CIRCULAR BUFFERING

A digital signal processor is a specialized microprocessor for the kind of algorithms employed in digital signal processing (DSP). The main goal is to accelerate the calculations while keeping the power consumption as low as possible. In this article, we review a basic addressing capability of DSP processors, i.e. circular buffering, which allows us to significantly accelerate the data transfer in a real-time system.

Please note that since the acronym “DSP” stands for both “digital signal processing” and “digital signal processor," we will use the term “DSP processor” when referring to the hardware rather than the algorithm.

Since the finite-impulse-response (FIR) filtering is a common operation in DSP, we will continue our discussion based on examining the difference equation of an FIR filter. This simple example will show the typical properties of many DSP algorithms. After reviewing the problem of handling the incoming samples, we will discuss the circular buffering as an efficient solution to the problem.

www.binils.com

Linear Buffering

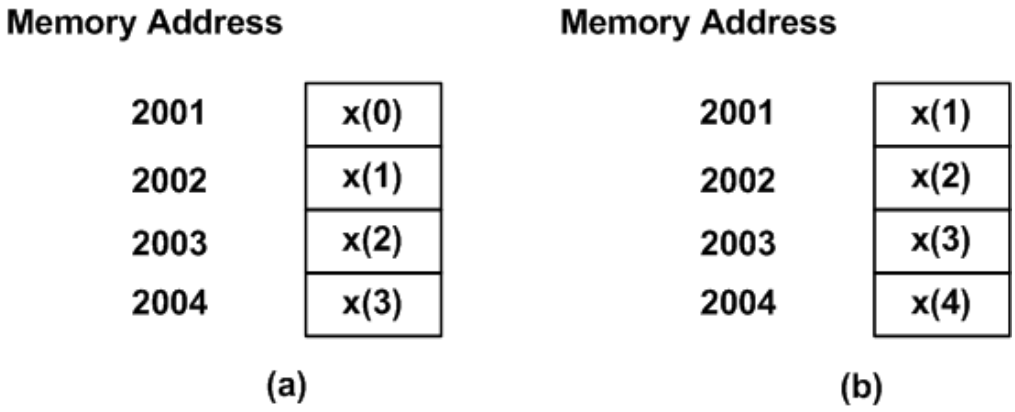


Figure (a) shows that, at time index n = 3 n=3, the last four samples stored in the

memory are x (3) x(3), x (2) x(2), x (1) x(1), and x (0) x(0). When a new sample is acquired at n = 4 n=4, the last four samples, x (4) x(4), x (3) x(3), x (2) x(2), and x (1) x(1), will be stored in the memory as shown in Figure (b). This approach to storing the incoming samples is called the linear buffering. As we will see in a minute, it is simple but not at all efficient.

The main problem with linear buffering is the amount of the data transfer that we need to handle. Consider the above linear buffer for the four-tap FIR filter. With each new sample, we have to read three memory locations and write their contents to another location in our array. This is shown in Figure 4. We observe that the number of read and write operations are proportional to the length of the filter. That's why the linear buffering is not an efficient method of storing the incoming samples. For example, if we use a linear buffer to implement a 256-tap filter, then, with each new sample acquired, we have to perform almost 256 read and write operations!

www.binils.com

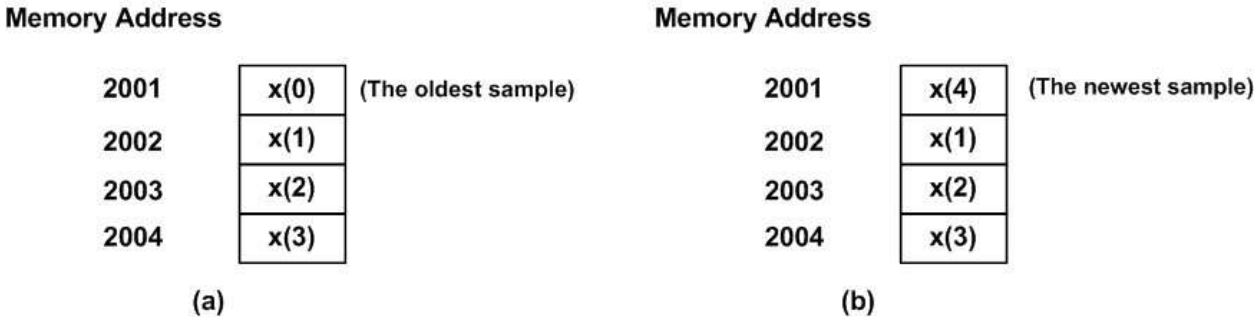
Circular Buffering

Let's examine the above Figure one more time. The memory in Figure (a) stores four input samples: x (0) x(0), x (1) x(1), x (2) x(2), and x (3) x(3). Figure (b) stores x (1) x(1), x (2) x(2), x (3) x(3) along with the new sample x (4) x(4). We observe that three values, i.e. x (3) x(3), x (2) x(2), x (1) x(1), are common between the two cases

shown in Figure 3; however different memory locations are used to store these common values.

Can we use the above observation to reduce the number of the required read and write operations? For example, what's the point of copying x (3) x(3) from the hypothetical memory location 2004 in Figure (a) to the address 2003 in Figure (b), and then using it in the upcoming calculations? If we keep the common values where they currently reside, then we only need to store the new sample, i.e. x (4) x(4), in the memory. When x (4) x(4) is acquired, we no longer need x (0) x(0), hence, we can use the memory location 2001 to store x (4) x(4). The result is shown in below Figure . As shown in this figure, the newest sample, x (4) x(4), replaces the oldest sample x (0) x(0). This technique for storing the incoming samples is called the circular buffering. In this way, with each new sample, only a single memory write operation is required.

www.binils.com



Pipelining

Pipelining is a technique which allows two or more operations to overlap during execution. In pipelining, a task is broken down into a number of distinct subtasks which are then overlapped during execution. It is used widely in digital signal processors to increase speed. A pipeline is akin to a typical production line in a factory, such as a car or television assembly plant. As in the production line, the task is broken into small, independent subtasks called pipe stages. The pipe stages are connected in series to form a pipe and the stages are executed sequentially. As we have seen in the last example, an instruction can be broken down into three steps. Each step in the instruction can be regarded as a stage in a pipeline and so can be overlapped. By overlapping the instructions, a new instruction is started at the start of each clock cycle as shown in Figure 6(a).

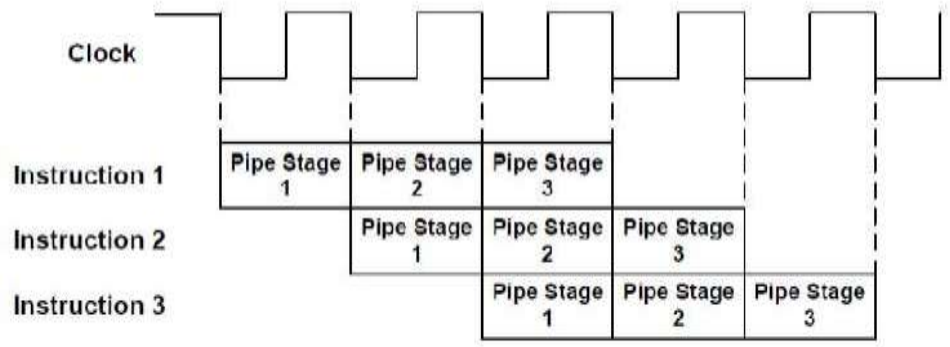


Figure 6 (a)

www.binils.com

Figure 6(b) gives the timing diagram for a three-stage pipeline, drawn to highlight the instruction steps. Typically, each step in the pipeline takes one machine cycle.

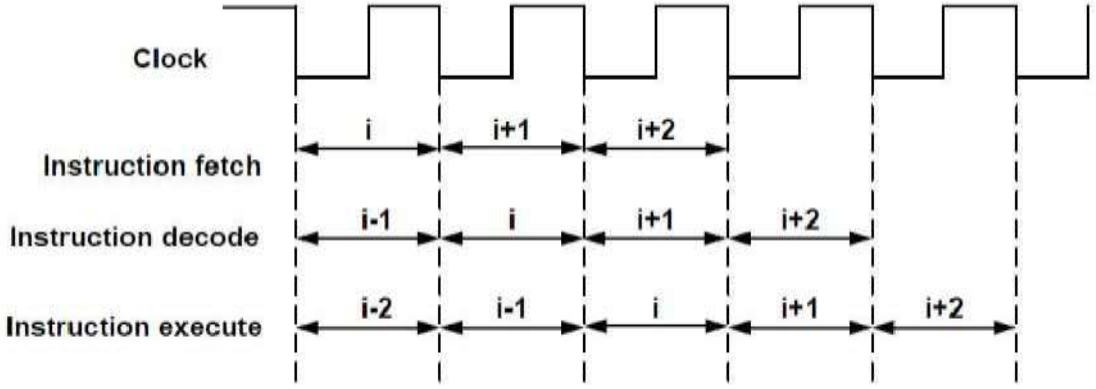


Figure 6 (b)

Thus during a given cycle up to three different instructions may be active at the same time, but the completion stage of each will be different. The key to an instruction pipeline is that the three parts of the instruction (that is, fetch, decode and execute) are independent and so the execution of multiple instructions can be overlapped. In Figure 6(b), it is seen that, at the i^{th} cycle, the processor could be simultaneously fetching the i^{th} instruction, decoding the $(i-1)^{th}$ instruction and at the same time executing the $(i-2)^{th}$ instruction, which are then overlapped during execution. It is used extensively in digital signal processors to increase speed.

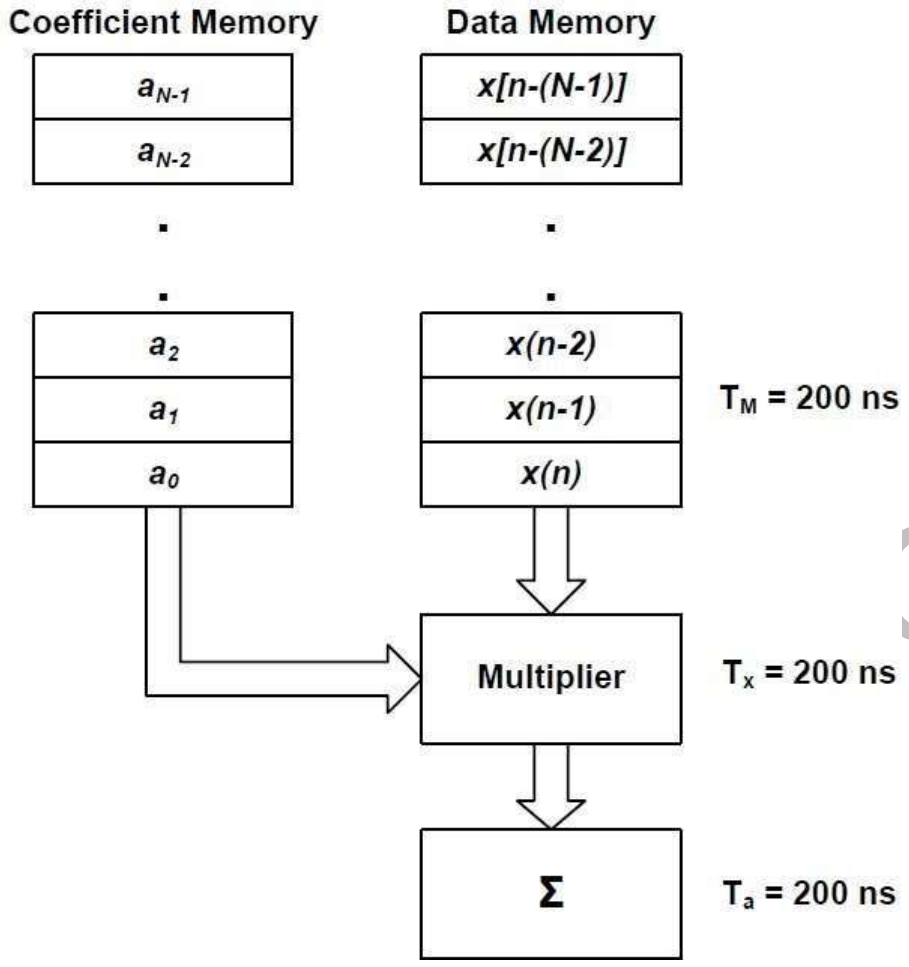


Figure 7. Non-pipelined MAC configuration. Products are clocked into the accumulator every 400 ns.

Solution:

1. The coefficients and the data arrays are stored in memory as shown in Figure 7. In the non-pipelined mode, the coefficients and data are accessed sequentially and applied to the multiplier. The products are summed in the accumulator. Successive multiplication-accumulation (MAC) will be performed once every 400 ns (200 + 100 + 100), giving a throughput of *2.5* x 10^6 operations per second.
2. The arithmetic operations involved can be broken up into three distinct steps: memory read, multiply, and accumulate. To improve speed these steps can be overlapped. A speed improvement of 2:1 can be achieved by inserting pipeline registers between the memory and multiplier and between the multiplier and accumulator as shown in Figure 8

Download Binils Android App in Playstore Download Photoplex App

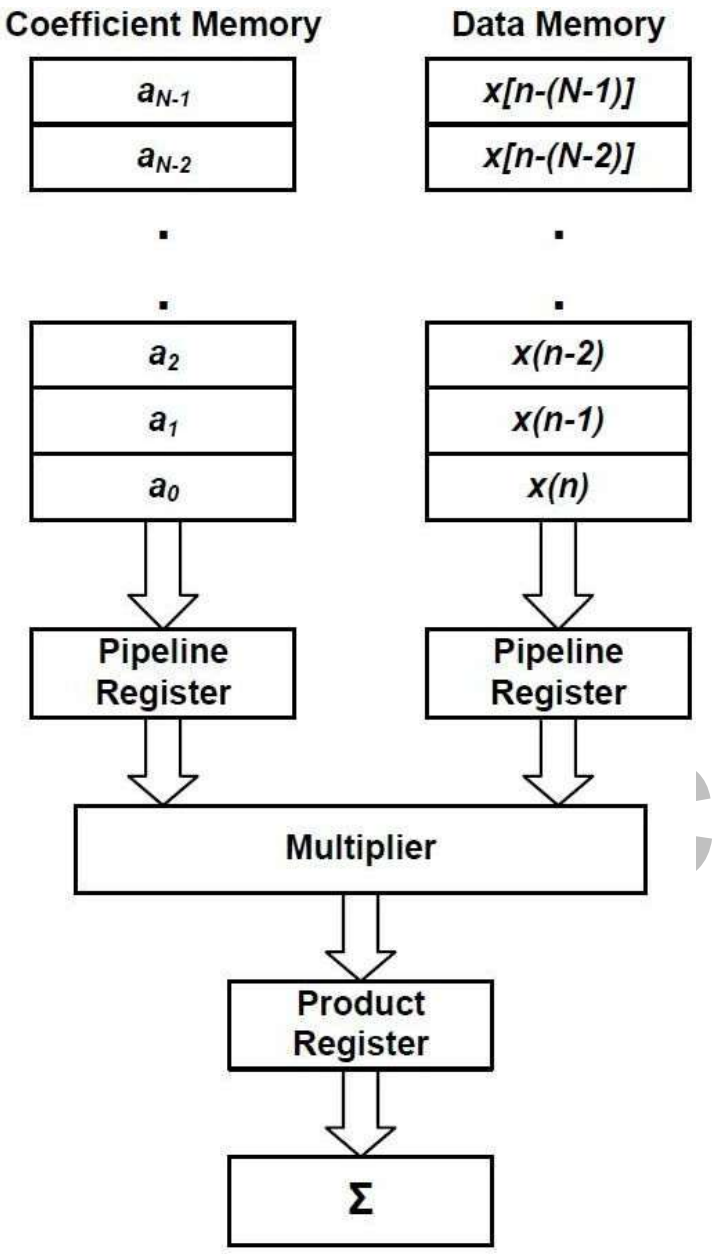


Figure 8. Pipelined MAC configuration.

The pipeline registers serve as temporary storage for coefficient and data sample pairs. The product register also serves as a temporary store for the product.

The timing diagram for the pipeline configuration is shown in Figure 9. As is evident in the timing diagram, the MAC is performed once every 200 ns. The limiting factor is the basic transport delay through the slowest element, in this case the memory. Pipeline overheads have been ignored.

Download Binils Android App in Playstore Download Photoplex App

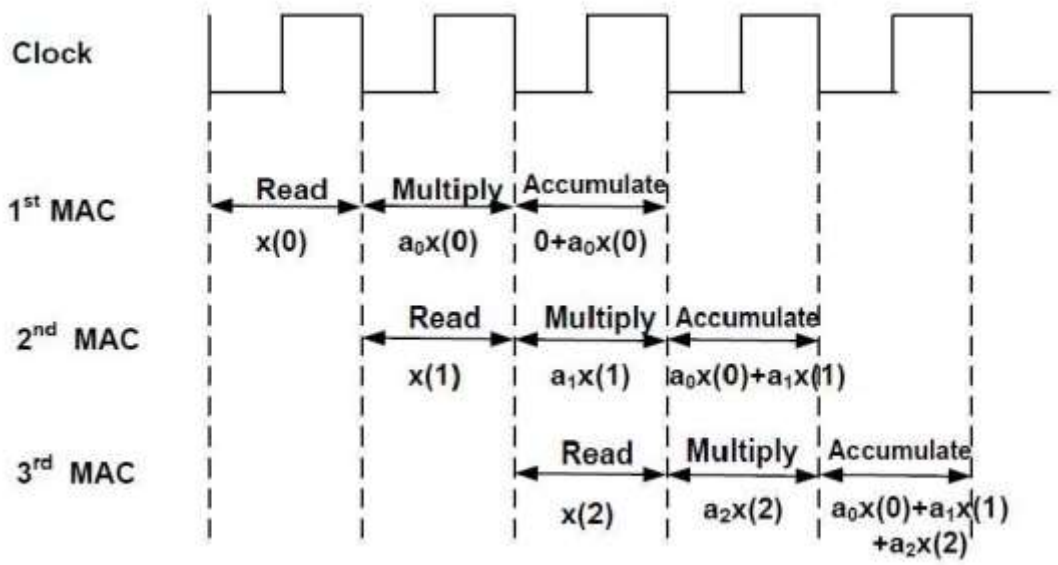


Figure 9. Timing diagram for a pipelined MAC unit. When the pipeline is full, a MAC operation is performed every clock cycle (200 ns).

DSP algorithms are often repetitive but they are highly structured. This made them well suitable for multilevel pipelining. For example, FFT requires the continuous calculation of butterflies. Although each butterfly requires different data and coefficients, the basic butterfly arithmetic operations are identical. Thus arithmetic units such as FFT processors can be tailored to take advantage of this. Pipelining ensures a steady flow of instructions to the CPU, and in general leads to a significant increase in system throughput.

www.binils.com

However, on occasions pipelining may cause problems. For example, in some digital signal processors, pipelining may cause an unwanted instruction to be executed, especially near branch instructions, and the designer should be aware of this possibility.

Hardware multiplier—accumulator

The basic numerical operations in DSP are multiplications and additions. Multiplication, in software, is notoriously time consuming. Additions are even more time consuming if floating point arithmetic is used. To make real-time DSP possible a fast, dedicated hardware multiplier-accumulator (MAC) using fixed or floating point arithmetic is mandatory. Fixed or floating hardware MAC: is now standard in all digital signal processors. In a fixed point processor, the hardware multiplier typically accepts two 16-bit 2's complement fractional numbers and computes a *32*-bit product in a single cycle (25 ns typically) The average MAC instruction time can be significantly reduced through the use of special repeat instructions.

A typical DSP hardware MAC configuration is depicted in Figure 10. In this configuration the multiplier has a pair of input registers that hold the inputs of the multiplier, and a 32-bit product register which holds the result of a multiplication. The output of the P (product) register is connected to a double-precision accumulator, where the products are accumulated.

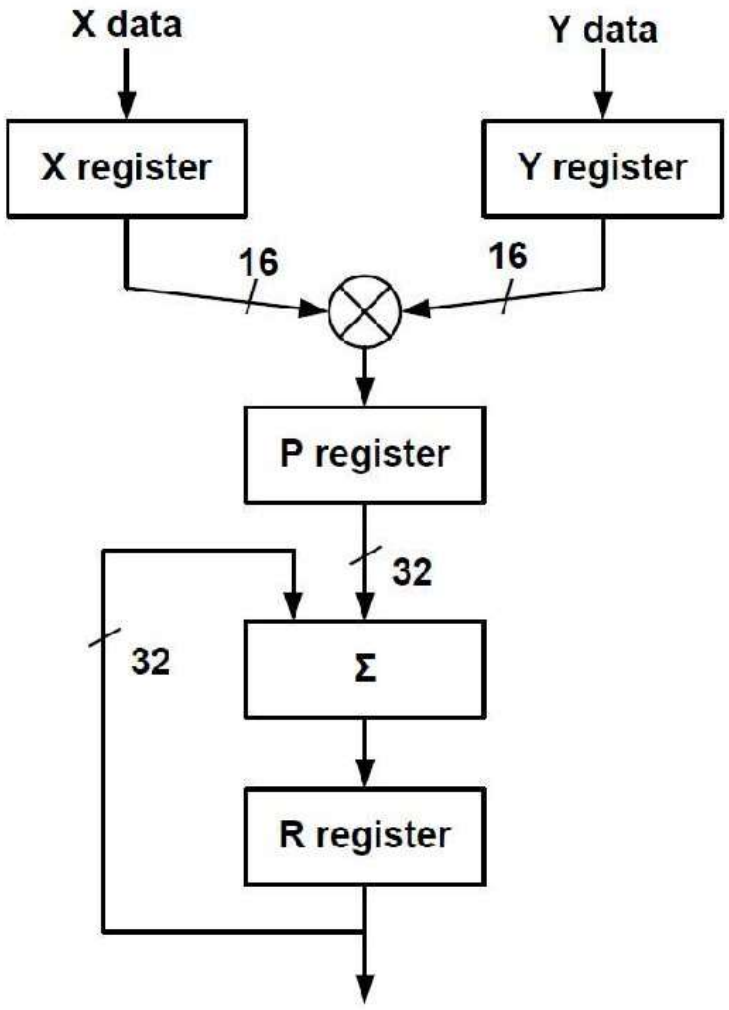


Figure 10. A typical MAC configuration in DSPs.

The principle is very much the same for hardware floating-point multiplier - accumulators, except that the inputs and products are normalized floating- point numbers. Floating-point MACs allow fast computation of DSP results with minimal errors. The DSP algorithms such as FIR and IIR filtering suffer from the effects of finite word-length (coefficient quantization and arithmetic errors). Floating point offers a wide dynamic range and reduced arithmetic errors, although for many applications the dynamic range provided by the fixed-point representation is adequate.

General-purpose digital signal processors

General-purpose digital signal processors are basically high speed microprocessors with hardware architectures and instruction sets optimized for DSP operations. These processors use parallelism extensively. Harvard architecture, pipelining and dedicated hardware are used whenever possible for performing time-consuming operations, such as shifting/scaling, multiplication, and so on.

General-purpose DSPs have been evolved substantially over the last decade as a result of the never-ending quest to find better ways to perform DSP operations, in terms of computational efficiency, ease of implementation, cost, power consumption, size, and application-specific needs. The quenchless

hunger for improved computational efficiency has led to substantial reductions in instruction cycle times and, more importantly, to increasing sophistication in the hardware and software architectures. It is now common to have dedicated, on-chip arithmetic hardware units (e.g. to support fast multiply / accumulate

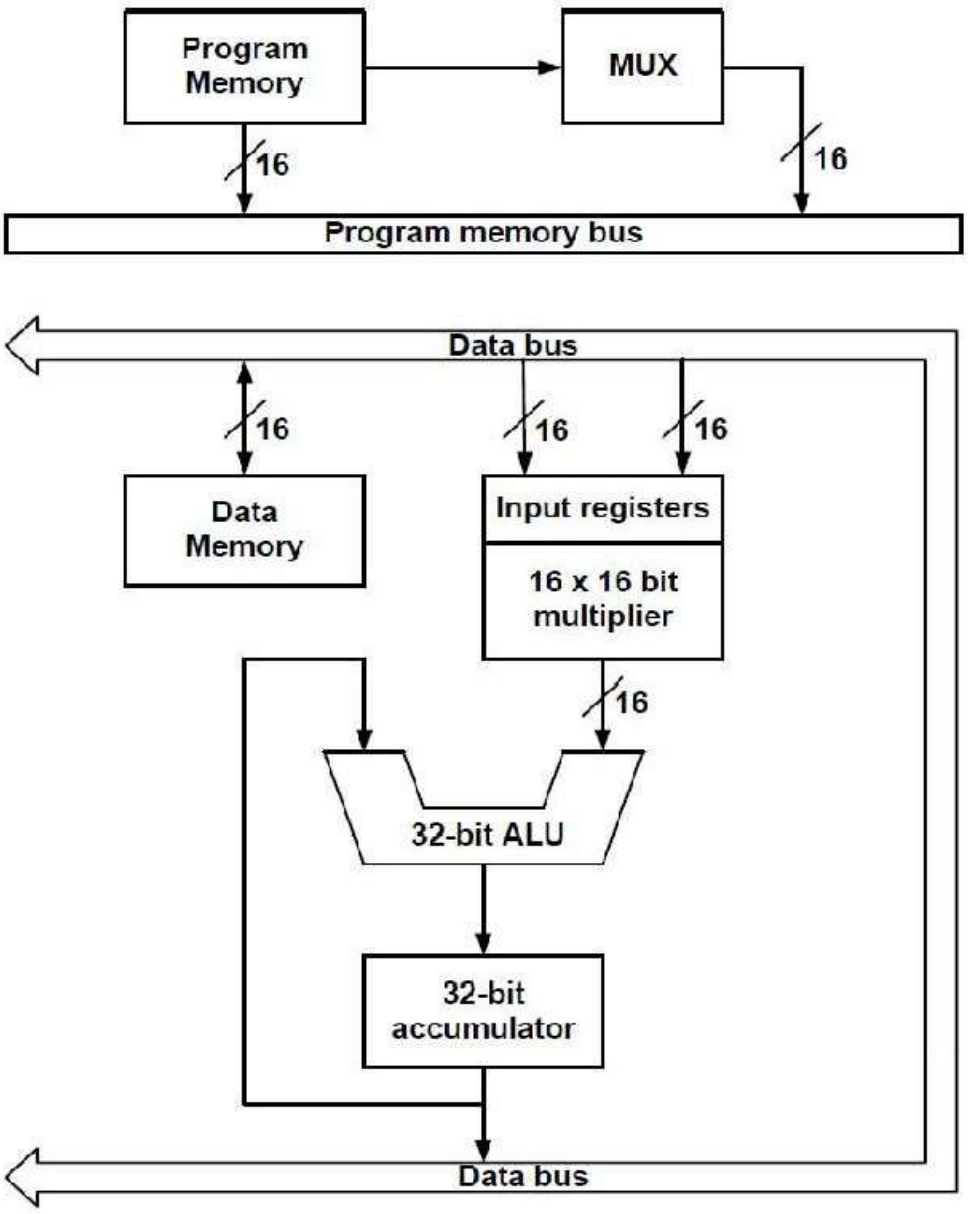
www.binils.com

operations), large on-chip memory with multiple access and special instructions for efficient execution of inner core computations in DSP. There is also a trend towards increased data word sizes (e.g. to maintain signal quality) and increased parallelism (to increase both the number of instructions executed in one cycle and the number of operations performed per instruction). Thus, in newer general-purpose DSP processors increasing use is made of multiple data paths/arithmetic to support parallel operations. DSP processors based on SIMD (Single Instruction, Multiple Data), VLIW (Very Large Instruction Word) and superscalar architectures are being introduced to support efficient parallel processing. In some DSPs, performance is enhanced further by using specialized, on-chip co-processors to speed up specific DSP algorithms such as FIR filtering and Viterbi decoding. The explosive growth in communications and digital audio technologies has had a major influence in the evolution of DSPs, as has growth in embedded DSP processor applications.

Fixed Point Digital Signal Processors

Fixed-point DSP processors available today differ in their detailed architecture and the onboard resources provided. A summary of key architectures of four generations of fixed-paint- DSP processors from four leading semiconductor manufacturers is given in Table 7.1. The classification of DSP processors into the four generations is partly based on historical reasons, architectural features, and computational performance.

www.binils.com



Download Binils Android App in Playstore Download Photoplex App

DSP functionalities - circular buffering – DSP architecture – Fixed and Floating point architecture principles – Programming – Application examples.

Introduction

In general, DSP processors can be classified into two broad categories as general purpose and special purpose. Fixed-point devices such as Texas Instruments TMS320C54x, and Motorola DSP563x processors, and floating- point processors such as Texas Instruments TMS320C4x and Analog Devices ADSP21xxx SHARC processors are included in DSP Processors.

Special purpose hardware are divided into two categories,

1. One type of special- purpose hardware is sometimes called an algorithm-specific digital signal processor. Hardware designed for efficient execution of specific DSP algorithms such as digital filters, Fast Fourier Transform comes under this category.

2. Another type of hardware is sometimes called an application-specific digital signal processor. Hardware designed for specific applications: for example telecommunications, digital audio, or control applications comes under this category.

In most cases application-specific digital signal processors execute specific algorithms, such as PCM encoding/decoding, but they are also required to perform other application-specific operations. Examples of special-purpose DSP processors are Cirrus's processor for digital audio sampling rate converters (CS8420), Intel's multi- channel telephony voice echo canceller (MT9300), FFT processor (PDSPI65I5A) and programmable FIR filter (VPDSP 16256).

www.binils.com

Both general-purpose and special-purpose processors can be designed with single chips or with individual blocks of multipliers, ALUs, memories, and so on. First, let us discuss the architectural features of digital signal processors that have made real-time DSP in many possible areas.

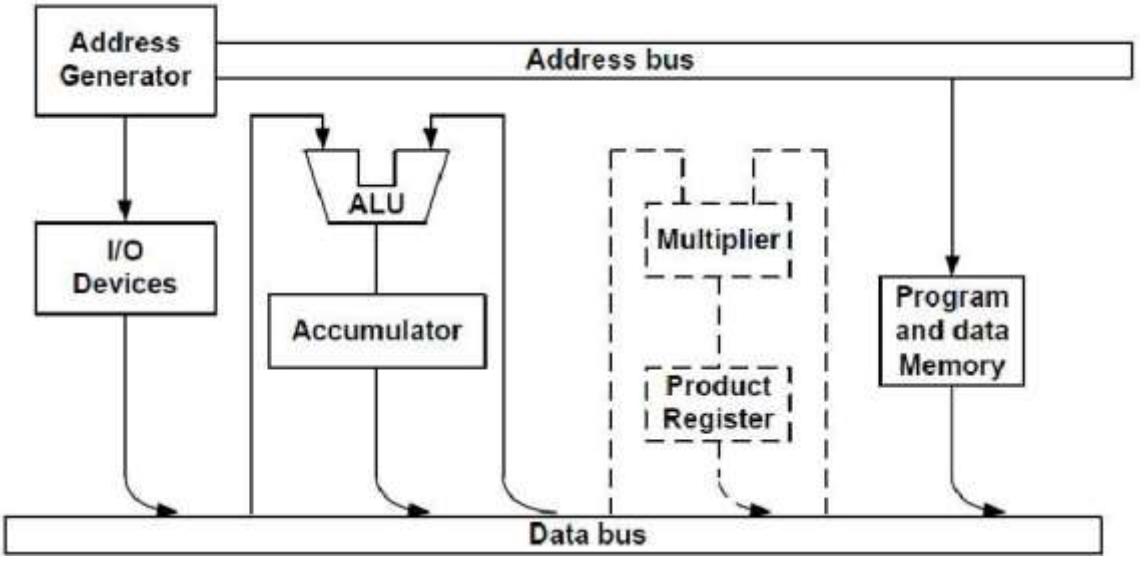


Figure 1. A simplified architecture for standard microprocessor

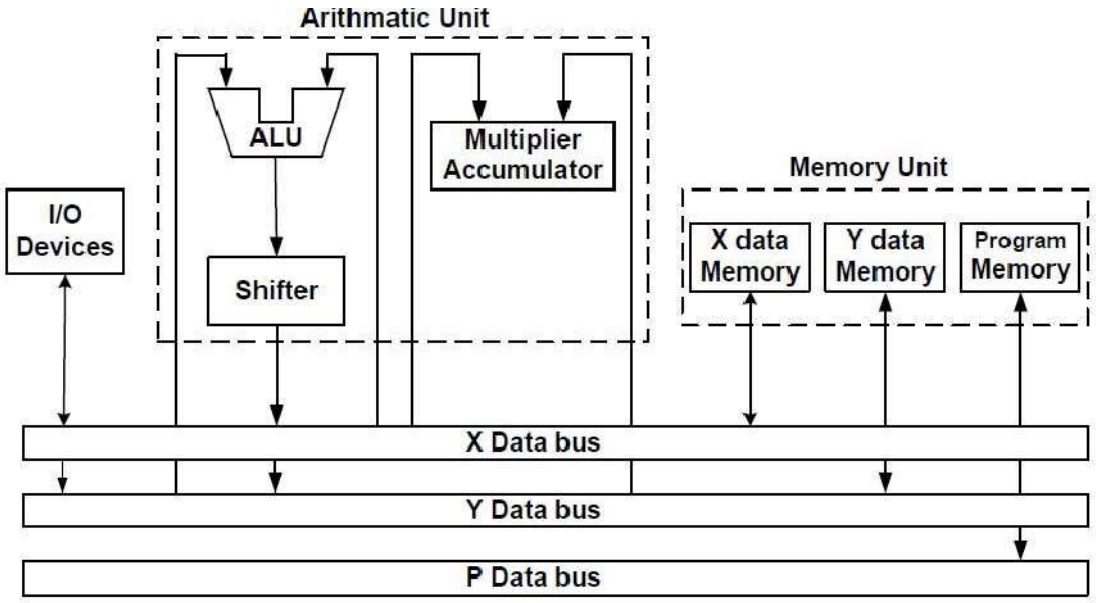


Figure 2. Basic generic hardware architecture for signal processing

Figure 2 shows generic hardware architecture suitable for real time DSP It is characterized by the multiple bus structure with separate memory space for data and program instructions. The data memories hold input data, intermediate data values and output samples, as well as fixed coefficients for example, digital filters or FFTs. The program instructions are stored in the program memory.

www.binils.com

The I/O port provides a means of passing data to and from external devices such as the ADC and DAC or for passing digital data to other processors. Direct memory access (DMA), if available, allows for rapid transfer of blocks of data directly to or from data RAM, typically under external control.

Arithmetic units for logical and arithmetic operations include an ALU, a hardware multiplier and shifters (or multiplier--accumulator)

The main necessary of this architecture is that most DSP algorithms (such as filtering correlation and fast Fourier transform) involve repetitive arithmetic operations such as multiply, add, memory accesses, and heavy data flow through the CPU. The architecture of standard microprocessors is not suited for this type of activities. So an important goal in DSP hardware design is to optimize both the hardware architecture and the instruction set for DSP operations. In digital signal processors, this is achieved by making use of the concepts of parallelism. In particular, the following techniques are used:

1. Harvard architecture;
2. pipe-lining;
3. fast, dedicated hardware multiplier/accumulator;
4. special instructions dedicated to DSP;
5. replication;
6. on-chip memory/cache;
7. Extended parallelism — SIMD, VLIW and static superscalar processing.

Download Binils Android App in Playstore Download Photoplex App

For successful DSP design, it is important to understand these key architectural features.

Harvard architecture:

The principal feature of the Harvard architecture is that the program and data memories lie in two separate spaces, permitting a full overlap of instruction fetch and execution. Standard microprocessors, such as the Intel 6502, are characterized by a single bus structure for both data and instructions, as shown in Figure 1.

Suppose that in a standard microprocessor if a value op I at address ADR 1 in memory into the accumulator is to be read and then to be stored at two other addresses, ADR2 and *ADR3*. The instructions could be

LDA ADRI load the operand op1 into the accumulator from ADRI STA

ADR2 store op1 in address ADR2

STA ADR3 store op1 in address ADR3

Typically, each of these instructions would involve three distinct steps:

• instruction fetch;

• instruction decode;

• instruction execute.

www.binils.com

In our case, the instruction fetch involves fetching the next instruction from memory, and instruction execute involves either reading or writing data into memory. In a standard processor, without Harvard architecture, the program instructions (that is, the program code) and the data (operands) are held in one memory space; see Figure 3. Thus the fetching of the next instruction while the current one is executing is not allowed, because the fetch and execution phases each require memory access.

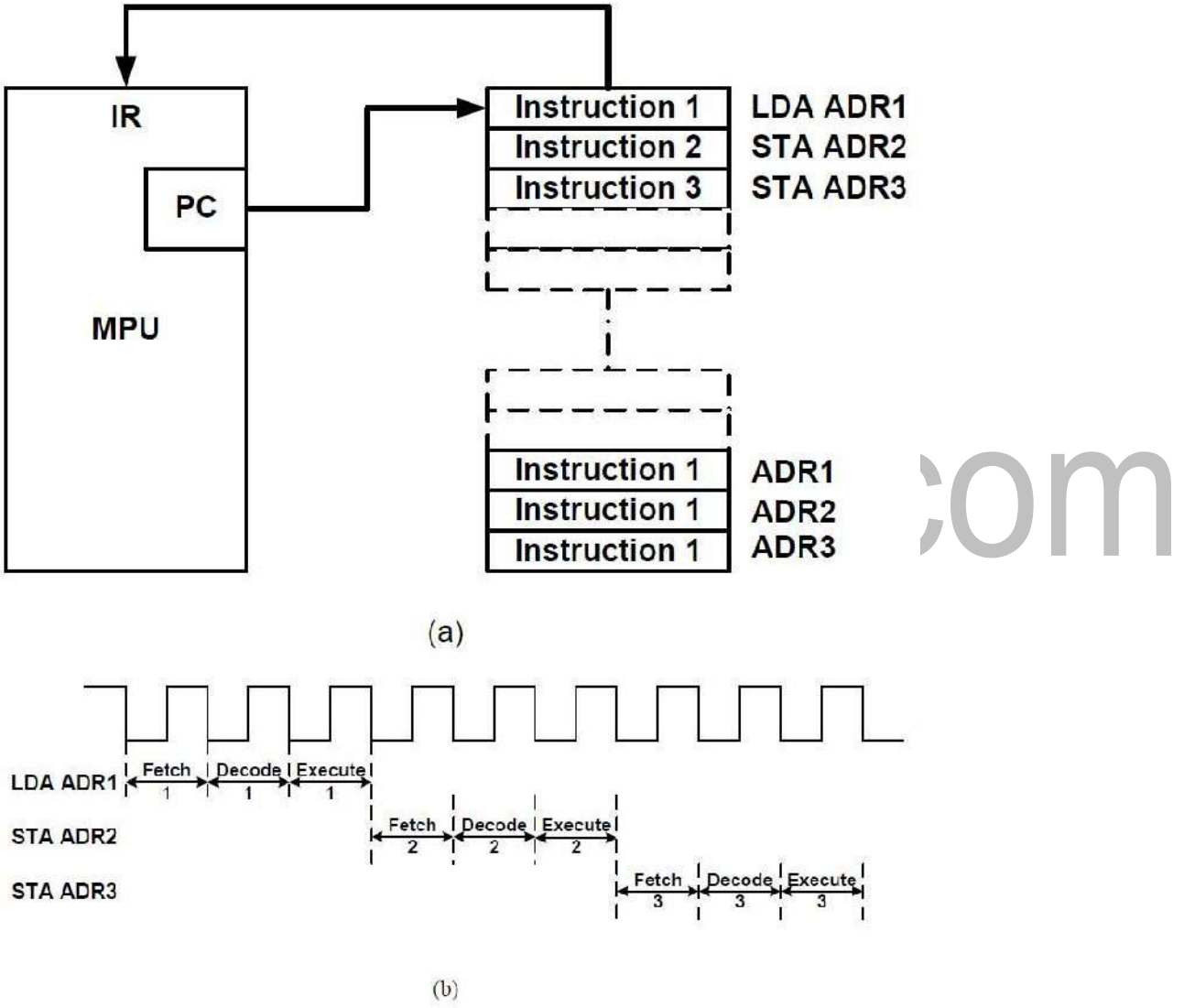


Figure 3. An illustration of instructions fetch, decode, and execute in a Non-Harward architecture with single memory space. (a) instruction fetch from memory (b) timing diagram

In a Harvard architecture (Figure 4), since the program instructions and data lie in separate memory spaces, the fetching of the next instruction can overlap the execution of the current instruction as shown in Figure 5. Normally, the program memory holds the program code, while the data memory stores variables such as the input data samples.

Download Binils Android App in Playstore Download Photoplex App

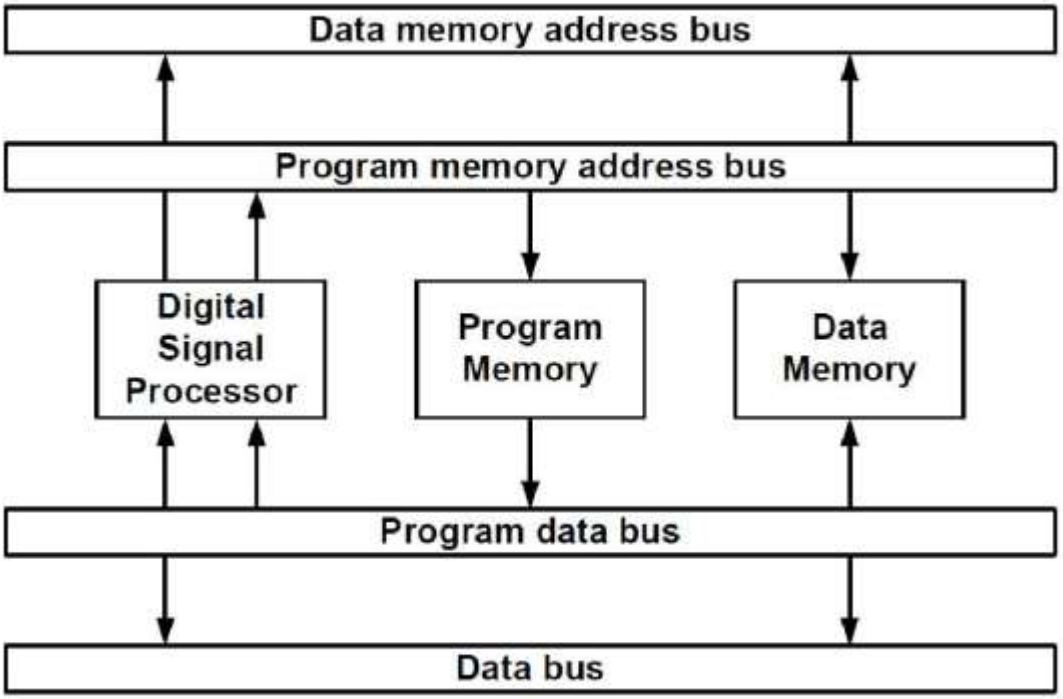


Figure 4. Basic Harvard architecture with separate data and program memory spaces

www.binils.com

It may be seen from Figure 4 that data and program instruction fetches can be overlapped as two independent memories are used in the architecture. This is explained with the help of the timing diagram as shown in Figure 5 below.

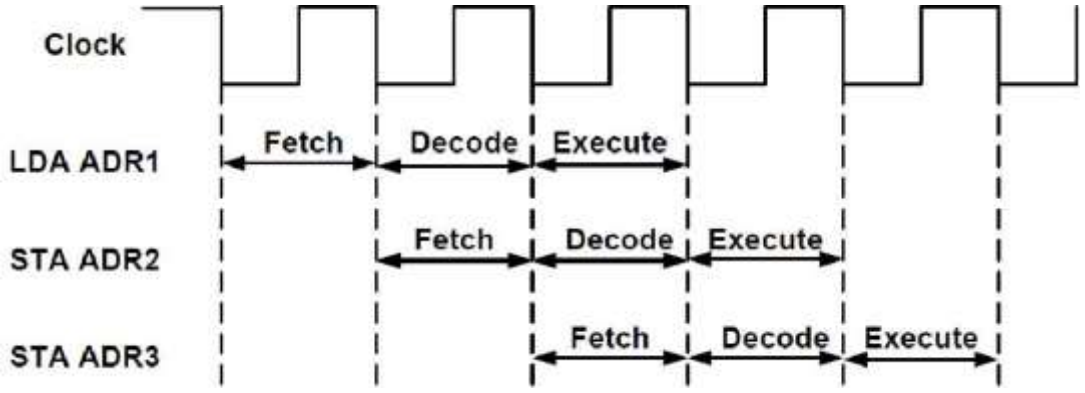


Figure 5. An illustration of instruction overlap made possible by the Harvard architecture

Strict Harvard architecture is used by some digital signal processors (for example Motorola D5P56000), but most use a modified Harvard architecture (for example, the TMS32O family of processors). For example in the modified architecture used by the TMS32O, separate program and data memory spaces are still maintained. But unlike the strict Harvard architecture, communication between the two memory spaces is permitted here.

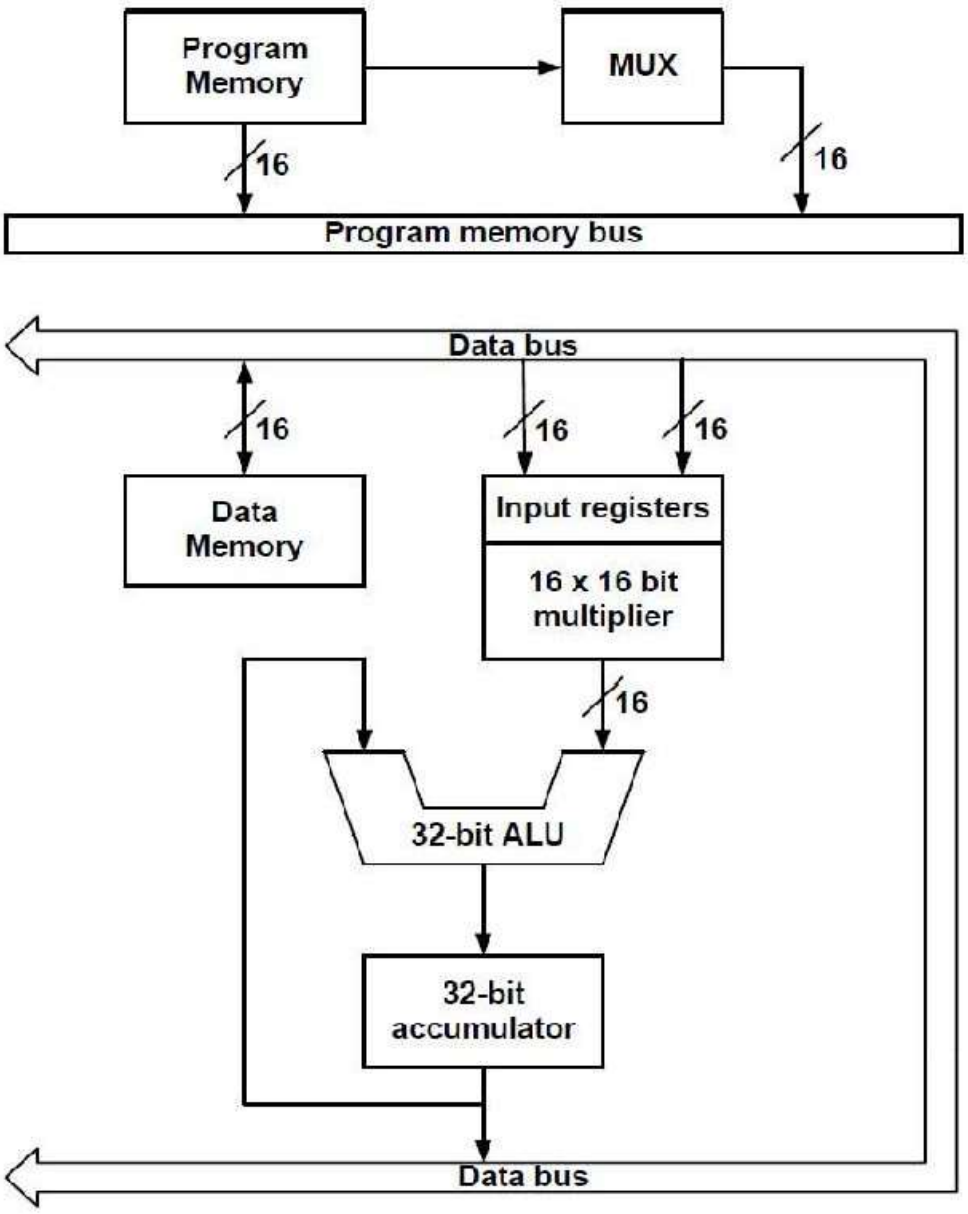
Download Binils Android App in Playstore Download Photoplex App

operations), large on-chip memory with multiple access and special instructions for efficient execution of inner core computations in DSP. There is also a trend towards increased data word sizes (e.g. to maintain signal quality) and increased parallelism (to increase both the number of instructions executed in one cycle and the number of operations performed per instruction). Thus, in newer general-purpose DSP processors increasing use is made of multiple data paths/arithmetic to support parallel operations. DSP processors based on SIMD (Single Instruction, Multiple Data), VLIW (Very Large Instruction Word) and superscalar architectures are being introduced to support efficient parallel processing. In some DSPs, performance is enhanced further by using specialized, on-chip co-processors to speed up specific DSP algorithms such as FIR filtering and Viterbi decoding. The explosive growth in communications and digital audio technologies has had a major influence in the evolution of DSPs, as has growth in embedded DSP processor applications.

Fixed Point Digital Signal Processors

Fixed-point DSP processors available today differ in their detailed architecture and the onboard resources provided. A summary of key architectures of four generations of fixed-paint- DSP processors from four leading semiconductor manufacturers is given in Table 7.1. The classification of DSP processors into the four generations is partly based on historical reasons, architectural features, and computational performance.

www.binils.com



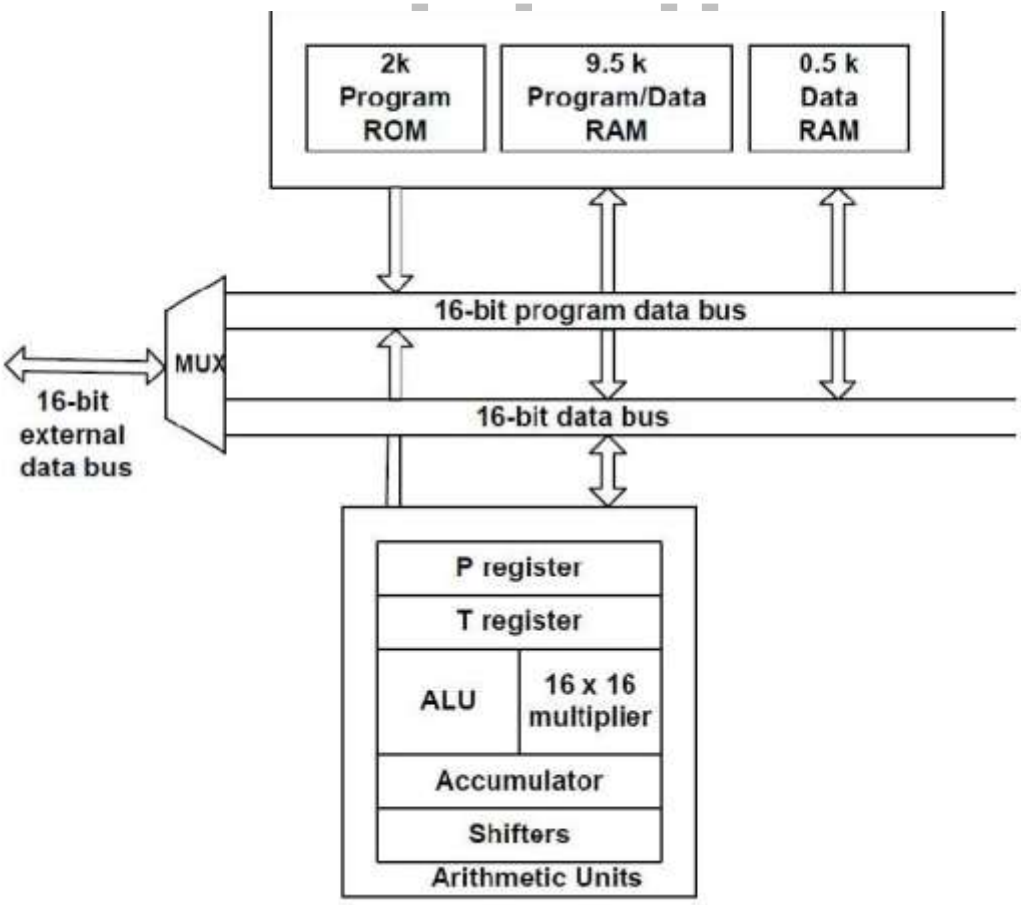
Download Binils Android App in Playstore Download Photoplex App

Figure 11 A simplified architecture of a first generation fixed-point DSP processor (Texas Instruments TMS32OC1O).

Key features of the TMS32OC1x are the dedicated arithmetic units which include a multiplier and an accumulator. The processor family has a modified Harvard architecture with two separate memory spaces for programs and data. It has an on-chip memory and special instructions for execution of basic DSP algorithms, although these are limited.

Second generation fixed-point DSPs have substantially enhanced features as compared to the first generation. In most cases, these include much larger on-chip memories and more special instructions to support efficient execution of DSP algorithms. As a result, the computational performance of second generation DSP processors is four to six times that of the first generation.

Typical second generation DSP processors include Texas Instruments TMS320C5x, Motorola DSP5600x, Analog Devices ADSP2 I xx and Lucent Technologies DSPI6xx families. Texas Instruments first and second generation DSPs have a lot in common, architecturally, but second generation DSPs have more features and increased speed. The internal architecture that typifies the TMS320C5x family of processors is shown in Figure 12 in a simplified form to emphasize the dual internal memory spaces which are characteristic of the Harvard architecture.



Figne 12. A simplified architecture of a second generation fixed-point DSP (Texas Instruments TMS320C5O).

Download Binils Android App in Playstore Download Photoplex App

Special instructions for DSP operations include a multiply and accumulate with data move instruction which, for example, can be combined with a repeat instruction to execute an FIR filter with considerable time savings. Its bit-reversed addressing capability is useful in FFT's. Unlike the first generation fixed-point processor family Clx, which has a very limited internal memory, the C5x provides more on-chip memory.

www.binils.com

The Motorola DSP5600x processor is a high-precision fixed point digital signal processor. Its architecture is depicted in Figure 13.

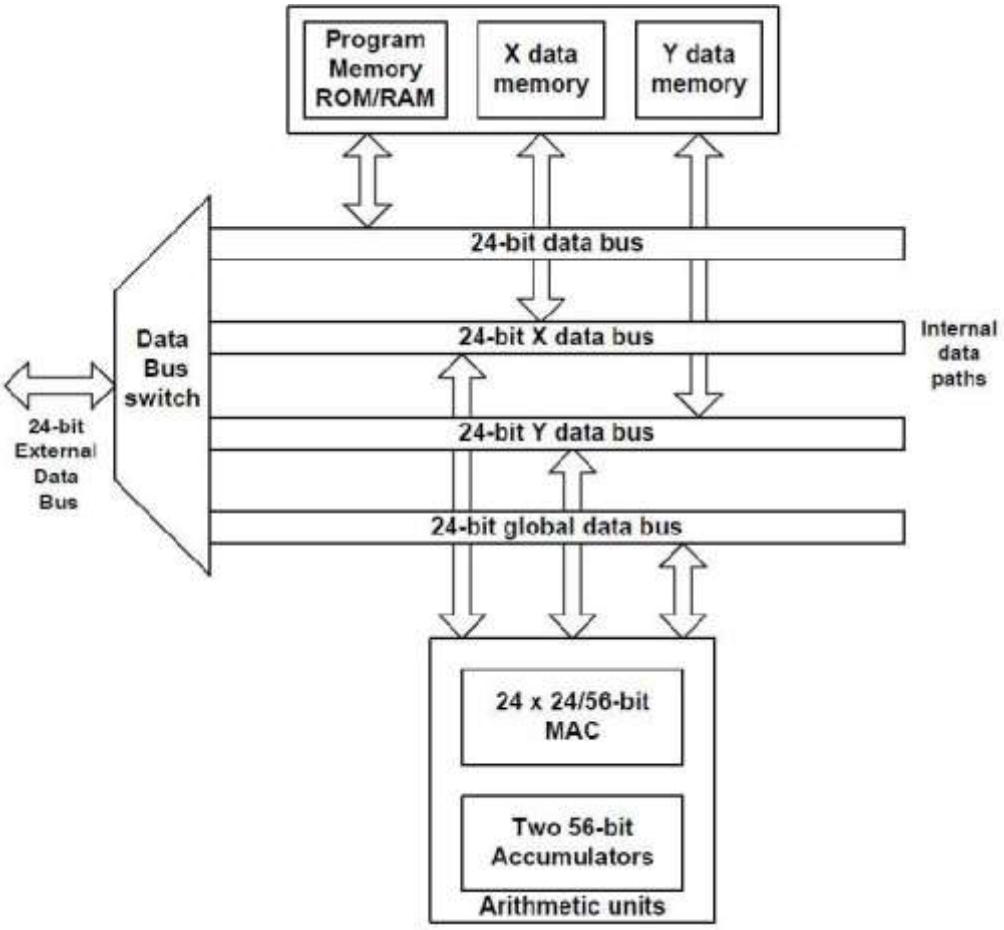


Figure 13. A simplified architecture of a second generation fixed-point DSP (Motorola D5P56002).

Internally, it has two independent data memory spaces, the X-data and Y- data memory spaces, and one program memory space. Having two separate data memory spaces allows a natural partitioning of data for DSP operations and facilitates the execution of the algorithm. For example, in graphics applications data can be stored as X and Y data, in FIR filtering it is stored as coefficients and data, and in FFT as real and imaginary. During program execution, pairs of data samples can be fetched or stored in internal memory simultaneously in one cycle. Externally, the two data spaces are multiplexed into a single data bus. This somewhat reduces the benefits of the dual internal data memory. The arithmetic units consist of two 56-bit accumulators and a single cycle, fixed-point hardware multiplier-accumulator (MAC). The MAC accepts 24-bit inputs and produces a 56-bit product. The 24-bit word length provides sufficient accuracy for representing most DSP variables while the 56-bit accumulator (including eight guard bits) prevents arithmetic overflows. These word lengths are adequate for most applications, including digital audio, which imposes stringent requirements. The 5600x processors provide special instructions that allow zero-overhead looping and bit reversed addressing capability for scrambling input data before FFT or unscrambling the fast Fourier transformed data.

Download Binils Android App in Playstore Download Photoplex App

Analog Devices ADSP2lxx is another family of second generation fixed- point DSP processors with two separate external memory spaces - one holds data only, and the other holds program code as well as data. A simplified block diagram of the internal architecture of the ADSP2lxx is depicted in Figure 14.

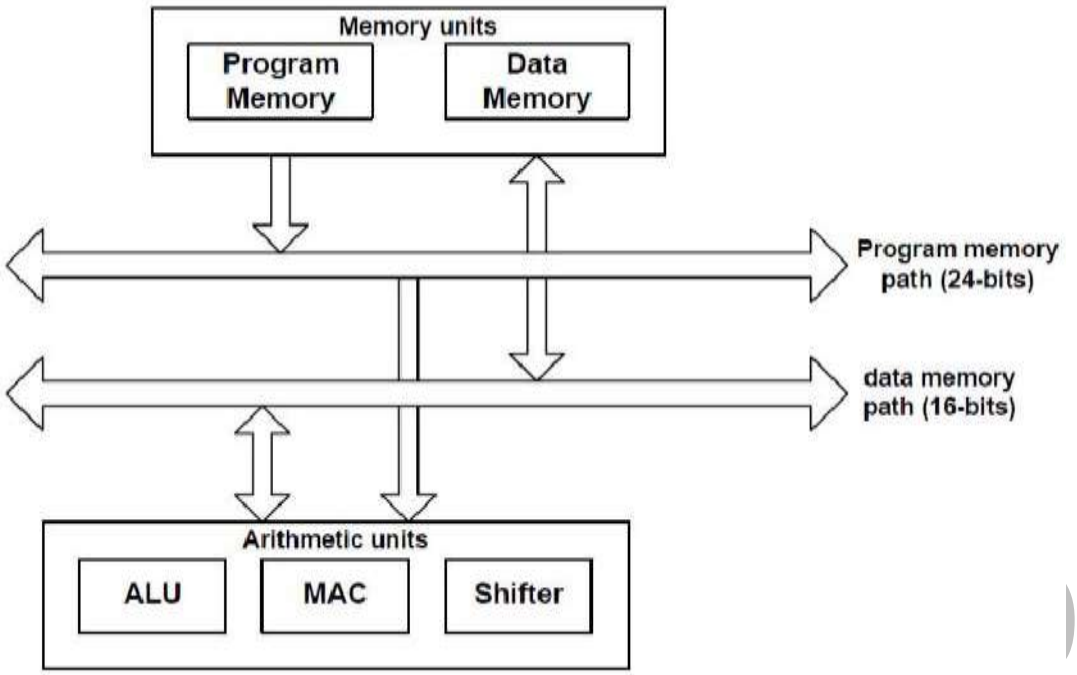


Figure 14. A simplified architecture of a second generation fixed-point DSP (Analog Devices ADS P2100).

The main components are the ALU, multiplier--accumulator, and shifters. The MAC accepts 16 x 16-bit inputs and produces a 32- bit product in one cycle. The accumulator of the ADSP2lxx has eight guard bits which may be used for extended precision. The ADSP2 1 xx departs from the strict Harvard architecture, as it allows the storage of both data and program instructions in the program memory. A signal line (data access signal) is used to indicate when data and not program instructions are being fetched from the program memory. Storage of data in the program memory inhibits a steady data flow through the CPU as data and instruction fetches cannot occur simultaneously. To avoid a bottleneck, the ADSP2 I xx family has an on-chip program memory cache which holds the last 16 instructions executed. This eliminates the need, especially when executing program loops, for repeated instruction fetches from program memory. The ADSP2lxx provides special instructions for zero-overhead looping and supports a bit-reversing addressing facility for FFT. The processor family has a large on-chip memory (up to 64 Kbytes of internal RAM is provided for increased data transfer). The processor has an excellent support for DMA. Without processor intervention the external devices can transfer data and instructions to or from the DSP processor. RAM Lucent Technologies' DSP l6xx family of fixed-point DSPs (see Figure 15) is targeted at the telecommunications and modem market.

Download Binils Android App in Playstore Download Photoplex App

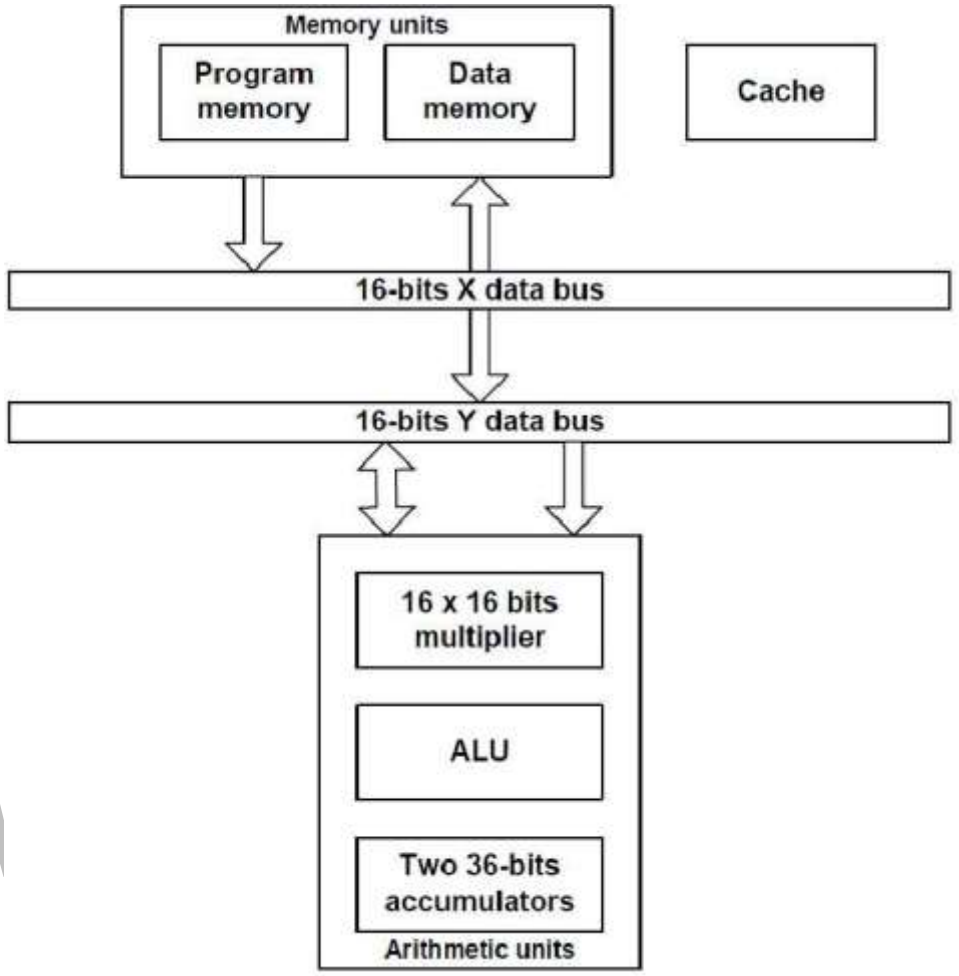
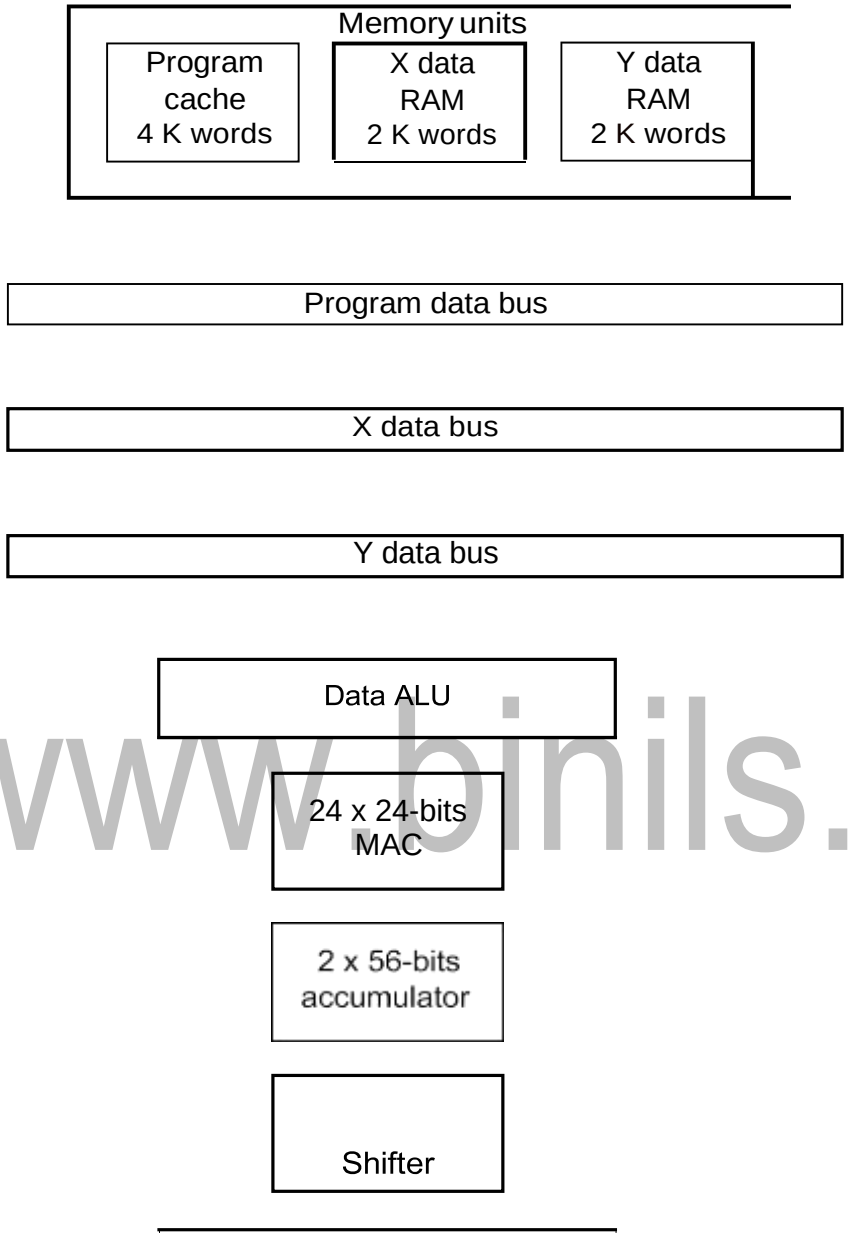


Figure 15. A simplified architecture of Lucent Technologies' DSP l6xx fixed-point DSP.

In terms of computational performance, it is one of the most powerful second generation processors. The processor has Harvard architecture, and like most of the other second generation processors, it has two data paths, the X and Y data paths. Its data arithmetic units include a dedicated 16 x 16- bit multiplier, a 36-bit ALU/shifter (which includes four guard bits) and dual accumulators. Special instructions such as those for zero-overhead single and block instruction looping are provided.

Third generation fixed point DSPs are essentially enhancements of second generation DSPs. In general, performance enhancements are achieved by increasing and/or making more effective use of available On-Chip resources. Compared to the second generation DSPs, features of the third generation DSPs include more data paths (typically three compared to two in the second generation), wider data paths, larger on-chip memory and instruction cache and in some cases a dual MAC. As a result, the performance of third generation DSPs is typically two or three times superior to that of the second generation DSP processors of the same family. Simplified architectures of three third generation DSP processors, TMS320C54x, DSP563x and DSPI6000, are depicted in Figures 16, 17 and 18.

Download Binils Android App in Playstore Download Photoplex App



www.binils.com

Figure 17. A simplified architecture of a third generation fixed-point DSP (Motorola DSP56300).

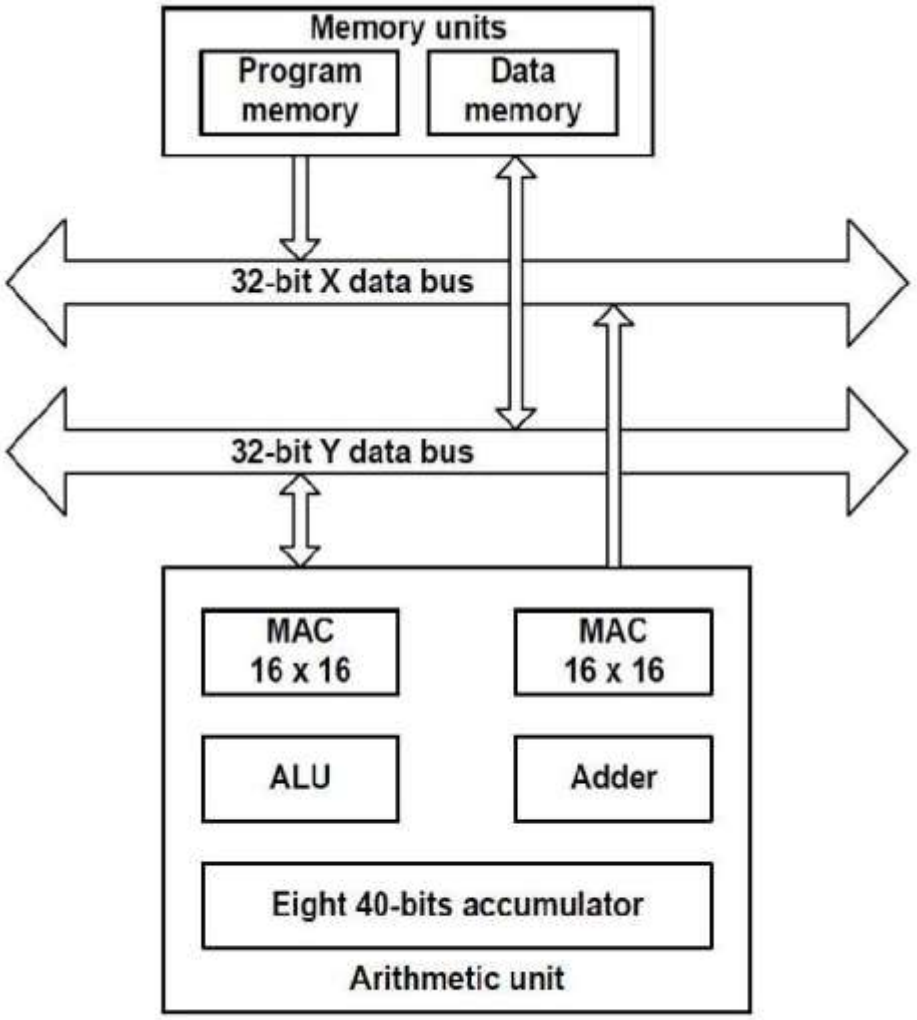


Figure 18. A simplified architecture of a third generatIon fixed-point DSP (Lucent Technologies DSP 16000).

Most of the third generation fixed point DSP Processors are aimed for the applications in digital communication and digital audio, reflecting the enormous growth and influence of these application areas on DSP processor development. Thus some of the processors are featured to support these applications. In the third generation processors, semiconductor manufacturer have also taken the issue of power consumption seriously because of its application in portable and hand held devices.

Fourth generation fixed point processors with their new architectures are primarily aimed at large and/or emerging multi channel applications, such as digital subscribers loop, remote access server modems, wireless base stations third generation mobile systems and medical imaging. The new fixed point architecture that has attracted a great deal of attention in the DSP community is the very long instruction word (VLIW). The new architecture makes extensive use of parallelism while retaining some of the good features of previous DSP processors. Compared to previous generations, fourth generation fixed point DSP processors, in general, have wider instruction words, wider data paths,

Download Binils Android App in Playstore Download Photoplex App

more registers, larger Instruction cache and multiple arithmetic units, enabling them to execute many more instructions and operations per cycle.

Texas Instruments' TMS320C62x family of fixed-point DSP processors is based on the VLIW architecture as shown in Figure 19.

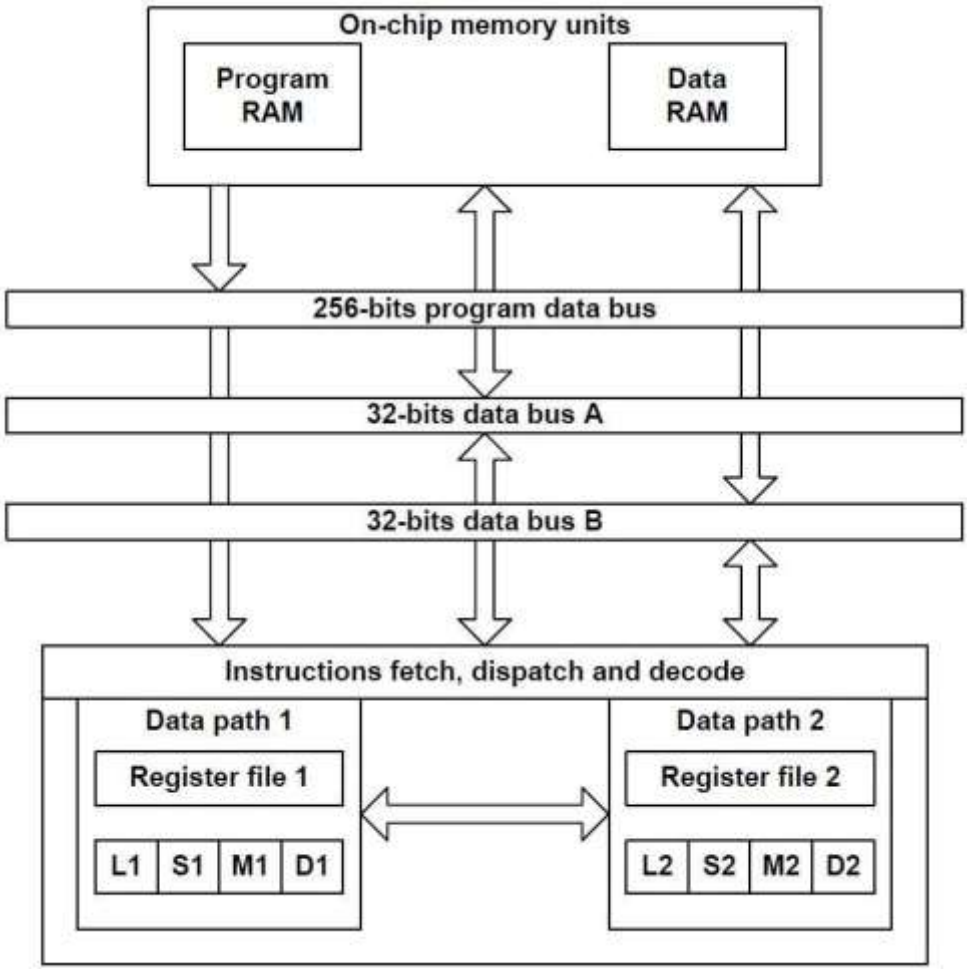


Figure 19. A simplified architecture of a fourth generation fixed-point, very long instruction word, DSP processor (Texas Instruments TMS320C62x). Note the two independent arithmetic data paths, each with four execution units -L1, S1, M1 and D1; L2, S2, M2 and D2.

The core processor has two independent arithmetic paths, each with four execution units - a logic unit (Li), a shifter/logic unit (Si), a multiplier (Mi) and a data address unit (Di). Typically, the core processor fetches eight 32- bit instructions at a time, giving an instruction width of 256 bits (and hence the term very long instruction word). With a total of eight execution units four in each data path, the TMS320C62x can execute up to eight instructions in parallel in one cycle. The processor has a large program and data cache memories (typically, 4 Kbyte of level 1 program/data caches and 64 Kbyte of level 2 program/data cache). Each data path has its own register file (sixteen 32-bit registers), but can also access registers on the other data path. Advantages of VLIW architectures include simplicity and high computational performance. Disadvantages include increased program memory usage (organization of codes to match the inherent parallelism of the processor may lead t

Download Binils Android App in Playstore Download Photoplex App

inefficient use of memory). Further, optimum processor performance can only be achieved when all the execution units are busy which is not always possible because of data dependencies, instruction delays and restrictions in the use of the execution units. However, sophisticated programming tools are available for code packing, instruction scheduling, resource assignment, and in general to exploit the vast potential of the processor.

Circular Addressing Mode:

Many algorithms such as convolution, correlation and finite impulse response (FIR) filters can use circular buffers in memory to implement a sliding window, which contains the most important data to be processed. The C5X supports two concurrent circular buffers operating via the ARs. The following 5 memory-mapped registers control the circular buffer operation:

CBSR1- Circular buffer 1 start register
CBSR1- Circular buffer 2 start register
CBER1- Circular buffer 1 end register
CBER2- Circular buffer 2 end register
CBCR1- Circular buffer control register

Applications of DSP

Here are some of the DSP applications in various fields

In the field of voice processing

- Speech coding and decoding
- Speech recognition
- Speech synthesis
- Digital vocoder

www.binils.com

In the field of musical and sound processing

- Digital music synthesis
- Musical sound processing for recording

In audio/video processing field

- Digital radio
- Digital television

In the field of communication

- DTMF generation and detection
- RADAR

Speech Processing

The speech signal is a slowly time varying signal. The speech signal can be broadly classified into voiced and unvoiced signal. The voiced signals are periodic in nature and unvoiced signals are random in nature. For representing a characteristic sound of the speech the voiced signals will have a fundamental frequency in a segment of 15 to 20 msec. The various frequency components of sounds in speech signal lies within 4 kHz.

The DSP based speech processing techniques can be classified into two broad categories viz.,

speech analysis and speech synthesis.

Speech analysis: In general, the process of extracting the features of speech, then coding or directly digitalizing the speech and then reducing the bit rate is called speech analysis.

Speech synthesis: In general, the process of decoding the speech signal represented in the form of codes is called speech synthesis. It is used in the conversion of speech into text.

Speech Coding and Decoding

The speech coding is digital representation of speech using minimum bit rate without affecting the voice quality. The speech decoding is the conversion of digital speech data to analog speech The old method for quality transmission and reception of digital speech signal through telephone lines, employs a bit rate of 64 kbps (kilo bits per second). This digital representation is called Pulse Code Modulation (PCM) in which the speech signal is sampled at 8 kHz and each sample is quantized to 13 bits and then compressed to 8 bits using μ law or A-law standards to achieve a transmission rate of 64 kbps(8000 samples per secondx8 bits per sample = 64000 bits per second) needed for transmission.

For effective utilization the transmission channels and to reduce memory requirements for storage and retrieval of speech, a number of digital speech coding techniques are developed to represent the speech at lower bit rates up to 1000 bits per second. The speech coding techniques can be broadly classified into waveform coding techniques and parametric coding techniques.

www.binils.com

Some of the popular waveform coding techniques employed are Adaptive Pulse Code Modulation (APCM), Differential Pulse Code Modulation (DPCM) and Adaptive Differential Pulse Code Modulation (ADPCM).

Some of the parametric methods of speech coding are Linear Prediction Coding (LPC), Mel-Frequency Cepstrum Coefficients (MFCC), Code Excited Linear Prediction Coding (CELP) and Vector Sum Excited Linear Prediction (VSELP).

Adaptive Differential Pulse Code Modulation (ADPCM):

The DPCM (Differential Pulse Code Modulation) method of speech analysis/coding is based on the assumption that a speech sample can be effectively represented by the difference between previous and current sample. In the ADPCM method, the difference signal is computed between an adaptively predicted sample and current sample. Since, the difference between two samples can be represented by fewer bits, a 2:1 compression can be achieved, so that a 64 kbps speech signal can be coded to 32 kbps signal.

The ADPCM method of speech coding is shown in figure 24 . The analog speech signal is converted into 64kbps digital speech signal by sampling at 8 kHz with 8-bit per sample using an audio codec. The ADPCM algorithm expand this 8-bit samples to 14-bit samples and subtract each expanded sample with an adaptively predicted sample to generate a difference signal which is quantized to 4 bits. The output of the quantizer is the coded speech signal. The adaptively predicted signal is a weighted average of some dequantized difference signals and some predicted samples.

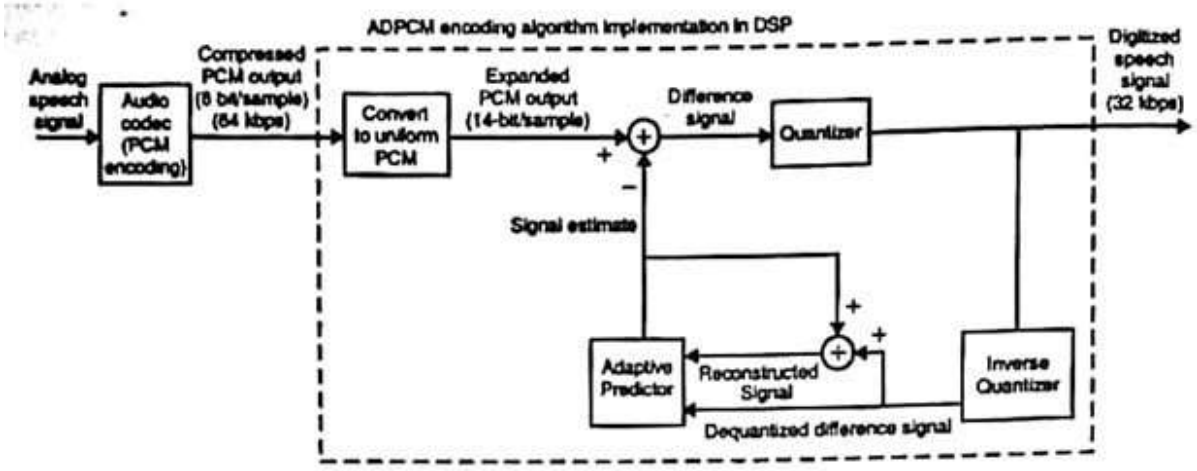


Fig 24. *: ADPCM speech encoder using digital signal processor (DSP).*

The ADPCM method of speech decoding is shown in figure 25. The ADPCM algorithm employs an inverse quantizer to generate the de quantized difference signal, from the coded speech sample. The ADPCM algorithm reconstructs the 14-bit sample of speech by adding the de quantized difference signal and an adaptively predicted signal estimate. Then the 14-bit speech samples are converted to 8-bit samples, which represents the decoded speech. The decoded speech can be converted to analog speech using an audio codec.

www.binils.com

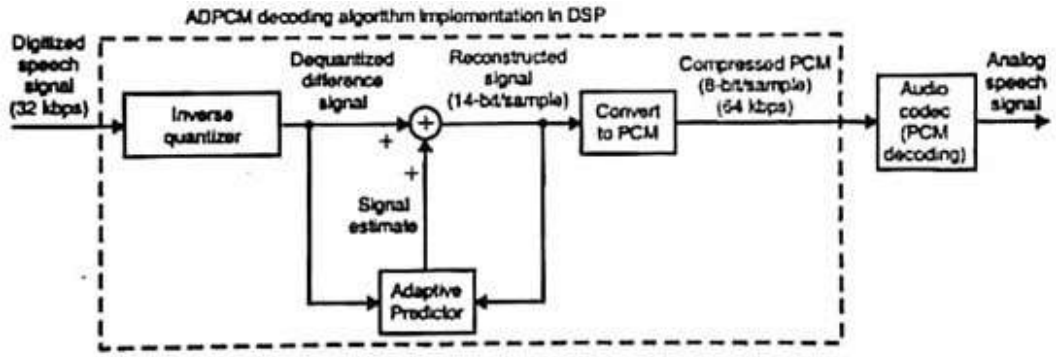


Fig 25 *ADPCM speech decoder using digital signal processor (DSP).*

Linear Predictive Coding (LPC):

The LPC method of speech analysis/coding is based on the assumption that a speech sample can be approximated as a linear combination of previous speech samples. In LPC coding method bit rates up to 24000 bits per second can be achieved.

For speech coding, first the speech signal is digitalized using a coding system, in which the speech signal is segmented to 20msec, sampled at 8 to 12 kHz. Then a set of filter coefficients are determined by using these samples. A pitch is also calculated for each voiced speech segment. The filter coefficients and the pitch represent the coded speech.

Download Binils Android App in Playstore Download Photoplex App