ISSN (Online) 2321 – 2004
ISSN (Print) 2321 – 5526

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH IN ELECTRICAL, ELECTRONICS, INSTRUMENTATION AND CONTROL ENGINEERING
Vol. 3, Issue 7, July 2015

High-Performance DSP Processors for Intelligence Applications

Vinni Sharma

Associate Professor, BIT, Durg,(C.G), India

Abstract: Digital signal processors (DSPs) are suitable for a wide variety of computationally intensive real-time applications. This paper describes the architectural features of DSPs for intelligence applications, and the node configuration of the IX-n general purpose neuro-computer, based on the commercially available DSP. DSPs provide high computing power by employing a high level of on-chip parallelism, integrated hardware multipliers, carefully tailored instruction sets, memory organization schemes, hardware support for loop execution, and specific sophisticated addressing modes. High-precisions control and fault-tolerance are achieved by exploiting the high-speed arithmetic, on-chip peripherals, direct memory access (DMA) controllers, multiprocessor support and bit manipulation capabilities of DSPs. Fast multiply/ accumulate time, integrated on-chip random access memory (RAM), large address space, high precision and multiprocessor support are necessary for efficient virtual implementation of neural networks. DSP architectural features make them applicable to both symbolic and connectionist AI models.

Keywords: DSP, Artificial intelligence, Neural network, CISC microprocessor, DSP architecture, FFT butterfly.

I. INTRODUCTION

Intel 2920, the first single-chip DSP device, was introduced in 1979. Although it lacked a hardware multiplier, a feature that has become a trademark of DSP's (canonic signed digit representation was used for filter coefficients instead), this chip marked the beginning of a new era in commercial VLSI chip design. The application area of early single-chip general-purpose DSPs was primarily in digital filter implementation. Digital filtering offers several major advantages compared to purely analog realizations. Among them are easier tuning, parameter modifying, higher reliability, reduced influence of noise, smaller size - especially at low frequencies, and an absence of problems such as components' tolerances, aging and temperature drift. DSPs were optimized to perform algorithms such as FFT or FIWIIR filtering efficiently. These algorithms call for extremely fast execution of multiplication, addition (accumulation) and shifting. To satisfy these requirements, barrel shifters and hardware multipliers must be integrated on chip. This requires a vast silicon area, 40% in the case of the 16x16 bit multiplier on the Texas Instruments TMS32020 DSP chip**.** However, 60-11s multiply/accumulate time for 32-bit floating-point numbers has already been reported, a speed far beyond the processing capabilities of the general purpose RISC microprocessors. Speeds in the 25-35 ns range are available on the enhanced versions of 16-bit fixed-point DSPs. Implementation of fast multipliers is one of the cornerstones of DSP design. Other important architectural features of DSPs include a high level of on-chip parallelism, carefully tailored instruction sets, memory organization, hardware support for loop execution, specific sophisticated addressing modes, integrated peripherals, and DMA controllers.

As DSPs moved from 16-bit (24-bit in the DSP56001) fixed-point chips to 32-bit floating-point devices, some of which are compatible with the IEEE754 floating-point standard, their applications have grown rapidly (see Table I). Faster and cheaper fixed-point DSPs are dominant in telecommunications. In personal computers, DSPs are used as standard peripherals to perform tasks such as graphics, audio processing, speech synthesis and modem implementation.

The NeXT PC. for example, incorporates the D5P56001. Applications such as high-resolution graphics benefit from floating-point arithmetic since its use leads to lower rounding noise, larger dynamic range and easier programming.

Another trend in DSP chip design is towards supporting parallel processing architectures. The TI's floating-point device TM5320C40 features two 32-bit external memory interfaces and six bidirectional parallel communication ports supported by six DMA channels. The TM5320C40 can be gluelessly connected to six other DSPs: an attractive feature for building parallel systems ranging from linear arrays to 2D/3D meshes and hypercubes.

An important reason for the increasing number of DSPs applications is the fact that both design and programming of DSP-based systems have become more pleasant recently. A microsecond-range multiplication speed of microcontrollers may give rise to a serious bottleneck, especially when compared to 25/35 ns multiply-and-accumulate (MAC) cycle of the DSPI6A or the TMS32OC5O. In order to cope with the main advantage of microcontrollers, compact system design, some DSPs include EPROM memories and a significant amount of

DOI 10.17148/IJIREEICE.2015.3732

on-chip peripherals. For example, TMS32OEJ4 DSP microcontroller having 60-ns single-cycle instruction execution time includes 256 words of RAM, 4K words of EPROM memory, two 16-bit general-purpose timers/ counters, a serial port timer, a watch-dog timer, 16 individual bit- selectable 110 pins, an event manager with 6-channel PWM D/A converter, and an asynchronous serial port with codec-compatible modes. Another important fact is that the DSP chips cost is decreasing over time and is comparable to that of general-purpose 16-bit microcontrollers.

TABLE I

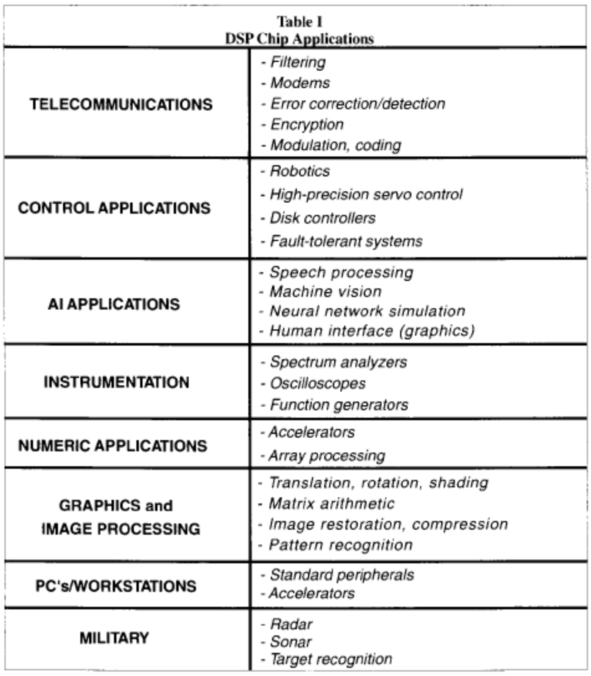


Table I
DSP Chip Applications

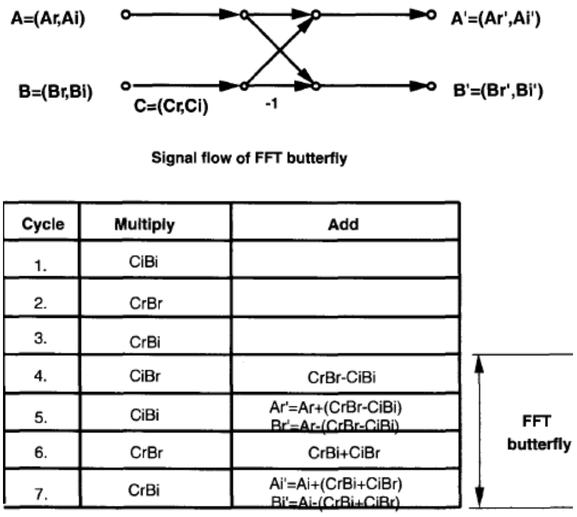
TELECOMMUNICATIONS	- Filtering - Modems - Error correction/detection - Encryption - Modulation, coding
CONTROL APPLICATIONS	- Robotics - High-precision servo control - Disk controllers - Fault-tolerant systems
AI APPLICATIONS	- Speech processing - Machine vision - Neural network simulation - Human interface (graphics)
INSTRUMENTATION	- Spectrum analyzers - Oscilloscopes - Function generators
NUMERIC APPLICATIONS	- Accelerators - Array processing
GRAPHICS and IMAGE PROCESSING	- Translation, rotation, shading - Matrix arithmetic - Image restoration, compression - Pattern recognition
PC's/WORKSTATIONS	- Standard peripherals - Accelerators
MILITARY	- Radar - Sonar - Target recognition

This paper discusses the architectural features of DSPs that make them applicable to intelligent control. The impact of parallel features on numeric processing capabilities and efficient program execution is explained. It also focuses on the role of numerical computations in preprocessing vocal, visual and written input data in real-time artificial intelligence (Al) systems, and illustrates the parallels between neural and DSP computations. Finally, a node configuration of the DC-n virtual implementation of a neural network, based on a commercially available DSP chip, is proposed.

II. ON-CHIP PARALLELISM

The main source of the DSPs' computational power is the efficient execution of the MAC instruction. This instruction supports vector and matrix arithmetic in scientific computations, and implements FIR filtering directly in DSP applications. The MAC instruction enables the simultaneous execution of several actions: multiply/accumulate, fetch of two operands for the next repeated MAC cycle, pointer updating, and data shift register movement. Typically, a shift register is simulated by means of a modulo buffer, although the TMS320C25 performs physical data transfer - an implementation that calls for increased memory bandwidth. Another example of parallelism exploitation is the execution of the FFT butterfly.

A single complex FFT butterfly consists of four real multiplications and six real additions. Fig1 illustrates how those ten operations are executed in four instruction cycles, once the execution pipeline has been properly initiated. The Motorola DSP96002 supports this type of execution pipeline. This processor can perform floating-point multiplication, addition and subtraction simultaneously in each cycle. Additional computational flexibility is obtained by breaking the multiply/accumulate pipeline and enabling separate operations on different sets of registers. Unlike some other DSPs, the DSP96002 follows the register-to register execution model. To increase the I/O throughput and FFT execution speed, DSPs support the *bit-reversed* addressing mode. An assembly/high-level language subroutine performing this function would be time consuming. Since many DSP programs are executed in short loops, most DSP chips support the zero overhead *"hardware Do loops".*



Cycle	Multiply	Add
1.	CiBi	
2.	CrBr	
3.	CrBi	
4.	CiBr	CrBr-CiBi
5.	CiBi	Ar'=Ar+(CrBr-CiBi) Br'=Ar-(CrBr-CiBi)
6.	CrBr	CrBi+CiBr
7.	CrBi	Ai'=Ai+(CrBi+CiBr) Bi'=Ai-(CrBi+CiBr)

Fig 1: Pipeline timing of FFT butterfly

Special dedicated registers are loaded with the loop count, the loop starting address and the loop ending address. They are updated and tested in parallel with the program execution, thus eliminating the need for 2-3 instruction cycles of overhead caused by executing compare, decrement and jump instructions. In the RISC microprocessors the same activity introduces overhead of one (Intel 80860) or two (SPARC, MC88 100) instructions. To support nesting of hardware Do loops, a separate hardware stack may be used. Moreover, to reduce change of-flow overhead, some other techniques, such as delayed branching and conditional instruction execution,

DOI 10.17148/IJIREEICE.2015.3732

are used, Branch prediction mechanisms (used in some general-purpose CISC microprocessors) have not been implemented on DSPs yet due to additional hardware complexity.

III. NEURAL COMPUTING WITH DSP'S

Symbolic processing, dominant in AI applications, is mostly VO intensive; it calls for more memory references and VO operations than numerical computations. DSPs are optimized for high-speed calculations, but not for algorithms such as searching and sorting, or for tasks such as hardware-supported garbage collection, intensive stack manipulations or data tagging. Moreover, development of efficient high-level language compilers is a difficult task, even in the long term prospective. However, the data entry level of many **AI** applications involves number crunching DSP algorithms. Visual, vocal or written input data is presented as sampled signals from sensors. Consequently, computationally intensive processing becomes a significant issue in real time AI applications. To support these real time requirements, a shared memory multiprocessor consisting of 16 crossbar interconnected processing nodes, MX-1/16, has been developed at the Lincoln Laboratory1231. Each node has a Motorola 68020 CISC microprocessor as a CPU, 8Mbytes of DRAM storage, and a three-chip Weitek ACCELfloating- point digital signal processor.

Connectionist intelligent systems do not operate on memory contents where a knowledge base is stored. The information is stored in weighted connections between the neuron cells. The brain, for example, is considered to be a highly parallel computing system consisting of approximately I 0" sparsely interconnected neurons. The number of interconnections per neuron is estimated to be
IO'. This massive parallelism is the reason for the extreme information processing throughput of the brain, since the firing rate of neurons (about 1000 pulse\s) is rather low compared to the electronic switching speed. Major neural computing features that have an impact on VLSI implementations include massive parallelism, high interconnectivity, simple processing nodes and fault-tolerance. DSP architectures provide support for this radically different AI approach.

Fig. 3 depicts a single neuron and the features of DSPs that can be exploited in a virtual implementation of a neural network, where a large number of neurons is allocated to a single powerful processor. Summation of weighted inputs is directly supported by fast multiply and accumulate instructions. All floating point DSPs have a significant amount of on chip RAM (6Kbytes on the DSP32C, and 8Kbytes on the DSP96002 and the TMS320C30), that can be used to store program, threshold function look-up tables, and a limited number of interconnection pattems and weights. Address space of the latest DSPs is very large (24 to 32 address bits), so a significant number of neurons may be allocated to a single DSP. To improve memory access times, interconnection pattems and weights of the currently simulated neuron can be transferred to on-chip RAM by DMA.

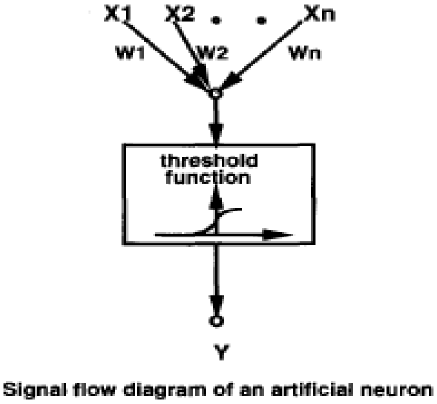
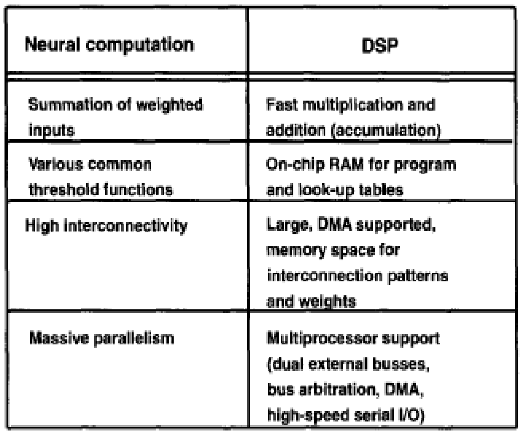


Fig 2: An Artificial neuron

DSPs provide support, such as bus arbitration hardware, multiple buses, DMA and high speed serial links, for building multiprocessor systems. For example, the DSP96002 can simultaneously communicate with another CPU during a DMA time-slice of an instruction cycle and execute its own program. It provides two 32-bit extemal address and datapaths, equipped with an arbitration logic.
Inter-DSP traffic is manageable since most of the neural networks can be partitioned into disjoint (core, influence and remote) regions, with highly interconnected cells mapped to the same DSP. Moreover, computation speed is of primary importance in high-performance neural computers. The communication bandwidth does not have to be extremely high, especially if broadcasting techniques are used in order to convey the state of a neuron hode to all the other neurons hodes .



Neural computation	DSP
Summation of weighted inputs	Fast multiplication and addition (accumulation)
Various common threshold functions	On-chip RAM for program and look-up tables
High interconnectivity	Large, DMA supported, memory space for interconnection patterns and weights
Massive parallelism	Multiprocessor support (dual external busses, bus arbitration, DMA, high-speed serial I/O)

Fig 3:Exploitation of DSP chips in virtual implementation of neural network.

The estimated performance of the MX-1/16 is 120 million interconnects per second (IPS). This is more than twice the estimated performance of Cray WMP-2 (50M IPS), and

almost an order of magnitude higher than the results obtained from the massively parallel CM-2 Connection Machine . A parallel array neuro computer based on the TMS32020 has been developed at IBM's Scientific Center (Palo Alto). Texas Instruments' Ariel has the goal of providing a general-purpose simulator capable of 100 billion IPS. Ariel is a scalable multiprocessor architecture which provides support for the connection of a few thousand processing modules by a hierarchical multiple-level broadcast bus. The single powerful processing module is built around the commercially available Motorola 68030 general-purpose CISC microprocessor, TI'S TMS320C30 DSP chip and 128 Mbytes of module memory. In addition to the broadcast bus interface, each module has its own disk memory and host interface, and four private parallel (64 bits) communication links for local inter-module communication operating
at 100 Mbyte\s.

IV. IX-N ARCHITECTURE

The two proposed node architectures for the IX-n, Intelligent experiment neuro computer are discussed. The node is based on a commercially available DSP chip, DSP96002, and consists of four processors. The DSP96002 chip is chosen for several reasons:

a) it has two independent external memory expansion ports equipped with bus-arbitration hardware,

b) the dual DMA controller enables DMA transfers in parallel with program execution (even if a transfer is between port A and port B, since an internal dedicated DDB internal bus is provided),

c) two DSP96002s may communicate with each other without an extemal shared memory, and

d) DSP96002 provides support for advanced DRAM addressing modes.

Fig 4 illustrates the linear array node architecture. Each memory is shared by two neighboring DSPs. The advantages of this configuration include simple, lower hardware requirements, since the DSP96002 supports memory subsystem design with a minimum amount of glue logic (buffers, transceivers and wait-state generators). The disadvantages include increased memory access time and lowered communication bandwidth. To avoid an overhead of frequent bus arbitration, each processor should primarily reference memory accessible through one port, and use the other port for communication purposes. For example, each DSP96002 could be programmed as a master at port A (memory access), and as a slave at port B.

fig 5 illustrates the shared-memory node configuration. The shared bus is used for communication and a local broadcasting within a single node. Each processor has one dedicated (local) memory, which may drastically increase its throughput (assuming that inter-node communication is kept reasonably low). The shared memory is accessible by four DSPs. It can be used efficiently for program booting, but may become a bottleneck if used for the purposes of an intensive synchronization or operand storage.

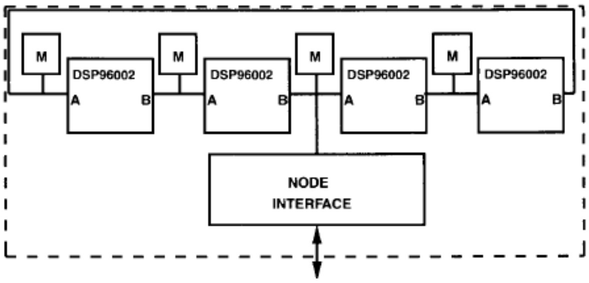


fig 4: IX-n node architecture- Linear **array** node configuration

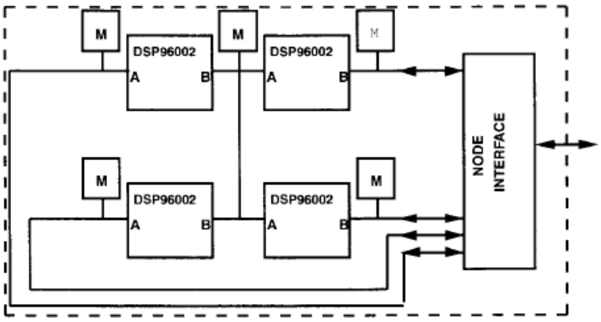


fig 5: IX-n node architecture- Shared. Memory node configuration

The choice of an interconnection network depends on scalability and speed requirements.

A 16x16 crossbar interconnection can be used to connect 16 nodes (64 DSPs). The system of 64 DSPs with 32Mbytes of memory storage allocated to each processor may provide support for approximately 250M interconnections.

This performance is high compared to the capabilities of the systems currently in use for large-scale neural simulation. If some speed is sacrificed and assuming that the major portion of communication is performed by broadcasting, a hierarchical network may lead to a scalable configuration, providing a platform for simulation of significantly more complex networks.

V. CONCLUSION

Commercially available DSP chips provide extremely fast arithmetic, and intensive I/O support features for many application areas including time-critical embedded control applications. Fault-tolerance is achieved by DSPs multiprocessor support. Speech processing or machine vision can exploit the DSP computational power in preprocessing the sampled data. Virtually implemented neural networks can use processing nodes built around DSPs, due to such features as fast multiply/accumulate time, integrated on chip RAMS, large address space and multiprocessor support.

DOI 10.17148/IJIREEICE.2015.3732

ACKNOWLEDGMENT

I would like to acknowledge the experts of DSP processors whose sources we have used and whose analysis we appreciate: **G.K. Ma, R.S. Piepho, E.A. Lee, K.L. Kloker** and others.

REFERENCES

[1] G.K. Ma and F.J. Taylor, "Multiplier policies for digital signal processing," IEEE ASSP Mag., Jan 1990.
[2] A.V. Oppenheim and R.W. Schafer, Discrere- Time Signal Processing. Englewood Cliffs, NJ:Prentice Hall, 1989.
[3] R.S. Piepho and W.S. Wu, "A comparison ofRISC architectures," IEEE Micro, Aug. 1989.
[4] E.A. Lee, "Programmable DSPs: A brief overview,"IEEE Micro, Oct. 1990.
[5] J.J.F. Cavanagh, Digital Computer Arithmetic:Design and Implementation. New York McGraw-Hill, 1984.
[6] E.A. Lee, "Prbgrammable DSP architectures: Part MI," IEEEASSP Mug., Oct. 1988Nan. 1989.
[7] K.L. Kloker, "The Motorola DSP56000 digitalsignal processor," IEEE Micro, Dec. 1986.
[8] G.R.L. Sohie and K.L. Kloker, "A digital signal processor with IEEE floating-point arithmetic,"IEEE Micro, Dec. 1988.
[9] B. Eichen, "NEC's pPD77230 digital signalprocessor," IEEE Micro, Dec. 1988.
[101 W. Andrews, "Designers pack intelligence,memory, speed into new DSPs," Comp. Des., Apr.1, 1990.
[11] K. H. Gurubasavaraj, "Implementation of a self-tuning controller using digital signal processor chips," IEEE Control Sysf. Mug., June 1989.

BIOGRAPHY

Vinni Sharma received the M.Tech degree in Instrumentation and Control Engineering from the CSVTU University in 2007.She is currently working as an Associate Professor from the year 2004 in the Department of Electronics and Telecommunications Engineering, Bhilai Institute of Technology, Durg(C.G). Her research interest includes signal processing architectures, DSP processors and embedded systems for wireless communications.

DOI 10.17148/IJIREEICE.2015.3732