Class Objective

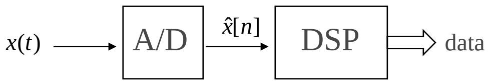
- **When you finish this class you will:**
  - Be familiar with dsPIC DSP features
  - Be familiar with using dsPIC DSP library and tools

Agenda

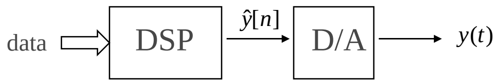
- **Why DSP**
- **dsPIC Support for DSP**
  - dsPIC DSP architecture
  - dsPIC DSP Tools and Libraries

 104 DSP Slide 3

DSP Systems



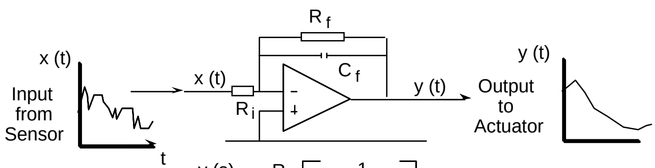
Analysis (Receiver / Encoder)



Synthesis (Transmitter / Decoder)

Signal Conditioning/Control

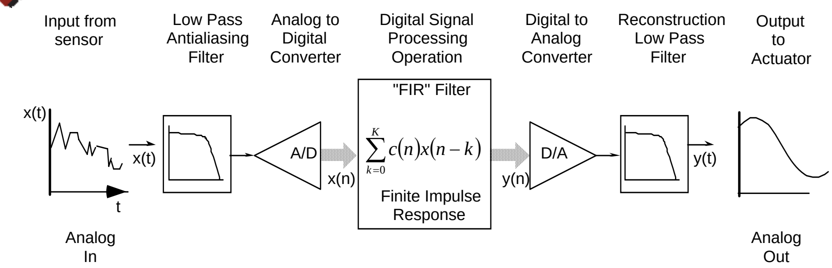
Analog Control Systems



$$\frac{y\,(s)}{x\,(s)} = -\frac{R_f}{R_i}\left[\frac{1}{1 + jw R_f C_f}\right]$$

- Drawbacks:
  - Low noise immunity
  - Little flexibility
  - Requires adjustments
  - Critical component specification, especially for high order filters
- Filter characteristics change with
  - Temperature
  - Component aging
  - Power supply variation

Digital Control Systems



- Some key advantages of digital filtering:
  - Less affected by noise
  - Lower power consumption
  - Programmable systems
  - Minimal effect of drift in characteristics with age

FIR Filters: Definition

- **The input-output relation of a FIR filter is given by**

$$y[n] = \sum_{k=0}^{T-1} b_k x[n-k]$$

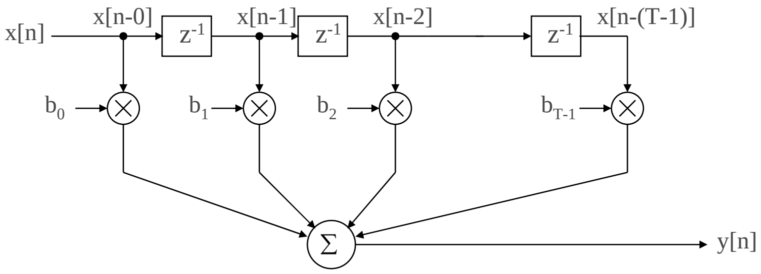
where:

T = filter length (# of taps)

b_k = the k^{th} coefficient of the filter

FIR Filters: Structure

- **FIR filter tapped delay line form**
  - z^{-1} refers to a unit sample delay



dsPIC support for Filters

- DSP Library Support
- Dual Harvard Architecture
- MAC instructions with data prefetch
- Modulo Addressing
- Loop control - DO and REPEAT instructions

 104 DSP Slide 11

C-callable fixed point FIR from MPLAB

Description: `FIR` applies an FIR filter to the sequence of source samples, places the results in the sequence of destination samples, and updates the delay values.

Include: `dsp.h`

Prototype:

```
extern fractional* FIR (
   int numSamps,
   fractional* dstSamps,
   fractional* srcSamps,
   FIRStruct* filter
);
```

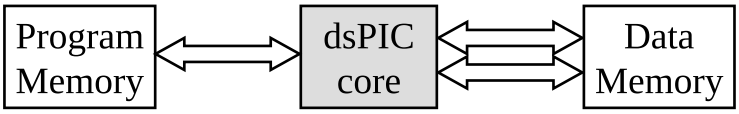
Arguments:

numSamps	number of input samples to filter (also N)
dstSamps	pointer to destination samples (also y)
srcSamps	pointer to source samples (also x)
filter	pointer to `FIRStruct` filter structure

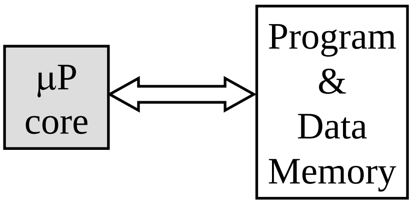
Return Value: Pointer to base address of destination samples.

 104 DSP Slide 12

dsPIC vs. Microprocessor Architectures



Dual Harvard Architecture



Von Neumman Architecture

MAC and dual parallel reads

MAC

Multiply and Accumulate

Syntax: {label:} MAC Wm*Wn, Acc {,[Wx], Wxd} {,[Wy], Wyd} {,AWB}

Operation:
(Acc(A or B)) + (Wm) * (Wn) → Acc(A or B)
([Wx]) → Wxd; (Wx) + kx → Wx
([Wy]) → Wyd; (Wy) + ky → Wy
(Acc(B or A)) rounded → AWB

Description: Multiply the contents of two working registers, optionally prefetch operands in preparation for another MAC type instruction and optionally store the unspecified accumulator results. The 32-bit result of the signed multiply is sign-extended to 40 bits and added to the specified accumulator.

Words: 1

Cycles: 1

One clock cycle, One program word!

Example 1: `MAC W4*W5, A, [W8]+=6, W4, [W10]+=2, W5`

MAC and dual parallel reads

MAC W4*W5, A, [W8]+=2, W4, [W10]-=6, W5, W13

Source operand registers

Destination accumulator

X prefetch source

X prefetch destination

Y prefetch source

Y prefetch destination

Other Acc. Write-back destination

Modulo or “Circular” Addressing

- Eliminates the software overhead associated with effective address (EA) correction
- Data address boundary checks are performed in hardware
- May be used with any instruction that utilizes indirect addressing

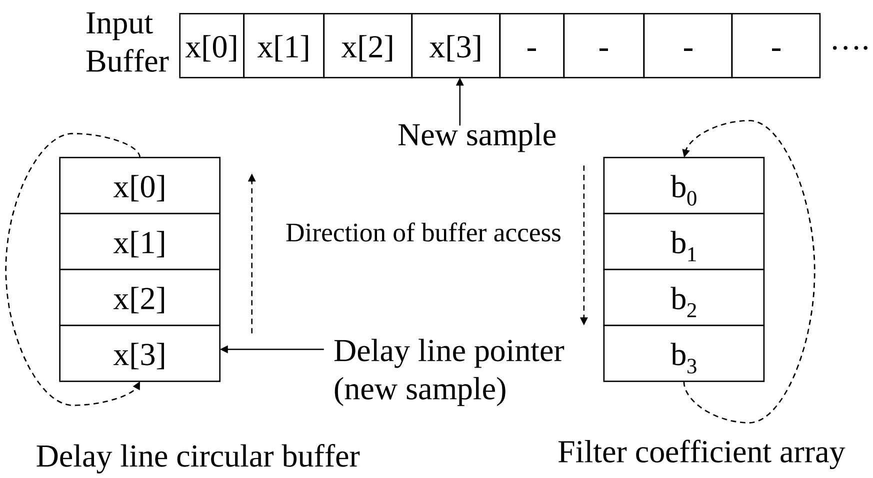
Modulo: FIR Filter Example

- Single Sample FIR Filter Operation (4 taps)
- Vector Dot Product of…
  - Input Delay Line
  - Coefficient Array

$$\underbrace{\begin{bmatrix} x[n] & x[n-1] & x[n-2] & x[n-3] \end{bmatrix}}_{\textbf{Delay line}} \times \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \end{bmatrix} = \boxed{y[n]}$$

Modulo: FIR Filter Example

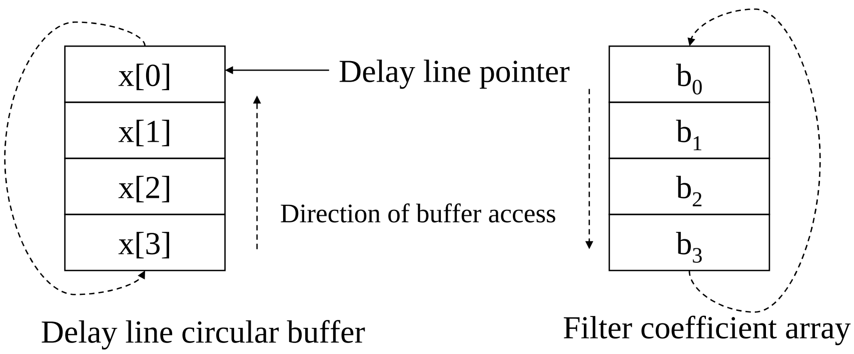
Buffer at n = 3 (before computation):



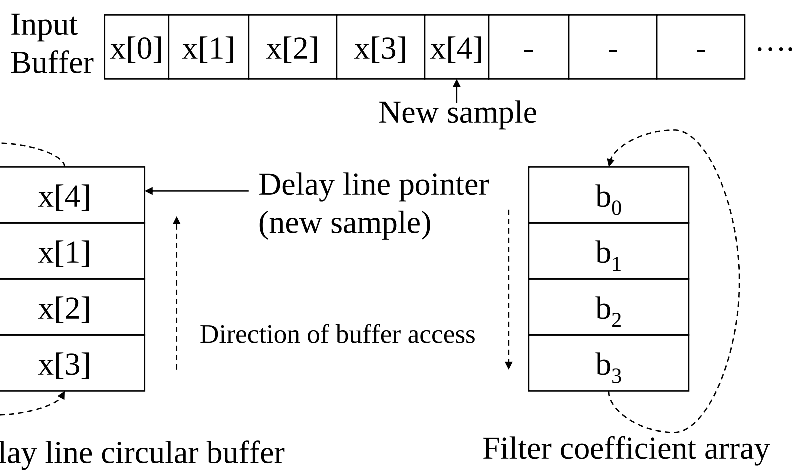
Modulo: FIR Filter Example

FIR output for n = 3:

$y[3] = x[3]*b_0+x[2]*b_1+x[1]*b_2+x[0]*b_3$



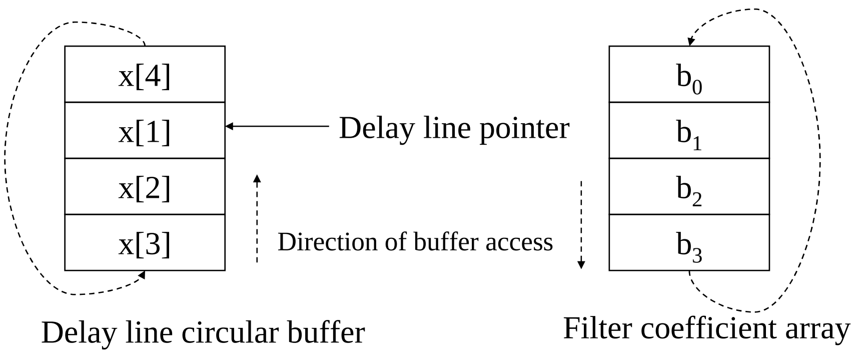
Modulo: FIR Filter Example



Modulo: FIR Filter Example

FIR output for n = 4:

$y[4] = x[4]*b_0+x[3]*b_1+x[2]*b_2+x[1]*b_3$



Zero Overhead Looping

- Two Instructions
  - **REPEAT** for single instruction loops
  - **DO** for multiple instruction loops
- Interruptable
- STATUS register indicates when looping is active

REPEAT Examples

- REPEAT #200
  MOV [W0--], [W1]++
- REPEAT W8
  CLR [W5]++
- REPEAT #0x20
  ADD A, [W3++]

Nested DO Example

```
        DO    LOOP1, #0x100
        MOV   [W0], W1
        AND   #0x3FF, W1
        DO    LOOP2, #0x8
        BTG   W1, #0x9
LOOP2:  IOR   W1, [W2]++, [W3]++
LOOP1:  MOV   W1, [W0]++
```


4 Lines (dsPIC) vs. 31 Lines (16-bit MCU)

dsPIC

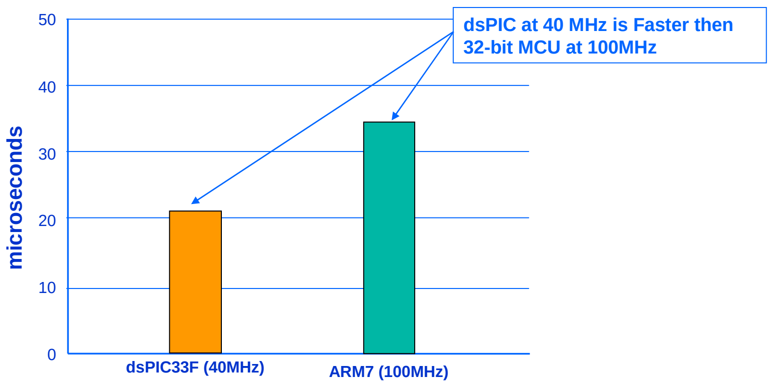
```
repeat   w4
mac      w5*w6,a,[w8]+=2,w5,[w10]+=2,w6
mac      w5*w6,a,[w8]+=2,w5,[w10],w6
mac      w5*w6,a
```

16-bit MCU

```
do_again1:
        mov   R6,[R5]     ;Get next x(n-I)
        mov   R7,[R12+]   ;Get next FIR filter coeff-a(i)
        mul   R6,R7       ;Form product a(I)*x(n-I)
        jmpr  cc_NV,copyl                 ;Test for only 16 bit result
        mov   R9,MDH      ;Move high portion of MD
copy1:
        mov   R6,MDL      ;Move low portion of MD
        add   R11,R6      ;acc. Product terms in R10 and R11
        addc  R10,R9
        sub   R5,#2       ;Point to a next item in x(n-I)
        cmp   R5,R13      ;Chk if ptr falls out of circ buffer table
        jmp   cc_NN,leap1                 ;If still in table keep going otherwise
                                          ;adj. Ptr. To top of table
        mov   R4,R2
        shl   R4,#1
        add   R5,R4       ;adj. Ptr to top of table
leap1:
        add   R1,#1h      ;Determine if all taps computed
        cmp   R1,R2
        jmp   cc_NZ,do_again1
        mov   R4,#curr_ptr
        mov   R5,[R4]
        add   R5,#2h      ;Inc. ptr. Into circ. Buffer
        mov   R3,R2
        shl               R3,#1
        sub               R3,#2
        mov               R6,R13
        add               R6,R3
        cmp               R5,R6           ;Does curr_ptr exceed top of table?
        jmpr              cc_NZ,leap3
        mov               R5,R13          ;Back to beginning of table
leap3:
        cmp               R10,#0h
        jmpr              cc_Z,leap2
```


DSP Benchmark

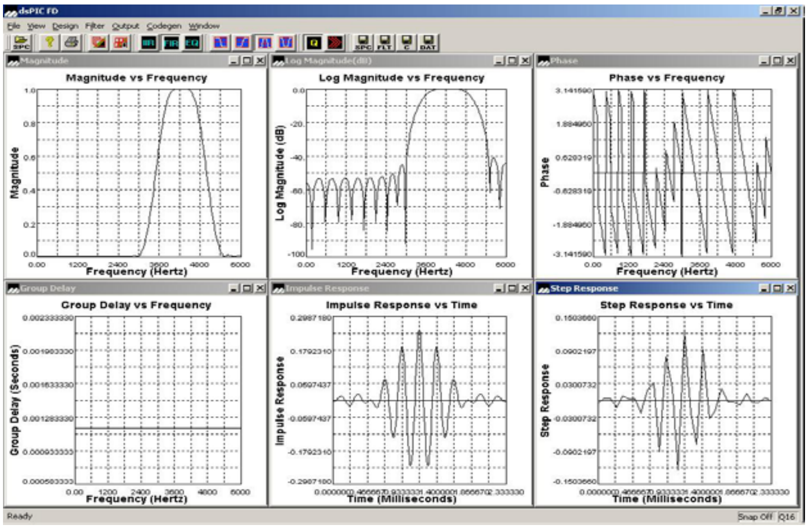
- **BDTI's Real Block FIR Filter Benchmark**
  - Execution time (lower is better)



dspIC® Digital Filter Design

- **Graphically design all types of digital filters**
  - Low-pass, high-pass
  - Band-pass, band-stop
- **Digital filter algorithm kernels are provided for:**
  - FIR—Finite Impulse Response
  - IIR—Infinite Impulse Response
- **Designing coefficients to control the filter response is the tricky part**
  - You can do the math or use this tool!!
- **Quickly observe filters response**
- **Generated code and coefficients fully compatible with MPLAB® C30 C Compiler language tools**

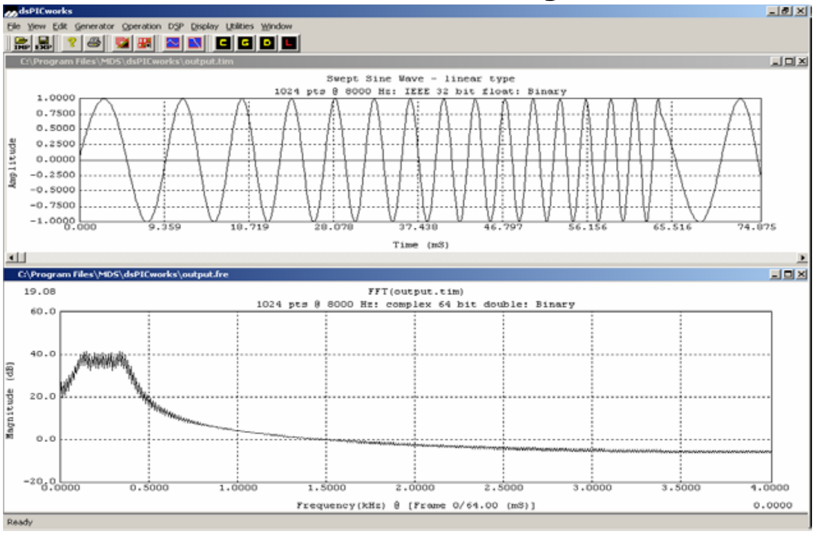
dsPIC® Digital Filter Design



dsPICWorks™ Data Analysis and DSP Software

- **Graphical signal analysis and generation tool**
  - Create waveforms and process them
  - Signal generation
    - **SIN, square, swept SIN, triangular, etc...**
  - Signal operations
    - **Add, Flip and Shift, Multiply, etc...**
  - DSP operations
    - **Filtering, correlation, FFT, LPC analysis, etc...**
- **Designed especially for the dsPIC30F**
  - Import/export data with MPLAB® compatible files
  - Processes filter files saved from dsPIC® Digital Filter Design

dsPICWorks™ Data Analysis



Lab 1

Using DSP Tools and Libraries

Lab 1 – Using DSP

- **Goals**
  - Learn DSP tools and Libraries
- **Lab**
  - Implement signal filtering using FIR fitler

 104 DSP Slide 33

Math Library

- **Supports all of <MATH.H> from ANSI "C"**
  - Trigonometric Functions
  - Exponent and Log Functions
  - Power, Square Root, Floor, Ceiling, etc.
- **IEEE-754 Compliant**
- **Optimized for code size, developed in Assembly**
- **Supports 2 data types**
  - 32-bit floating point
  - 64-bit double
- **Benefits any application which uses floats/doubles**
- **C and assembly callable**

 104 DSP Slide 35

DSP Library

- **Libraries for your DSP requirements**
- **Collection of 49 of the most common DSP functions**
  - Vector and Matrix Functions
    - **Add, Subtract, Multiply, Max, Min**
    - **Convolution, Correlation, Power, etc...**
  - Filter Functions—block and single sample processing
    - **FIR, IIR, Adaptive LMS, Lattice and more**
  - Transform Functions
    - **FFT, Inverse FFT, Discrete Cosine Transform**
  - Window Functions
    - **Bartlett, Blackman, Hamming, Hanning, Kaiser**
- **Most functions are hand coded in assembly language**
  - Fast execution time, C and assembly callable

 104 DSP Slide 36

Embedded Modem Library

- **Software Modem (soft-modem) Support**
  - Replacing external modem chip with software!
- **Benefits**
  - Single-chip integrated solution (up to 14.4 kbps)
  - Quick dialup and connection times
    - **Perfect for small transactions on analog phone line**
- **Application Examples**
  - Home automation, remote access, security systems
  - POS applications, web-enabled devices
  - Tele-metering, remote upgrades
- **Library supplied with full DTMF Generation/Detection modules**

Soft-Modem Resource Summary

- **dsPIC30F implementation leaves room for other tasks...**

ITU-T Specification	Bit Rate (bps)	RAM (Bytes)	Program Memory (Kbytes)	MIPS
V.32bis	14400	3600	36	15
V.32	9600	3200	31	12
V.22bis	2400	1700	22	7
V.23	600/1200	1000	15	4.5
V.21	300	1000	13	4.5
V.42 (Error Corr.)	na	2000	14	1.5

FREE

Includes DTMF Library

Speech Coding Library

- **Performs speech compression / decompression**
- **Encoder - 16:1 compression ratio**
  - Speech input: 8 KHz, 16-bit mono
  - Encoded output: 8kbps data stream
- **Decoder**
  - Decoder input: 8 kbps data stream
  - Speech output: 8 KHz, 16-bit mono
- **Based on Speex**
  - Open source technology
  - Numerous PC plug-ins / apps readily available
  - www.speex.org
- **Includes PC Utility for making “playback” files**

 104 DSP Slide 39

Speech Coding Library How may it be used?

- **By any application with voice communication**
- **Benefits**
  - Reduces communication bandwidth
  - Reduces storage requirements
- **Sample Applications**
  - Digital radios / walkie-talkies
  - Answering machines / voice recorders
  - Playback-only systems
    - **Security Systems (building evacuation)**
    - **Museum Guides**
    - **Application Voice Prompts**

 104 DSP Slide 40

Speech Coding Library Resource Requirements

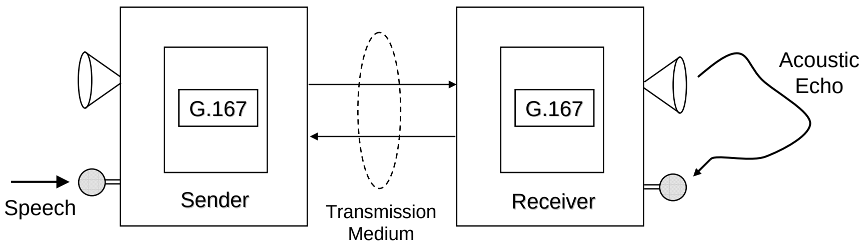
- Small Decoder footprint / Encoder is RAM intensive

	Encoder	Decoder
MIPs	19	3
RAM	5.4 KB	3.2 KB
Flash	33 KB	15 KB

- 2 analog interfaces supported
  - Silicon Labs Si-3000 Codec
  - Alternate interface for cost-sensitive applications
    - ADC for microphone input
    - Output compare for speaker output
    - Sample circuits provided in User's Guide

Acoustic Echo Cancellation Library

- **Provides 'far-end' echo cancellation**
  - Speaker output is picked up by the local microphone and transmitted back to the sender
- **Features**
  - G.167-compliant algorithm (up to 64 msec)
  - Supports full-duplex operation



Acoustic Echo Cancellation Library

- **G.167 CPU Resources**

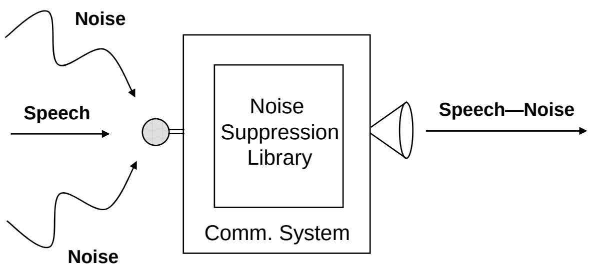
Echo Delay	RAM (KB)	Program Memory (KB)	MIPS
16 msec	4.6*	6	7.5
32 msec	5.4*	6	10.5
64 msec	7.7*	6	16.5

* Includes up to 2KB for CODEC and UART buffers and *A*/*U*-law compression

- **Target Applications**
  - Speakerphone
  - Hands-free car phone
  - Intercom
  - Emergency alarm units

Noise Suppression Library

- **Removes extraneous noise picked up by microphone**
  - Voice activity detector differentiates noise and speech
  - Noise reduction filters adjust every 10 ms during periods of speech inactivity (uses FFT analysis)
  - Speech is continuously filtered, reducing noise



Noise Suppression Library

- **Noise Suppression CPU Resources**
  - 3.3 MIPS
  - 5 Kbytes Flash memory
  - 2 Kbytes RAM
    - **Includes up to 1 Kbyte for buffering Codec data**
- **Target Applications**
  - Front end for any voice based system
    - **Headsets**
    - **Hands-free telephone**
    - **Intercom**
    - **Speech recognition**
    - **Speech coding**

 104 DSP Slide 45

Embedded Encryption Libraries

- **dsPIC30F Symmetric Key Library**
  - Message Digests: SHA-1, MD5
  - Symmetric Key Encryption: T-DES, AES
  - Deterministic Random Bit Generator: ANSI X9.82
- **dsPIC30F Asymmetric Key Library**
  - Signing and Verification: RSA, DSA
  - Public Key Encryption: RSA
  - Key Agreement Protocol: Diffie-Hellman
  - Big Integer Arithmetic Package

 104 DSP Slide 46

Embedded Encryption Applications

- **Secure Web Transactions**
  - Secure web access (SSL/TLS)
  - E-mail (S/MIME), secure XML transactions, and Virtual Private Networks (IPsec)
- **Smart Card Readers**
- **PDAs and other mobile devices**
- **Secure communications between a dsPIC® DSC application and personal computers**
  - Trusted Computing Group (TCG)
  - Microsoft® Next Generation Secure Computing Base (NGSCB)

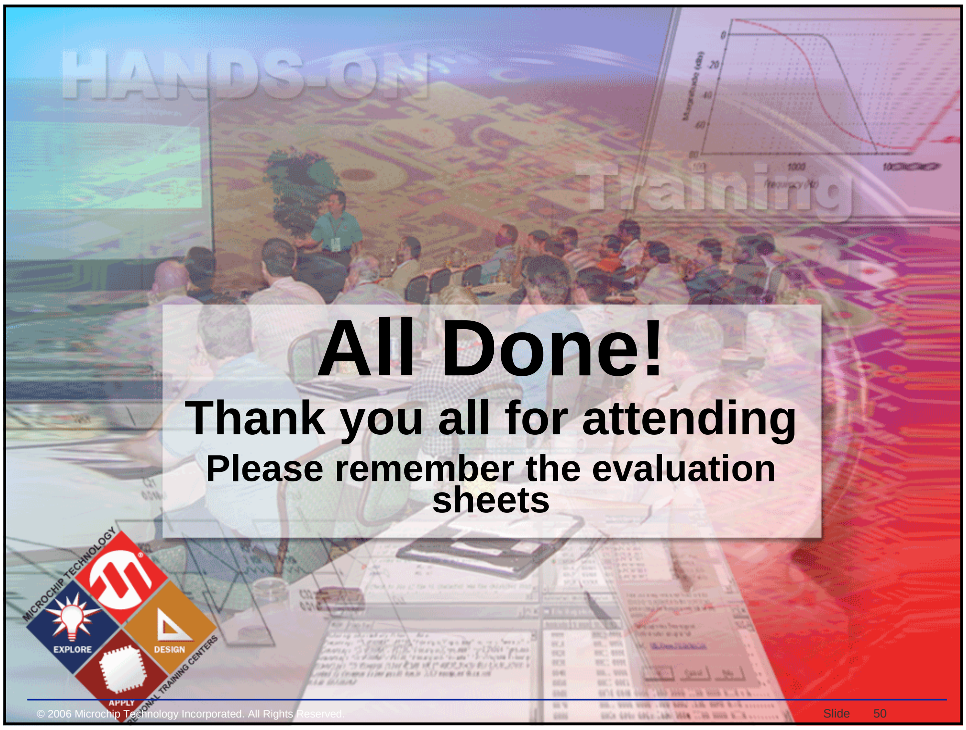
Summary

- We learned advantages of dsPIC DSP architecture
- We learned DSP tools and libraries available for dsPIC

References

- dsPIC30 & dsPIC33 Datasheet
- dsPICDEM1.1 User's Guide
- MPLAB® IDE
- C30 Compiler
- ICD2 In Circuit Debugger

 104 DSP Slide 49



Trademarks

The Microchip name and logo, the Microchip logo, Accuron, dsPIC, KeeLoq, microID, MPLAB, PIC, PICmicro, PICSTART, PRO MATE, PowerSmart, rfPIC and SmartShunt are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

AmpLab, FilterLab, Migratable Memory, MXDEV, MXLAB, SEEVAL, SmartSensor and The Embedded Control Solutions Company are registered trademarks of Microchip Technology Incorporated in the U.S.A.

Analog-for-the-Digital Age, Application Maestro, dsPICDEM, dsPICDEM.net, dsPICworks, ECAN, ECONOMONITOR, FanSense, FlexROM, fuzzyLAB, In-Circuit Serial Programming, ICSP, ICEPIC, Linear Active Thermistor, Mindi, MiWi, MPASM, MPLIB, MPLINK, MPSIM, PICkit, PICDEM, PICDEM.net, PICLAB, PICtail, PowerCal, PowerInfo, PowerMate, PowerTool, REAL ICE, rfLAB, rfPICDEM, Select Mode, Smart Serial, SmartTel, Total Endurance, UNI/O, WiperLock and ZENA are trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

SQTP is a service mark of Microchip Technology Incorporated in the U.S.A.

All other trademarks mentioned herein are property of their respective companies.