Name : Nimra Nasir
Reg No: 19-CP-35
Section: Alpha
Date: -Dec-2021
Submitted to: Sir Majid

Digital Signal Processing
Assignment : 1

Questions:

Question: 1

What is meant by digital Signal Processor? Explain architecture of any of the digital Signal processor.

Digital Signal Processor :

A digital signal processor is a specialized microprocessor chip, with its architecture optimized for the operational needs for digital signal processing. DSP's are fabricated on MOS integrated circuits chips. They are widely used in audio signal processing, telecommunication, digital image processing, radar, sonar and speech recognition systems, and in common consumer electronic devices such as mobile phones, disk drives and high-definition television products. The goal of DSP is usually to measure, filter or compress continuous real world analog signal. Most general purpose microprocessor can also execute digital signal processing algorithms successfully, but may not be able to keep up with such processing continuously in real time.

Architecture:

DSP's often use special memory architectures that are able to fetch multiple data or instructions at the same time. The architecture of a DSP is optimized specifically for digital signal processing.

Software Architecture:

By the standards of general purpose processors, DSP instruction sets are often highly irregular; while traditional instruction sets are made up of more general instructions that allow them to perform a wider variety of operations, instruction sets optimized for Digital signal processing contain instructions for common mathematical operations that occur frequently in DSP calculations. Both traditional and DSP-optimized instruction sets are able to compute any arbitrary operation but an operation that might require multiple ARM or x86 instructions to compute might require only one instruction in a DSP optimized instruction set. Even with modern compiler instructions optimizations hand-optimized assembly code is more efficient and many common algorithms involved in DSP calculations are hand written in order to take full advantage of architectural optimizations.

Instruction set:

Multiply-accumulates (MACs, including fused multiply)
* used extensively in all kinds of matrix:
  - convolution for filtering
  - polynomial evaluation
  - Dot product

- FIR Filters
- Fast Fourier Transform

* Related ISA and instructions
  - SIMO
  - VLIW
  - superscalar architecture

* Specialized instructions for modulo addressing in ring buffers and bit reversed addressing mode for FFT cross-referencing:

Data Instructions:

- saturation arithmetic, in which operations that produce overflows will accumulate at the maximum (or minimum) values that the register can hold rather than corapping around doesn't overflow to minimum as in many general purpose CPUs instead it stays at maximum
- Fixed point arithmetic is often used to speed up arithmetic processing.
- Single-cycle operations to increase the benefits of pipelining

Program flow:

- Floating-point unit integrated directly into the data path.
- Pipelined Architecture
- Highly parallel multiplier-accumulators (MAC) unit.
- Hardware controlled looping

Hardware Architecture:

In engineering, hardware architecture refers to the identification of a system's physical components and their relationships. This description, often called a hardware design model, allows hardware designers to understand how their

components fit into a system architecture, provides to software component designers important information needed for software development and integration. The need to effectively model how separate physical components combine to form complex systems is important over a wide range of applications, including (PDAs), cell phones, satellites and submarine

Memory Architecture:

DSPs are usually optimized for streaming data and use special memory architectures that are able to fetch multiple data or instructions at the same time, such as the Harvard architecture or Modified Von Neuman architecture, which are use separate Program and data memories

DSPs can sometimes rely on supporting code to know about cache hierarchies and the associated delays. This is a trade off that allows for better performance. In addition, extensive use of DMA is employed.

Addressing and virtual Memory:

DSPs frequently use multi-tasking operating systems, but have no support for virtual memory or memory protection. operating systems that use virtual memory require more time for context switching among processors, which increases latency.

- Hardware modulo Addressing:
  - Allow circular buffers to be implemented without having to test for wrapping
- Bit reversed addressing, a special addressing mode:
  - useful for calculating FFT's

Question: 2

Apply Convolution Sum on the following sequence

$x[n] = u[n]$

$$h[n] = \begin{cases} 0 & n < 0 \\ a^n & 0 \le n \le N_1 \\ 0 & N_1 < n < N_2 \\ a^{n-N_2} & N_2 \le n \le N_2 + N_1 \\ 0 & n > N_2 + N_1 \end{cases}$$

Solution:

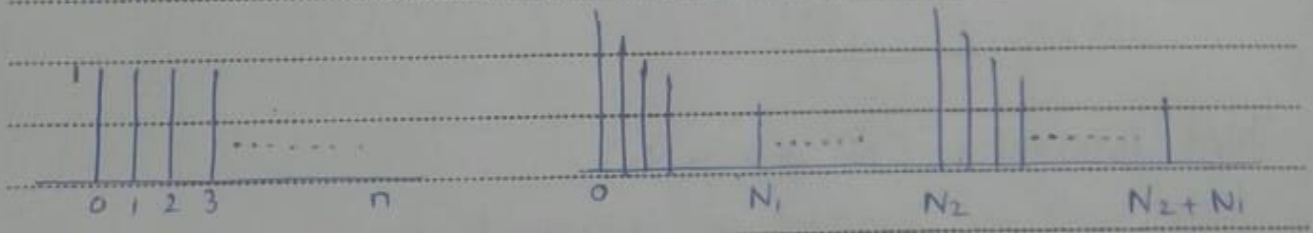
we know that

$$y[n] = x[n] * h[n]$$

$$y[n] = \sum_{n=-\infty}^{\infty} x[k]\, h[n-k]$$

In this $y[n] = \sum_{k=-\infty}^{\infty} h[k]\, x[n-k]$

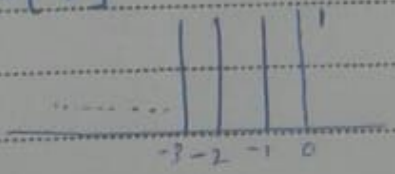
$x[n]$ $h[n]$; where $a = 0.7$



For Convolution :

reflect one signal

$x[-k]$



First Interval : $(n < 0)$

For $n < 0$ there will be no overlapping sample between $x[-k]$ and $h[k]$

$$y[n] = \sum_{n=-\infty}^{0} x[n-k]\, h[k]$$

$$y[n] = 0$$

Second Interval : $(0 \le n \le N_1)$

for $(0 \le n \le N_1)$

overlapping samples will increase

$$y[n] = \sum_{k=-\infty}^{\infty} h[k]\, x[n-k]$$

$$y[n] = \sum_{k=0}^{n} a^k \quad (1)$$

$$y[n] = \sum_{k=0}^{n} a^k$$

$$y[n] = \frac{a^0 - a^{n+1}}{1-a} \qquad \therefore \sum_{k=0}^{N} a^k = \frac{a^0 - a^{N+1}}{1-a}$$

$$y[n] = \frac{1 - a^{n+1}}{1-a}$$

Third Interval : $(N_1 < n < N_2)$

There will be constant no of overlapping samples.

$$y[n] = \sum_{k=-\infty}^{\infty} h[k]\, x[n-k]$$

$$y[n] = \sum_{k=0}^{N_1} a^k \quad (1)$$

Apply formula of geometric series

$$\frac{1 - a^{N_1+1}}{1-a}$$

Fourth Interval: $(N_2 \leq n \; N_2+N_1)$

There will be increasing number of overlapping samples

$$y[n] = \sum_{k=-\infty}^{\infty} h[k]\, x[n-k]$$

$$y[n] = \sum_{k=N_2}^{N_1+N_2} h[k]\, x[n-k]$$

$$y[n] = \sum_{k=0}^{N_1} a^k + \sum_{k=N_2}^{n} a^{(k-N_2)}$$

$$= \frac{a^0 - a^{N_1+1}}{1-a} + \frac{1-a^{n+1}}{1-a}$$

$$= \frac{1-a^{N_1+1}}{1-a} + \frac{1-a^{n+1}}{1-a}$$

$$= \frac{1-a^{N_1+1}+1-a^{n+1}}{1-a}$$

$$= \frac{2-a^{(N_1+1)}-a^{(n+1)}}{1-a}$$

FiFth Interval: $(n > N_2+N_1)$

Constant no of overlapping samples.

$$y[n] = \sum_{k=-\infty}^{\infty} x[n-k]\, h[k]$$

$$= \sum_{k=0}^{N_1} a^k + \sum_{k=N_2}^{N_1+N_2} a^{k-N_2}$$

$$= \sum_{k=0}^{N_1} a^k + \sum_{k=0}^{N_1} a^m$$

$$= \sum_{k=0}^{N_1} a^k + \sum_{k=0}^{N_1} a^k$$

$$= 2\sum_{k=0}^{N_1} a^k$$

∴ let

$m = k - N_2$

$k = 0$

$m = -N_2$

$m = 0$

$k = N_2$

$$= \frac{2\left(1 - a^{(N_1+1)}\right)}{1-a}$$

So;

$$y[n] = \begin{cases} 0 & n < 0 \\ \dfrac{1 - a^{n+1}}{1-a} & 0 \le n \le N_1 \\ \dfrac{1 - a^{N_1+1}}{1-a} & N_1 < n < N_2 \\ \dfrac{2 - a^{N_1+1} - a^{n+1}}{1-a} & N_2 \le n \le (N_1 + N_2) \\ 2\left(\dfrac{1 - a^{N_1+1}}{1-a}\right) & n > (N_1 + N_2) \end{cases}$$

Question: 3

Consider a system with input $x[n]$ and output $y[n]$ that satisfy the difference equation

$$y[n] = ny[n-1] + x[n]$$

The system is casual and satisfies initial-rest conditions, i.e if $x[n] = 0$ for $n < n_0$; then $y[n] = 0$ for $n < n_0$

a) if $x[n] = \delta[n]$, determine $y[n]$ for all n.

Solution:

for $n = 0$;

$y[0] = (0)\, y[0-1] + x[0]$

$= 0 + x[0]$

$= x[0]$ $\quad \because x[n] = \delta[n]$

$= \delta[0] = 1$

For $n > 0$

$y[1] = (1)\, y[1-1] + x[1]$

$y[1] = y[0] + \delta[1]$ $\quad \because x[n] = \delta[n]$

$= 1 + 0 \quad = 1$

$y[2] = 2\, y[2-1] + x[2]$

$= 2\, y[1] + \delta[2]$ $\quad \because x[n] = \delta[n]$

$y[2] = 2(1) + 0 = 2$

$y[3] = 3\, y[3-1] + x[3]$

$= 3y[2] + \delta[3]$ $\quad \because x[n] = \delta[n]$

$y[3] = 3(2) + 0 = 6$

$y[4] = 4\, y[4-1] + x[4]$

$= 4y[3] + \delta[4]$ $\quad \because x[n] = \delta[n]$

$y[4] = 4(6) + 0 = 24$

$y[5] = 5\, y[5-1] + x[5]$

$y[5] = 5\, y[4] + x[5]$

$= 5(24) + \delta[5]$

$y[5] = 120 + 0 \quad = 120$

So; $y[n] = n[n] = n!$ for $n > 0$

NOW for $n < 0$

$y[n] = ny[n-1] + x[n]$

$ny[n-1] = y[n] - x[n]$ $\quad x[n] = \delta[n]$

$y[n-1] = \frac{1}{n}\left[y[n] - x[n]\right]$ $\quad x[-1] = \delta[-1]$

$x[-1] = 0$

$y[n] = \frac{1}{n}\left[y[n+1] - x[n+1]\right]$

So;

$y[-1] = \frac{1}{(-1)}\left[y(-1+1) - x(-1+1)\right] = -1\left[y(0) - x(0)\right] = 0$

$y[-2] = \frac{1}{-2}\left[y(-2+1) - x(-2+1)\right] = \frac{-1}{2}\left[y(-1) - x(0)\right] = -\frac{1}{2}(0) = 0$

$y[-3] = \frac{1}{-3}\left[y(-3+1) - x(-3+1)\right] = \frac{-1}{3}\left[y(-2) - x(-2)\right] = \frac{-1}{3}(0) = 0$

In general, $y[n] = 0$ for $n < 0$

overall result is; $y[n] = n! + 0 = n!$

b) Is the system linear? Justify your answer.

To determine if the system is linear, consider the input

$y[n] = a\,\delta[n] + b\,\delta[n]$

$y[n] = 0$; for $n < 0$

$y[0] = a\,\delta[0] + b\,\delta[0]$

$= a + b$

$y[1] = a + b$

$y[2] = 2(a+b)$

$y[3] = 6(a+b)$

$y[4] = 24(a+b)$

$y[5] = 120(a+b)$

Because the output of the superposition of two input signal is equivalent to the superposition of the individual output of the system is linear.

c) Is the system time invariant? Justify your answer.

$x[n] = \delta[n-1]$

$y[n] = 0$ for $n < 0$

$y[0] = 0$

$y[1] = 1$

$y[2] = 2$

$y[3] = 6$

$y[4] = 24$

using $h[n]$ from part 'a'

$h[n-1] = (n-1)!\,u[n-1] \neq y[n] = x[n] = \delta[n-1]$

So it is not time invariant system.