the assembly language. This can largely contribute to widespread use of DSPs.

Another luck in software domain are simulators with multiprocessor simulation capability. The only exception in this respect is Motorola.

c) Hardware limitations

Irrespective of high chip density (about 700 000 transistors) and fast instruction cycle (60 ns) the present DSPs cannot process videofrequency range signals. The speed limitation is connected with the gate speed and propagation delays within a chip.

To fulfil the requirements of video-range signals principal changes can be expected both in DSPs' architecture and technology used. The future DSPs should be faster and they should include multiprocessing features.

Some solutions of the above mentioned limitations will be discussed in the next section.

4. FUTURE TRENDS

According to the development of programmable DSPs in the last ten years it is possible to extrapolate the current trends and predict their future development. We can expect main improvements in the following areas: VLSI, architecture, software and semi-custom design.

a) VLSI

As a result of advances in VLSI technology we can expect increased complexity of DSPs with more on-chip memory, faster multiply/accumulate cycle times, more I/0 flexibility, higher precision etc. This trend, however has severe limitations in the chip complexity which has reached the density of 700 000 transistors on a single chip with the available 1 μ-CMOS technology. Too complex chips need complicated testing procedures and in addition their price is high. It is in contradiction with the existing market which requires low-priced DSPs capable of processing signals at higher rates. Even with the introduction of better technologies, e.g. based on gallium arsenide, the classical von-Neumann type architecture is basically limited by sequential instruction execution. To overcome it, some new ideas in the architecture have to be introduced toward parallel processing.

b) Architectural improvements

A simple way to increase the computation power of the existing DSPs is to configure them in a multiprocessor system [28]. However, new features in hardware or software are necessary to synchronize processors in accessing shared resources, e.g. memory. In this case a controllable wait-state is invaluable. Some newer DSPs have pins that can be tested in software and can be used to synchronize multiple processors. The TMS320C30 has specialized instruction for doing it. Motorola facilitates the design of multiprocessor systems with the dual expansion ports in the DSP96002.

All of these are small steps only. One essential tool which is missing is a software simulator capable of simulating multiple DSP systems. Motorola is the only exception in this respect because its simulator supports multi-DSP simulation.

Another way to parallel processing is the use of processors structured as a simply connected replications of carefully optimized modules at the chip and sub-chip levels [29]. This approach promises lower chip complexity and higher sampling rates, but on the other hand, presupposes algorithms optimized for parallel processing. In this respect the existing algorithms optimized for single processor implementation have to be redesigned [30].

A more radical approach to parallel DSPs has been proposed by NEC in 1984 with the introduction of the μPD7281, a data flow machine for image processing [19]. This chip seems to be ahead of its time. Data flow machines are very promising in signal processing and a considerable theory has been developed about them [31, 32].

c) Software

One of the main impediments of widespread use of DSPs is their difficult programming compared to other microprocessors. Although the final code of the implemented algorithm can be stored in less than 1 K words of program memory, it takes months to write efficient programs. The reason is that for the majority of DSPs the only way for programming them is in assembler language. Writing efficient assembler programs assumes experienced programmers with thorough knowledge of both the DSP's architecture and the structure of the algorithm.

Fortunately for newer processors some C compilers have appeared [27]. These are the full implementation of the popular Kerninghan & Ritchie's C language. By the introduction of compilers the user can program in a high-level language though he can still include in-line assembly codes to increase the program's efficiency. Especially the optimizing C compilers of the third generation of floating-point DSPs are promising both in efficiency and easy handling [33 – 35].

The introduction of compilers is only the first step on the way of facilitating the programming of signal processing systems using DSPs. Higher level design environments are being constructed to permit rapid algorithm development and efficient code generation for DSPs [36]. The most promising systems under development are based on block-diagram programming, in which the user graphically constructs the block diagram of the algorithm. Such a programming environment named Gabriel is developed at Berkeley [37, 38].

There are two reasons for block-diagram approach. First, it is a natural description of many signal processing algorithms. Second, these can be automatically partitioned for the execution on parallel processors [37]. The user does not need to know the detail of the architecture, or even the type of DSPs. Block-diagram languages fit the data-flow model of computation [31].

d) Semi-custom processors

The complex general-purpose DSPs offer a convenient tool for real-time simulation of different algorithms. But, when the simulation is ready, the error-free code should be transferred into a mask programmed DSP which can be considered as an application specific IC. For many algorithms, however, not all of the features offered by DSPs are fully utilized. In such a case it would be useful to user customize the DSP by eliminating parts that are not used. Possibilities of customizing include:

- adapt the arithmetic word length to what is actually needed,
- remove the multiplier for low-speed applications,
- customize the size of all memories (for program, data and coefficients),
- eliminate unused addressing modes etc.

A user would develop the algorithm using a high level description such as C, a block-diagram language or some other language, and given a real-time constraint, a compiler would automatically determine the required architecture parameters. From them an automated layout program could generate layouts. Such a system called Lager is under development at Berkeley [39].

Another approach is to utilize the existing VLSI design tools for designing algorithm or application specific processors. They usually execute a single algorithm at high sampling rates e.g. the FFT computation [40–42], FIR filtering [43], Wigner distribution [44] etc. A wide use of application specific processors can be expected in communications terminal equipment, high speed data modems, acoustic echo cancellers, speech analysis, synthesis and recognition [29].

5. CONCLUSIONS

It is impossible within the limited space to give a complete discussion of all existing DSPs. In addition to the above mentioned DSPs there are more excellent processors e.g. the ADSP-2100 from Analog Devices [45, 46] or the ZR35325 from Zoran [45].

Instead of giving a detailed description of DSPs we have tried to give the reader an idea about the present situation, performances, limits and future trends.

We believe, the evolutionary process in this area will continue and DSPs will essentially contribute to the widespread use of digital signal processing methods in real-time applications.

(Received September 29, 1989.)

REFERENCES

[1] A. V. Oppenheim: Application of digital signal processing. Prentice-Hall, Englewood Cliffs, N. J. 1978.

[2] L. R. Morris: Digital signal processing microprocessors: forward to the past? IEEE Micro *6* (1986), 6, 6–8.

[3] M. Towsend, M. E. Hoff and R. E. Holm: An NMOS microprocessor for analog signal processing. IEEE J. Solid-State Circuits *SC-15* (1980), 1.

[4] T. Nishitani et al.: A single digital signal processor for telecommunication applications. IEEE J. Solid-State Circuits *SC-16* (1980), 4, 372–376.

[5] μPD77C25 digital signal processor product description. NEC Electronics (Europe) GmbH, 1987.

[6] S. S. Magar at al.: A microcomputer with digital signal processing capability. Proc. International Solid State Conference, 1982.

[7] μPD77220/230 advanced digital signal processor product description. NEC Electronics (Europe) GmbH, 1987.

[8] S. Abiko at al.: Architecture and applications of a 100-ns CMOS VLSI digital signal processor. Proc. of ICASSP'86 (Tokyo), 393–396.

[9] G. A. Frantz, K. Lin, J. B. Reimer and J. Bradley: The Texas Instruments TMS320C25 digital signal microcomputer. IEEE Micro *6* (1986), 6, 10–28.

[10] E. A. Lee: Programmable DSP architectures: Part I. IEEE ASSP Magazine *5* (1988) 4, 4–19.

[11] K. L. Kloker: The Motorola DSP56000 digital signal processor. IEEE Micro *6* (1986), 6, 29–48.

[12] B. Eichen: NEC's μPD77230 digital signal processor. IEEE Micro *6* (1986), 6, 60–69.

[13] P. Papamichalis and R. Similar: The TMS320C30 floating-point digital signal processor. IEEE Micro *8* (1988), 6, 13–29.

[14] M. L. Fuccio et al.: The DSP32C: AT & T's second-generation floating-point digital signal processor. IEEE Micro *8* (1988), 6, 30–48.

[15] J. R. Boddie et al.: The architecture, instruction set and development support for the WE*DSP32 digital signal processor. Proc. of ICASSP'86 (Tokyo), 421—424.

[16] G. R. L. Sohie and K. L. Kloker: A digital signal processor with IEEE floating-point arithmetic. IEEE Micro *8* (1988), 6, 49—67.

[17] K. L. Kloker et al.: The Motorola DSP96002 IEEE floating-point digital signal processor. Proc. of ICASSP'89 (Glasgow), 2480—2483.

[18] ANSI/IEEE standard 754—1985: IEEE standard for binary floating-point arithmetic. IEEE Service Center, Piscataway, N. J. 1985.

[19] T. Nishitani et al.: CMOS floating point signal processor. Proc. of Internat. Conf. on Digital Signal Processing (Florence) 1984, 187—191.

[20] R. E. Owen: VLSI architecture for digital signal processing. VLSI Design (1984), 20—28.

[21] R. H. Cushman: Third-generation DSPs put advanced functions on chip. EDN, July 11 (1985), 59—68.

[22] J. Titus: DSP ICs. EDN, October 16 (1986), 162—176.

[23] A. Aliphas and J. A. Feldman: The versatility of digital signal processing chips. IEEE Micro *7* (1987), 3, 40—45.

[24] R. H. Cushman: μP-like DSP chips. EDN, September *3* (1987), 155—186.

[25] M. Leonard: Digital signal processors. ED, October 13 (1988), 161—166.

[26] D. Shear: EDN's DSP benchmarks. EDN, September 29 (1988) 126—148.

[27] D. Shear: HLL compilers and DSP run-time libraries make DSP system programming easy. EDN, June 23 (1988), 69—76.

[28] W. S. Gass et al.: Multiple digital signal processor environment for intelligent signal processing. Proc. IEEE *75* (1987), 9, 1246—1259.

[29] T. Nishitani: Trends for future microprogrammable signal processor. 6th Kobe Int. Symp. on Electronics and Information Sciences (Kobe), Japan, 1987.

[30] R. C. Agarwal and J. W. Cooley: Vectorized mixed radix discrete Fourier transform algorithms. Proc. IEEE *75* (1987), 9, 1283—1292.

[31] E. Lee and D. G. Messerschmitt: Synchronous data flow. Proc. IEEE *75* (1987), 9, 1235 to 1245.

[32] J. Gaudiot: Data-driven multicomputers in digital signal processing. Proc. IEEE *75* (1987), 9, 1220—1234.

[33] J. Hartung, S. L. Gay and S. H. Haigh: A practical C language compiler/optimizer for real-time implementations on a family of floating point DSPs. Proc. of ICASSP'88 (New York), 1674—1677.

[34] R. Similar and A. Davis: The application of high-level languages to single-chip digital signal processors. Proc. of ICASSP'88 (New York), 1678—1681.

[35] P. E. Papamichalis: FFT implementation on the TMS320C30. Proc. of ICASSP'88 (New York), 1399—1402.

[36] D. Bursky: Operating system for DSPs streamlines Programming. ED, October 13 (1988), 145—147.

[37] E. A. Lee and D. G. Messerschmitt: Static scheduling of synchronous data flow programs for digital signal processing. IEEE Trans. Computers *36* (1987), 2.

[38] E. A. Lee: Programmable DSP architectures: Part II. IEEE ASSP Magazine *6* (1989), 1, 4—14.

[39] S. Pope, J. Rabary and R. W. Brodersen: An integrated automatic layout generation system for DSP circuits. IEEE Trans. on CAD *4* (1985), 3, 285—296.

[40] S. Gomez, S. Gonzales, D. Hsu and A. E. Kuo: An application-specific FFT processor. Electronic Engineering, June (1988), 99—106.

[41] M. Leonard: Building-block chips are busy widening DSP horizons. ED, March 31 (1988), 68—77.

[42] G. Luikuo, M. Fleming and S. Magar: A 500 MOPS DSP chip set. Electronic Engineering, June (1988), 109—113.

[43] G. D. Hillman: DSP56200: An algorithm-specific digital signal processor peripheral. Proc. IEEE *75* (1987), 9, 1185—1191.

[44] N. M. Marinovic, V. G. Oklodzija and L. Roytman: VLSI architecture of a real-time Wigner distribution processor for acoustic signals. Proc. of ICASSP'88 (New York), 2112—2115.

[45] J. P. Roesgen: A high performance microprocessor for DSP applications. Proc. of ICASSP'86 (Tokyo), 397—400.

[46] J. P. Roesgen: The ADSP-2100 DSP microprocessor. IEEE Micro *6* (1986), 6, 49—59.

[47] A. Genusov, P. Feldman, R. Friedlander and R. Shenhav: A new, highly parallel, 32 bit floating point DSP vector signal processor. Proc. of ICASSP'88 (New York), 2116—2119.

[48] S. A. Dyer and L. R. Morris: Floating-point signal processing chips: A new era for DSP systems design? IEEE Micro *8* (1988), 6, 10—12.

[49] R. A. Roberts and C. T. Mullis: Digital Signal Processing. Addison-Wesley 1987.

[50] L. R. Morris and S. A. Dyer: Floating-point digital signal processing chips: The end of the supercomputer era? IEEE Micro *8* (1988), 6, 86.

[51] J. B. G. Roberts: Recent developments in parallel processing. Proc. of ICASSP'89 (Glasgow), 2461—2467.

Ing. Andrej Magyar, CSc., Ústav radiotechniky a elektroniky ČSA V(Institute of Radio Engineering and Electronics — Czechoslovak Academy of Sciences), Lumumbova 1, 182 51 Prague 8. Czechoslovakia.

Prof. Dr.-Ing. Arild Lacroix, Institute of Applied Physics, University of Frankfurt, Robert--Mayer Strasse 2-4, D-6000 Frankfurt am Main 11. Federal Republic of Germany.