Signal Processing

F6-4

Signal Processing

Signalbehandling

1

Signal Processing

PE 2 ECTS

Course Teacher:

D. M. Akbar Hussain (1 ECTS)

2

Signal Processing

AALBORG UNIVERSITET ESBJERG

Course Objectives:

- To learn some practical facts about DSP architectures. Objective is that you can apply this knowledge to map signal processing algorithms on a digital signal processing hardware.
- To provide you with a global view of embedded micro-architecture implementation options and design methodologies for signal processing.
- The interaction between the algorithm formulation and the underlying architecture that implements the algorithm will be focused.

3

Signal Processing

AALBORG UNIVERSITET ESBJERG

Achievement Through These Objectives:

This course is to compliment your ability to understand both hardware and software for a particular DSP architecture, which hopefully will help you in doing a project in the digital signal processing domain. By the end of this course you will have more clear picture about various signal processing algorithms. How they can be designed and implemented, also you learn the hardware architecture of a DSP system.

Hopefully, you are expected do well with your project development.

4

Material and Books Used for the Course:

- The DSP Handbook: Algorithms, Applications and Design Techniques.
  By: Andrew Bateman & Ian Paterson-Stephens
- Computer Organization & Design: The Hardware/Software Interface.
  By: David A. Patterson & John L. Hennessy
- High Performance Computing.
  By: Kevin Dowd & Charles Severance
- Computer Architecture Software and Hardware.
  By: Richard Y. Kain

Signal Processing
© Dr. D. M. Akbar Hussain

DSP Realization of Digital Filters

Signal Processing

Course Outline & Structure:

Each lecture consists of two 45 minute sessions.
This course will cover the following topics in General, the main focus of the course is DSP Realization of Digital Filters.

1. Description of a General DSP System, GPP, PDSP, ASIC & FPGA.
2. What is signal & Signal Processing, different types of processing.
3. Mapping algorithms on hardware.
4. Fixed Point Processors, different format and description of data with these formats.
5. DSK TMS320C6416 detailed discussion, discussion on McBSP, including all registers.
6. A real signal sampling by TMS320C6416 DSK at various sampling rates.
7. Generation of a corrupted signal using both MATLAB & Code Composer.
8. Application of a signal processing algorithm e.g., Moving Average filter.
9. How can we design a filter for a random signal which is corrupted due to noise.
10. Designing an FIR filter for Implementation, Magnitude Response to determine its attenuation.
11. How can you design filter with MATLAB Signal Processing Tool.

7

Signal Processing

Books & Resources used for preparing these lectures:

1. Teaching Material From Texas Instruments.
2. Multiple Processing: A System Overview By: A. John Anderson.
3. DSP System Design By: Nasser Kehtarnavaz & Mansour Keramat.
4. http://www.dspguru.com/sw/opendsp/index.htm
5. http://www.ti.com/sc/docs/general/dsp/programs/shareware/c2000.htm
6. http://dspvillage.ti.com/
7. McBSP as High Speed Communication Port
8. McBSP Initialization
9. McBSP Used as Digital Controller with Codec
10. McBSP IO2M Interface
11. McBSP Reference Guide

8

Signal Processing

What is a SIGNAL ?

9

Signal Processing

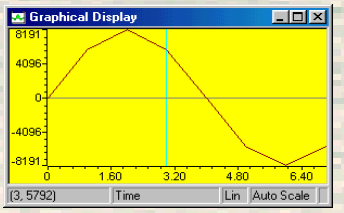
– A *SIGNAL* is a measurement of a physical quantity of certain medium.

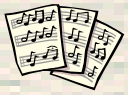
Importantly Signal Contains ***INFORMATION***!

10

Signal Processing

- **Examples of Signals:**
  - Visual patterns (written documents, picture, video, gesture, facial expression)
  - Audio patterns (voice, speech, music)
  - Change patterns of other physical quantities: temperature, EM wave, etc.





Signal Processing

Medium and Modality

- **Medium:**
  - Physical materials that carry the signal.
  - Examples: paper (visual patterns, handwriting, etc.), Air (sound pressure, music, voice), various video displays (CRT, LCD)
- **Modality:**
  - Different modes of signals over the same or different media.
  - Examples: voice, facial expression and gesture.

What is Signal Processing?

Function of Time & Spatial Coordinates

Signal Processing
© Dr. D. M. Akbar Hussain

Types of Processing:

- Transformation
- Filtering
- Detection
- Estimation
- Recognition and Classification
- Coding (compression)
- Synthesis and Reproduction
- Recording, Archiving
- Analyzing, Modeling

Signal Processing Applications

- **Communications:**
  - Modulation/Demodulation (modem)
  - Channel estimation, equalization
  - Channel coding
  - Source coding: compression
- **Imaging:**
  - Digital camera,
  - scanner
  - HDTV, DVD
- **Audio**
  - 3D sound,
  - surround sound
- **Speech**
  - Coding
  - Recognition
  - Synthesis
  - Translation
- **Virtual reality, animation**
- **Control**
  - Hard drive
  - Motor

❑ Digital Signal Processing

- ❑ Signals generated via physical phenomenon are *analog* in that
  - ➢ Their amplitudes are defined over the range of real/complex numbers
  - ➢ Their domains are continuous in time or space.
- ❑ Processing analog signal requires dedicated, special hardware.
- ❑ Digital signal processing concerns processing signals using digital computers.
  - ➢ A continuous time/space signal must be *sampled* to yield countable signal samples.
  - ➢ The real-(complex) valued samples must be quantized to fit into internal word length.

Signal Processing

Signal Processing Systems

The task of digital signal processing (DSP) is to process sampled signals (from A/D analog to digital converter), and provide its output to the D/A (digital to analog converter) to be transformed back to physical signals.

17

Signal Processing

Implementation of DSP Systems

- **Platforms:**
  1. Native signal processing (NSP) with general purpose processors (GPP)
     - Multimedia extension (MMX) instructions
  2. Programmable digital signal processors (PDSP)
     - Media processors
  3. Application-Specific Integrated Circuits (ASIC)
  4. Re-configurable computing with field-programmable gate array (FPGA)

18

Native Signal Processing

- Use GPP to perform signal processing task with no additional hardware.
  - Example: soft-modem, soft DVD player, soft MPEG player.
- Reduce hardware cost!
- May not be feasible for extremely high throughput tasks.
- Interfering with other tasks as GPP is tied up with NSP tasks.
- MMX (multimedia extension instructions): special instructions for accelerating multimedia tasks.
- May share same data-path with other instructions, or work on special hardware modules.
- May use sub-word parallelism to improve numerical calculation speed.
- Implement DSP-specific arithmetic operations, eg. Saturation arithmetic ops.

19

Programmable Digital Signal Processors (PDSPs)

- Micro-processors designed for signal processing applications.
- Special hardware support for:
  - Multiply-and-Accumulate (MAC) ops
  - Saturation arithmetic ops
  - Zero-overhead loop ops
  - Dedicated data I/O ports
  - Complex address calculation and memory access
  - Real time clock and other embedded processing supports.
- PDSPs were developed to fill a market segment between GPP and ASIC:
  - GPP flexible, but slow
  - ASIC fast, but inflexible
- As VLSI technology improves, role of PDSP changed over time.
  - Cost: design, sales, maintenance/upgrade
  - Performance

20

ASIC: Application Specific Integrated Circuits

- Custom or semi-custom IC chip or chip sets developed for specific functions.
- Suitable for high volume, low cost productions.
- Example: MPEG codec, 3D graphic chip, etc.
- ASIC becomes popular due to availability of IC foundry services. ***Fab-less* design** houses innovative design into profitable chip sets using *CAD tools*.
- **Design automation** is a key, enabling technology to facilitate fast design cycle and shorter time to market delay.

21

Signal Processing
© Dr. D. M. Akbar Hussain

Re-configurable Computing using FPGA

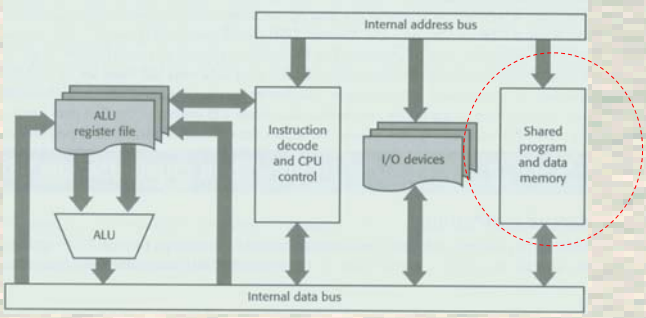
- FPGA (Field programmable gate array) is a derivative of PLD (programmable logic devices).
- They are ***hardware configurable*** to behave differently for different configurations.
- Slower than ASIC, but faster than PDSP.
- Once configured, it behaves like an ASIC module.
- Use of FPGA
  - **Rapid Prototyping:** Run fractional ASIC speed without fabrication delay.
  - **Hardware Accelerator:** Using the same hardware to realize different function modules to save hardware
  - **Low quantity system deployment**

22

Multimedia Signal Processors

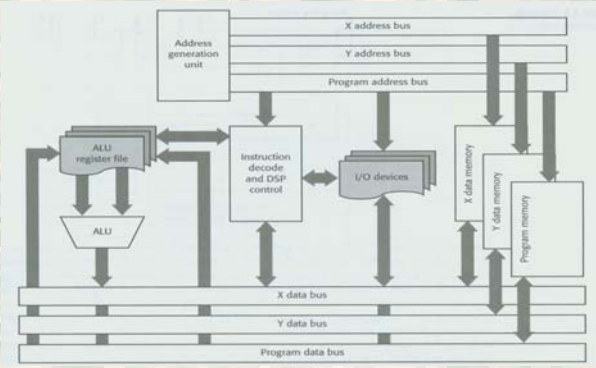
- Specialized PDSPs designed for multimedia applications
- Features:
  - Multi-processing system with a GPP core plus multiple function modules
  - VLIW-like instructions to promote instruction level parallelism (ILP)
  - Dedicated I/O and memory management units.
- Main applications:
  - Video signal processing, MPEG, H.324, H.263, etc.
  - 3D surround sound
  - Graphic engine for 3D rendering

Von-Neumann Architecture



Where is actually the problem ?

Harvard Architecture



Signal Processing
© Dr. D. M. Akbar Hussain

Various Requirements in Design

1. **Real Time**
   - Processing must be done before a pre-specified deadline.
2. **Streamed Numerical Data**
   - Sequential Processing
   - Fast Arithmetic Processing
3. **High Throughput**
   - Fast data input/output
   - Fast manipulation of data

How Fast is Enough for DSP?

- It depends!
- Real time requirements:
  - **Data capture** speed must match sampling rate. Otherwise, data will be lost.
  - Example: In verbal conversation, delay of response can not exceed 50ms end-to-end.
  - Processing must be done by a specific ***deadline,*** *a* constraint on ***throughput.***

Signal Processing
© Dr. D. M. Akbar Hussain

Different throughput rates for processing different signals

- CD music: 44.1 kHz
- Speech: 8-22 kHz
- Video (depends on frame rate, frame size, etc.) range from few hundreds kHz to MHz.

- **Throughput ∝ sampling rate.**

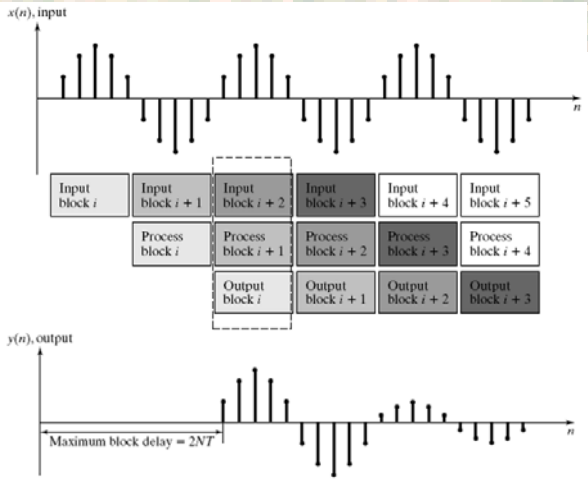
Early Signal Processing Systems

- Implemented with either main frame computer or special purpose computers.
- Batch processing rather than real time, **Streamed Data Processing**.
- Acceleration of processing speed was of main concern.

Different Processing Techniques

- **Stream Processing**
  - Results are Current (updated, within sampling period)
  - Delay between Input/Output (Minimum)
  - Storage of Input & Output (Minimum)
- **Block Processing**
- **Vector Processing**

Block Processing



Block Processing

In block processing incoming samples are first stored in a memory buffer, after N samples have arrived the entire block of data samples are processed at once to produce the output signal. Therefore, block processing occurs for every NT seconds. So 3 activities occurs simultaneously;

1. Processing of N samples in the current block, i + 1
2. Collection and storage of input samples for the next block, i + 2
3. Output of the data samples in the previous block, i

Block Processing

The real time constraints for the block processing shown above are;

$T_b \leq NT - T_o$, T_b is the computational time taken for block processing, T is the sampling interval (time) and T_o is the overhead time for block processing (with modern DMA it can be very small).

Advantages of Block Processing:

1. With a slower processor you can keep up with input sampling, these samples can be buffered.
2. It reduces the over head of read/write operation to memory, as it performs IO operations every NT seconds rather than T seconds.

However, biggest disadvantage is the delay 2NT and the need of additional storage (memory). For example 4N memory locations are required if we use a double buffer for input and output data.
Also, there is switching between processing and storage operations as the processor has to process the block data before the next block and also it needs to store the input and output at right locations.

Vector Processor

A **vector processor**, or **array processor**, is a central processing unit (CPU) that implements an instruction set containing instructions which operate on one-dimensional arrays of data called *vectors*. This is in contrast to a scalar processor, whose instructions operate on single data items. The vast majority of CPUs are scalar.

Basically vector processor takes in multiple inputs at any given time and process it. A simple example could be stereo input, taking both left and right channel at the same time and can undergo for same or different processing.

Vector Processing

Basically, vector processors take the concept of instruction pipelining one step further. Instead of pipelining just the instructions, they also pipeline the data itself.

Vector processor fed instructions for example which say not just to add A to B, but to add all of the numbers "from here to here" to all of the numbers "from there to there". Instead of constantly having to decode instructions and then fetch the data needed to complete them, it reads a single instruction from memory, and "knows" that the next address will be one larger than the last. This allows for significant savings in decoding time.

Key approach:

- Faster hardware
- Faster algorithms

- **Faster algorithms**
  - Reduce the **number of arithmetic operations**.
  - Reduce the **number of bits** to represent each data.
  - Most important example: **Fast Fourier Transform**.

- **Fast Fourier Transform**
  - Reduce the computation to $O(N \log_2 N)$ complex multiplications
  - Makes it practical to process large amount of digital data.
  - Many computations can be "Speed-up" using FFT
  - **Dawn of modern digital signal processing**

Evolution of Micro-Processor

- Micro-processors implemented a central processing unit on a single chip.
- Performance improved from 1MFLOP (1983) to 1GFLOP or above
- Word length (# bits for reg, data, addr, etc) increases from 4 bits to 64 bits today.
- Clock frequency increases from 100KHz to over 2GHz
- Number of transistors increases from 1K to 50M
- Power consumption reduces with lower supply voltage: 5 V drops to 1.5V

37

Signal Processing
© Dr. D. M. Akbar Hussain

VLSI

- The term VLSI, Very Large Scale Integration was emerged in late 1970s.
- Usage of VLSI:
  - Micro-processor
    - General purpose
    - Programmable DSP
    - Embedded µ-controller
  - Application-specific ICs
  - Field-Programmable Gate Array (FPGA)
- Impacts:
  - Design methodology
  - Performance
  - Power
- Characteristics
  - High density:
    - Reduced feature size: 0.25µm -> 0.16 µm
    - % of wire/routing area increases
  - Low power/high speed:
    - Decreased operating voltage: 1.8V -> 1V
    - Increased clock frequency: 500 MHz-> 1GH.
  - High complexity:
    - Increased transistor count: 10M transistors and higher
    - Shortened time-to-market delay: 6-12 months

38

Design Issues

- Given a DSP application, **which implementation option** should be chosen?
- For a particular implementation option, **how to achieve optimal design**?
  - Optimal in terms of what criteria ?
- Software design:
  - NSP/MMX, PDSP/MSP
  - Algorithms are implemented as programs.
  - Often still require programming in assembly level manually
- Hardware design:
  - ASIC, FPGA
  - Algorithms are directly implemented in hardware modules.
- S/H Co-design: System level design methodology.

Signal Processing
© Dr. D. M. Akbar Hussain

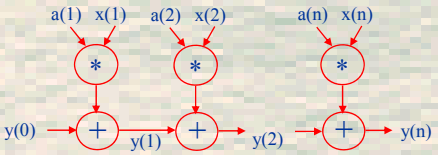
Design Process Model

- Design is the process that **links *algorithm* to *implementation***
- Algorithm
  - Operations
  - Dependency between operations determines a partial ordering of execution
  - Can be specified as a *dependence graph*
- Implementation
  - Assignment: Each operation can be realized with
    - One or more instructions (software)
    - One or more function modules (hardware)
  - Scheduling: Dependence relations and resource constraints leads to a *schedule.*

Design Example ...

$$y = \sum_{k=1}^{n} a(k)x(k)$$

- Operations:
  - Multiplication
  - Addition
- Dependency
  - y(k) depends on y(k-1)
  - Dependence Graph:



Program:

```
y(0) = 0
For k = 1 to n Do
  y(k) = y(k-1)+ a(k)*x(k)
End
y = y(n)
```


- **Software Implementation:**
  - Map each * op. to a MUL instruction, and each + op. to a ADD instruction.
  - Allocate memory space for {a(k)}, {x(k)}, and {y(k)}.
  - Schedule the operation by sequentially execute y(1)=a(1)*x(1), y(2)=y(1) + a(2)*x(2).
  - Note that each instruction is still to be implemented in hardware.

- **Hardware Implementation:**
  - Map each * op. to a multiplier, and each + op. to an adder.
  - Interconnect them according to the dependence graph.

Signal Processing

Reality

- Implementation is realized with a hardware.
- However, by using the same hardware to realize different operations at different time (scheduling), we have a software program!
- Bottom line – Hardware/ software co-design. There is a continuation between hardware and software implementation.
- A design must explore simultaneously both best performance and cost trade-off.

43

Signal Processing

- **Matching hardware to algorithm**
  - Hardware architecture must match the characteristics of the algorithm.
  - Example: ASIC architecture is designed to implement a specific algorithm, and hence can achieve superior performance.
- **Formulate algorithm to match hardware**
  - Algorithm must be formulated so that they can best exploit the potential of architecture.
  - Example: GPP, PDSP architectures are fixed. One must formulate the algorithm properly to achieve best performance, **e.g. Minimize number of operations.**

44

Signal Processing

AALBORG UNIVERSITET ESBJERG

Algorithm Reformulation

- Matching algorithm to architectural features
  - Similar to optimizing assembly code
  - Exploiting equivalence between different operations
- Reformulation methods
  - Equivalent operation with a particular representation:
    - a*2 is the same as left-shift a by 1 bit in binary representation
  - Algorithmic level equivalence
    - Different filter structures implementing the same specification!
- Exploiting parallelism
  - Regular iterative algorithms and loop reformulation
    - Well studied in parallel compiler technology
  - Signal flow/Data flow representation
    - Suitable for specification of pipelined parallelism

45

Signal Processing

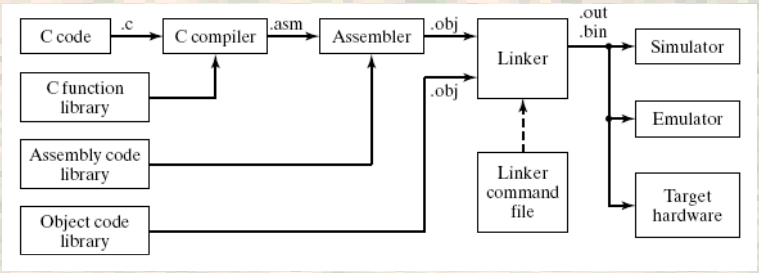
AALBORG UNIVERSITET ESBJERG

Mapping Algorithm to Architecture

- Scheduling and Assignment Problem
  - Resources: hardware modules, and time slots
  - Demands: operations (algorithm), and throughput
- Constrained optimization problem
  - Minimize resources (objective function) to meet demands (constraints)
- For regular iterative algorithms and regular processor arrays -> algebraic mapping.
- Irregular multi-processor architecture:
  - linear programming (is a mathematical method for determining a way to achieve the best outcome)
  - Heuristic methods (refers to experience-based techniques for problem solving, learning, and discovery)
  - Algorithm reformulation for recursions.
- Instruction level parallelism
  - MMX instruction programming
  - Related to optimizing compilation.

46

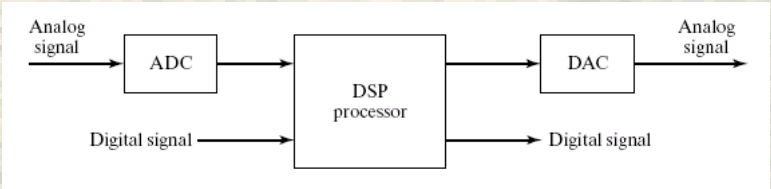
Signal Processing



Block Diagram & Relationship of Software Development

47

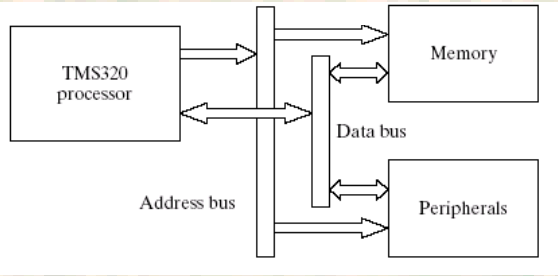
Signal Processing



Typical DSP System

48

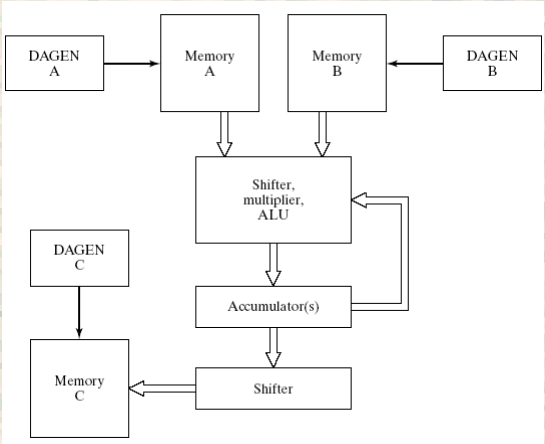
Signal Processing



External Interfaces for TMS320 DSP

49

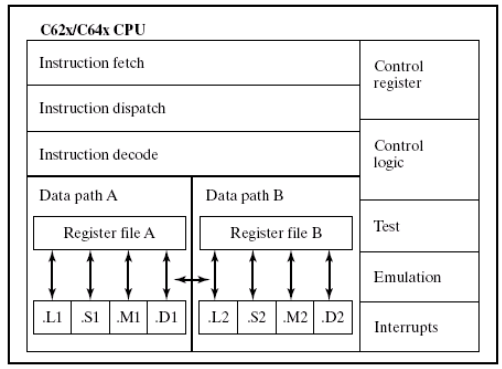
Signal Processing



Data Computation Unit for a Generic DSP architecture

50

Signal Processing



TMS320C62x/C64x CPU Core

51

Signal Processing

8 – Units means DSP can issue/execute 8, 32 bit instruction per clock cycle.

This also means that if the clock frequency is 200 MHz, it can perform 1600 MIPS.

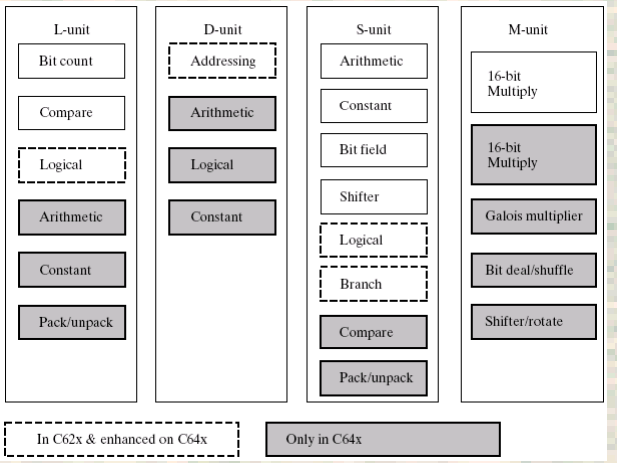
TMS320C6416 also have XOR, AND-NOT logical operations.

Modified Harvard Architecture.

Inside a Bus of 256 bits is constructed to hold 8, 32 bit instructions.

52

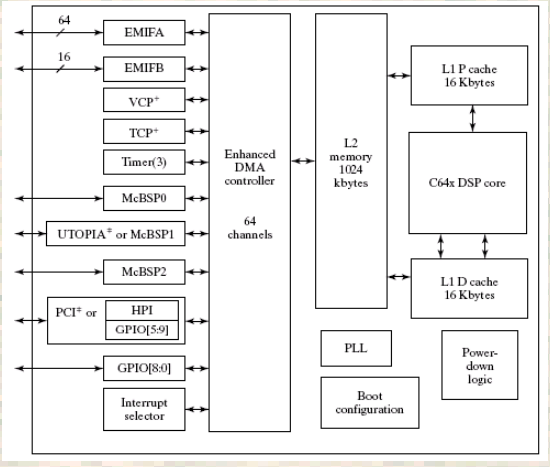
Signal Processing



Functional Units for 62x & 64x

53

Signal Processing

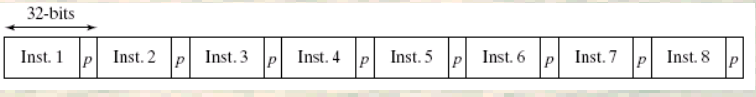


Block Diagram of TMS320C64x

54

Optional field	Optional field	Opcode field	Optional field	Operand field with one to four operands	Comment field
| |	[B1]	MPY	.M	A0, A1, A2	; multiply

Assembly Language Format



Basic Format of Fetch Packet

Signal Processing

Pipelining of TMS320C6416:

Fetch packet	Clock cycle 1	2	3	4	5	6	7	8	9	10	11	12	13
n	PG	PS	PW	PR	DP	DC	E1	E2	E3	E4	E5		
n+1		PG	PS	PW	PR	DP	DC	E1	E2	E3	E4	E5	
n+2			PG	PS	PW	PR	DP	DC	E1	E2	E3	E4	E5
n+3				PG	PS	PW	PR	DP	DC	E1	E2	E3	E4
n+4					PG	PS	PW	PR	DP	DC	E1	E2	E3
n+5						PG	PS	PW	PR	DP	DC	E1	E2
n+6							PG	PS	PW	PR	DP	DC	E1
n+7								PG	PS	PW	PR	DP	DC
n+8									PG	PS	PW	PR	DP
n+9										PG	PS	PW	PR
n+10											PG	PS	PW

Signal Processing

© Dr. D. M. Akbar Hussain

Pipelining of TMS320C6416:

- **Fetch:** PG, PS, PW & PR.
- **Decode:** DP & DC.
- **Execute:** E1, E2, E3, E4 & E5.

Signal Processing

PG: Program address generation phase, it computes the next sequential fetch-packet address or branch instruction.

PS: Program address send phase, it sends the program address to memory.

PW: Program address ready wait phase, it waits until either a memory access or a tag compare is completed.

PR: Program fetch packet receive phase, it receives the fetch packet from memory.

DP: The instruction dispatch phase separates fetch packets into execute packets.

DC: The instruction decode phase decodes source registers destination registers and associated paths.

59

Signal Processing

E1 to E5: The execute stage is divided into **5** phases (stage **7** to **11** in the pipeline). Different types of instructions require various number of clock cycles to complete the operation. Most instruction are completed in phase **E1** and do not require any delay. However, multiply instruction for example requires two stages, therefore, latency of two instruction cycles and a delay of one instruction cycle are introduced.

60

Signal Processing

Critical Factors Determining the Program Execution on a DSP are:

- **Sampling Rate**
- **Complexity of the Algorithm**

61

Signal Processing

DSP Product Design Constraints:

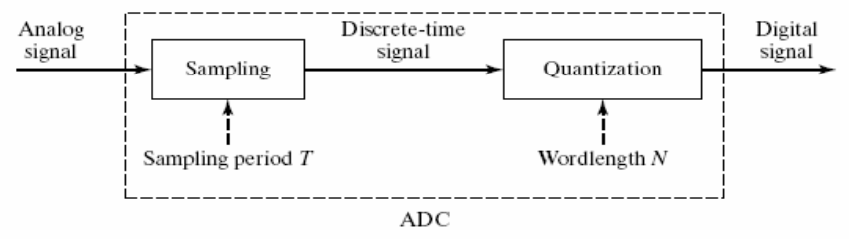
- **Cost of the product.**
- **Cost of the design.**
- **Upgradeability.**
- **System integration.**
- **Power consumption.**

62

Signal Processing

Quantization:

- A process to represent a sampled value by the nearest value which corresponds to an integer scale.
- This process introduce quantization, greater is the gap between discrete true value and the value represented, greater will be the quantized noise.



63

Signal Processing

Sampling & Quantization:

- **Sampling:** Converting a continuous time signal into discrete time signal (still having continuous amplitude at discrete time intervals).
- DSP systems only have Digital Signal (Sampling & Quantization is performed by ADC).
- Amplitude values are converted into digital values by given word length (N) of the DSP (which introduces quantization).

64

Signal Processing

Shannon Sampling Theorem

Sampling period is dependent on the input signal frequency.

To accurately construct a given signal the sampling frequency must be at-least twice the bandwidth.

$$f_s \geq 2\, f_m$$

If Shannon theorem is violated Aliasing will occur.

65

Signal Processing

Aliasing:

Example: Music sampled at 44.1 KHz.

Voice sampled at 8 KHz.

66

Signal Processing

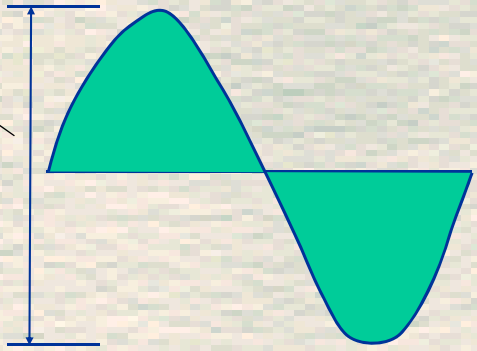
SQNR (Signal to Quantization Noise Ratio):

SQNR = 6N dB

$\Delta = V_{FS} / 2^N$

$V_{FS} = 2$ V, $N = 3$

$\Delta = 0.25$ V



67

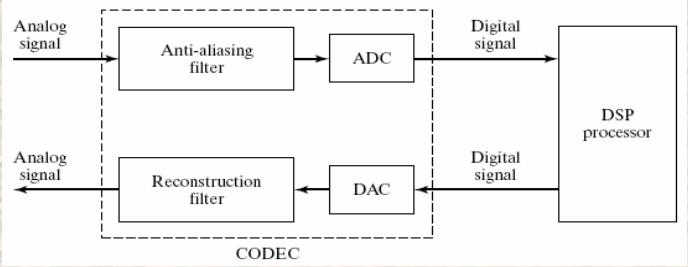
Signal Processing

CODEC:

Actually includes an Anti-Aliasing filter.

Which could be a switched capacitor filter (it can be programmable with different frequencies).

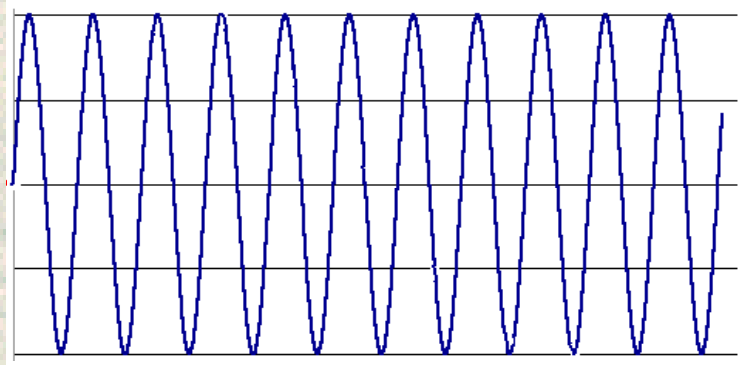
μ – law **A – law**



68

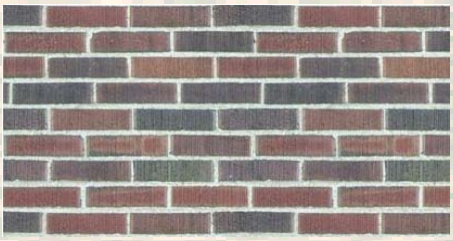
Signal Processing

The following figure illustrates what happens if a signal is sampled at regular time intervals that are slightly less often than once per period of the original signal. The blue curve is the original signal, and the red dots indicate the sampled values.



69

Signal Processing



70

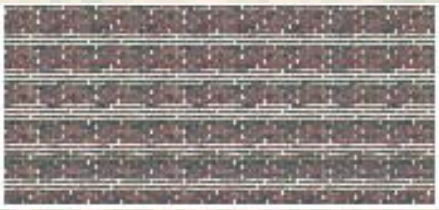
Signal Processing

AALBORG UNIVERSITET ESBJERG

71

Signal Processing

AALBORG UNIVERSITET ESBJERG



72