SOPC WORLD 2002

The MathWorks

MathWorks Integrated Design Solution

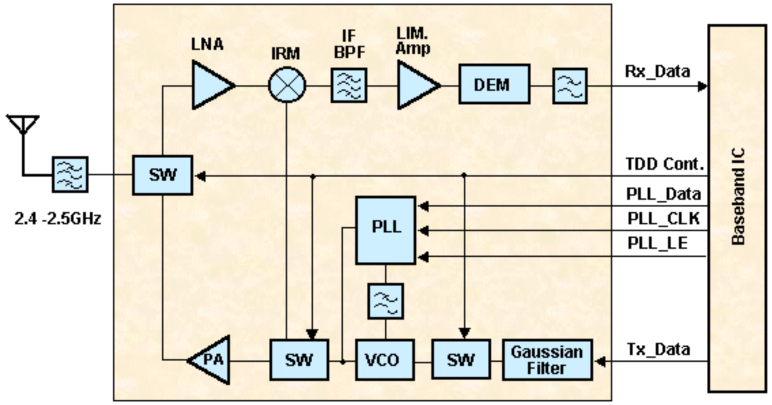
ALTERA.

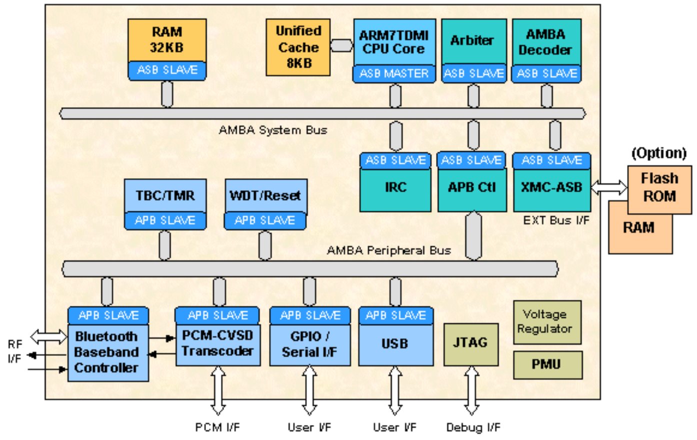
The MathWorks at a Glance

- Founded in 1984, privately held
- Headquarters in Natick, Massachusetts (near Boston)
- European offices in the UK, France, Germany, Switzerland, Italy, Spain, and Benelux region
- Over 1000 employees, including $^1/_3$ in product development
- More than 500,000 users in 100 countries, on all seven continents!

Today's System and IC Design Challenges

- Hardware: RF/Analog, Digital IC, FPGA
- Software: DSP, MAC, Control, Use interface
- Moving partitioning boundaries
  - Analog ↔ Digital IC ↔ FPGA ↔ DSP S/W ↔ Micro controller S/W
- Implementation specific tools lock your IP into one target type
  - Spice, HDL, ASM , C/C++

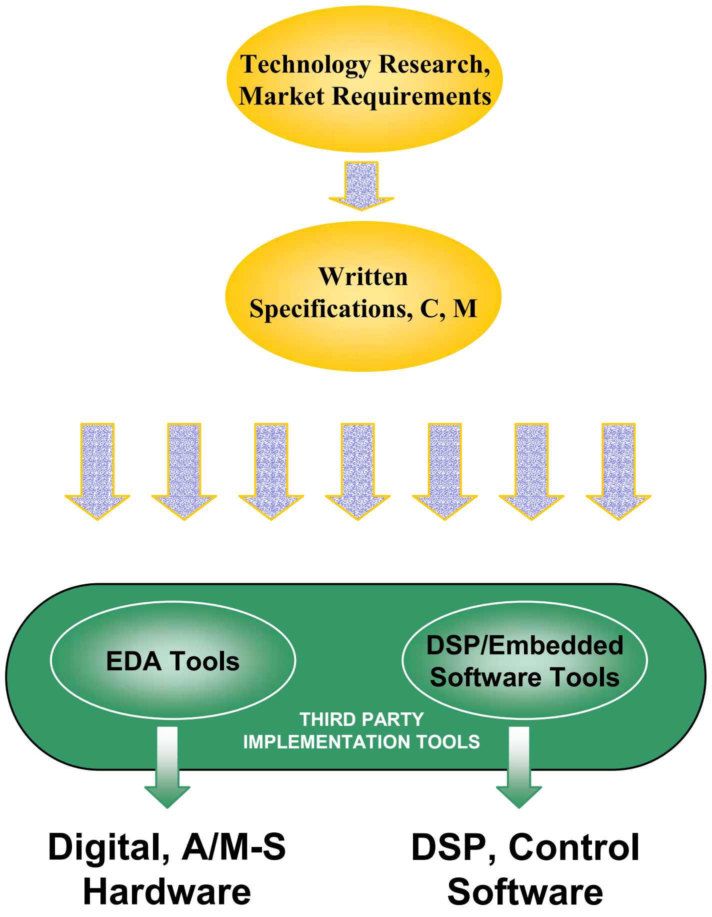




Today's Business and Organization Challenges

- Time-to-market pressure
  - A few months delay has huge revenue impact
- Team integration
  - Analog/Mixed-Signal, Digital hardware, DSP S/W, Control S/W teams
  - All speak a different language and communicate via written documents
- Increasing ASIC mask costs

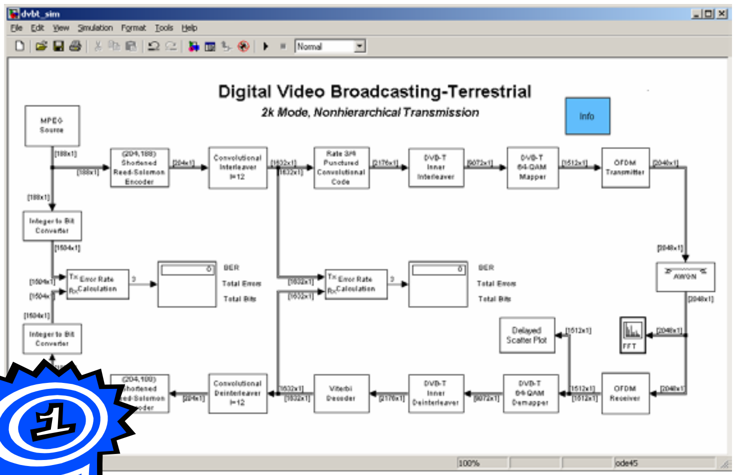
Traditional Flow: Little or No Early Simulation


Technology Research, Market Requirements
Written Specifications, C, M
EDA Tools
DSP/Embedded Software Tools
THIRD PARTY IMPLEMENTATION TOOLS
Digital, A/M-S Hardware
DSP, Control Software

- Technology research and market requirements
- Systems engineering
- Partition into components create written specifications for teams
- Minimal or no simulation. C, M
- Design failure risk high. Flaws detected late, during circuit level, RTL or C/ASM code design
- Risk of time-to-market delays

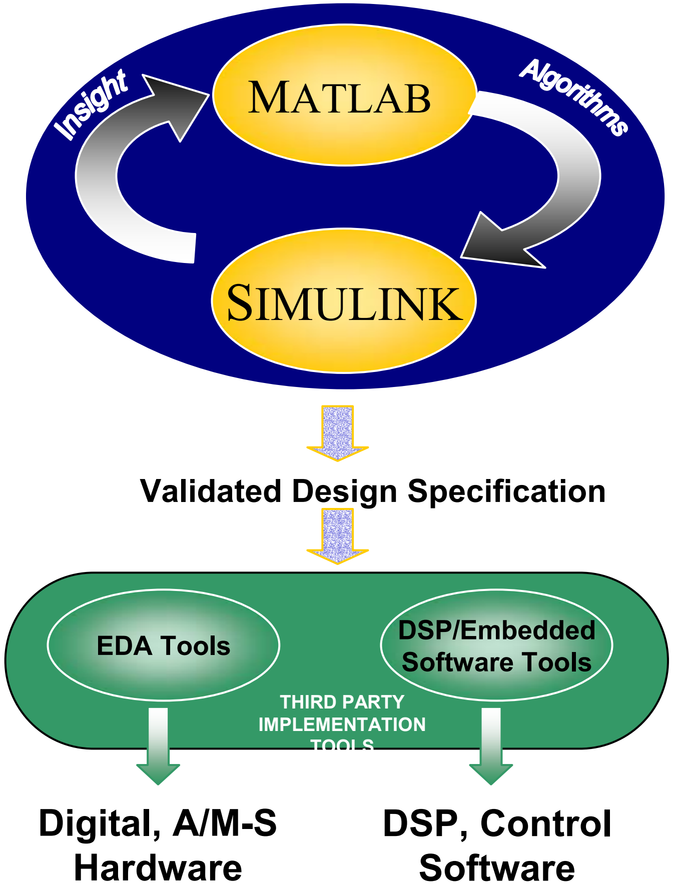
What if You Could…?

- Build a model of a complete system in minutes or days
- Simulate the behavior of the whole system before starting low-level design work?
  - Analog, digital and control together
  - Test for design flaws early
  - Trade-off architectures and parameter in minutes find best design
- Communicate your ideas and share with other teams?
- Keep your IP in an implementation neutral tool as long as possible?
- Only start development with a validated, fully tested design?

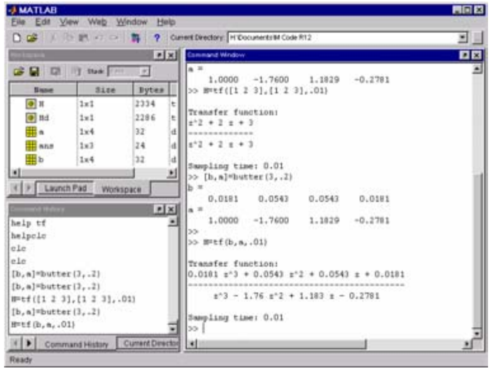


dvbt_sim
File Edit View Simulation Format Tools Help
Normal
Digital Video Broadcasting-Terrestrial
2k Mode, Nonhierarchical Transmission
Info
MPEG Source
(204,188) Shortened Reed-Solomon Encoder
Convolutional Interleaver I=12
Rate 3/4 Punctured Convolutional Code
DVB-T Inner Interleaver
DVB-T 64-QAM Mapper
OFDM Transmitter
Integer to Bit Converter
Error Rate Calculation
BER
Total Errors
Total Bits
AWGN
Delayed Scatter Plot
FFT
Viterbi Decoder
Convolutional Deinterleaver I=12
DVB-T Inner Deinterleaver
DVB-T 64-QAM Demapper
OFDM Receiver
100%
ode45

The MathWorks System-Level Solution


Insight
MATLAB
Algorithms
SIMULINK
Validated Design Specification
EDA Tools
DSP/Embedded Software Tools
THIRD PARTY IMPLEMENTATION TOOLS
Digital, A/M-S Hardware
DSP, Control Software

- MATLAB and Simulink
- Before circuit level, RTL or C/ASM code design
- Create a validated reference design



Name	Size	Bytes
H	1x1	2334
Hd	1x1	2286
a	1x4	32
ans	1x3	24
b	1x4	32

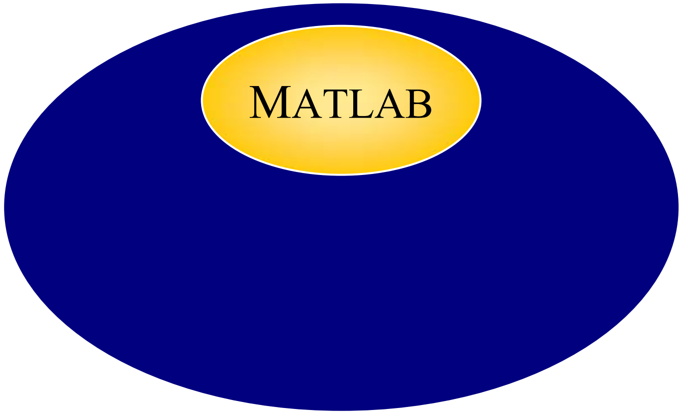
```
help tf
helpclc
clc
clc
[b,a]=butter(3,.2)
[b,a]=butter(3,.2)
H=tf([1 2 3],[1 2 3],.01)
[b,a]=butter(3,.2)
H=tf(b,a,.01)
```

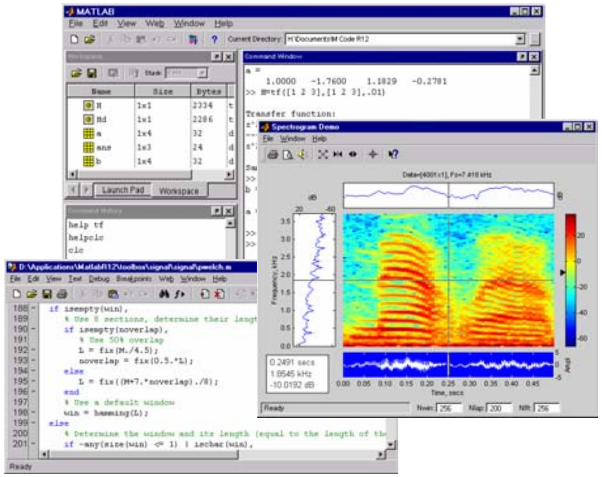
```
a =
    1.0000   -1.7600    1.1829   -0.2781
>> H=tf([1 2 3],[1 2 3],.01)

Transfer function:
z^2 + 2 z + 3
-------------
z^2 + 2 z + 3

Sampling time: 0.01
>> [b,a]=butter(3,.2)
b =
    0.0181    0.0543    0.0543    0.0181
a =
    1.0000   -1.7600    1.1829   -0.2781
>>
>> H=tf(b,a,.01)

Transfer function:
0.0181 z^3 + 0.0543 z^2 + 0.0543 z + 0.0181
-------------------------------------------
     z^3 - 1.76 z^2 + 1.183 z - 0.2781

Sampling time: 0.01
>>
```

SOPC WORLD 2002

The MathWorks

MATLAB




MATLAB
File Edit View Web Window Help
Current Directory: H:\Documents\M Code R12
Command Window
Spectrogram Demo
D:\Applications\MatlabR12\toolbox\signal\signal\pwelch.m
Launch Pad
Workspace
Ready

Name	Size	Bytes
H	1x1	2334
Hd	1x1	2286
a	1x4	32
ans	1x3	24
b	1x4	32

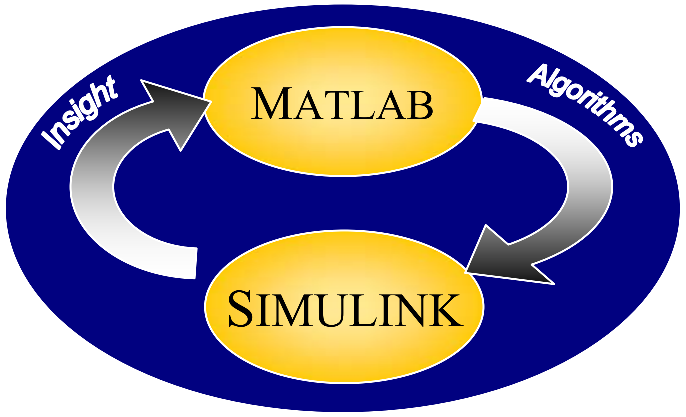
```
a =
    1.0000   -1.7600    1.1829   -0.2781
>> H=tf([1 2 3],[1 2 3],.01)

Transfer function:
```

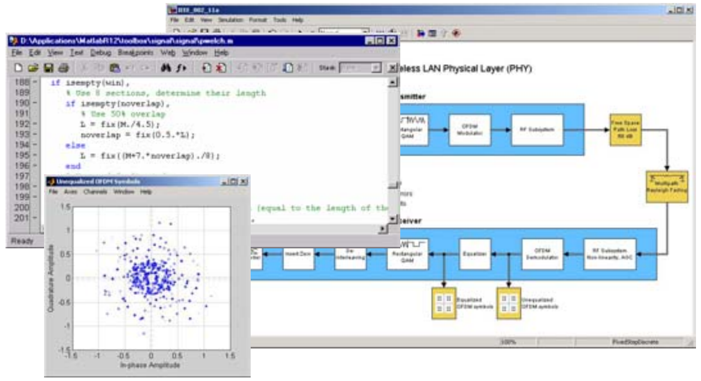
```
help tf
helpclc
clc
```

```
if isempty(win),
    % Use 8 sections, determine their lengt
    if isempty(noverlap),
        % Use 50% overlap
        L = fix(M./4.5);
        noverlap = fix(0.5.*L);
    else
        L = fix((M+7.*noverlap)./8);
    end
    % Use a default window
    win = hamming(L);
else
    % Determine the window and its length (equal to the length of the
    if ~any(size(win) <= 1) | ischar(win),
```

- Research new technology
- Perform mathematical modeling
- Development algorithms
- Acquire, visualize and analyze data

Simulink

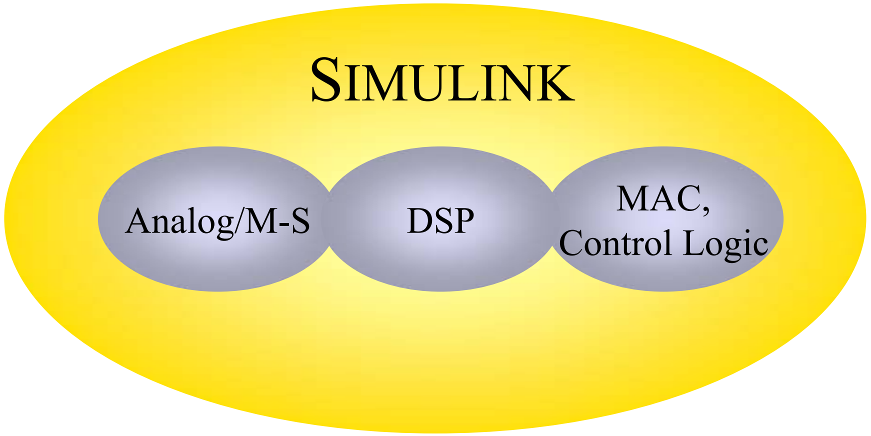


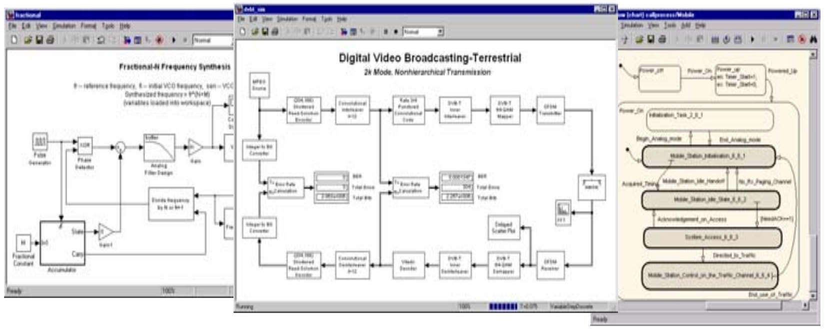


```
if isempty(win),
   % Use 8 sections, determine their length
   if isempty(noverlap),
      % Use 50% overlap
      L = fix(M./4.5);
      noverlap = fix(0.5.*L);
   else
      L = fix((M+7.*noverlap)./8);
   end
```

- Graphically design architecture and simulate behavior of whole system. Bit-true cycle accurate.
- From libraries of pre-built blocks
- Import C or MATLAB Code
- Test, optimize, explore parameter and architecture trade-offs

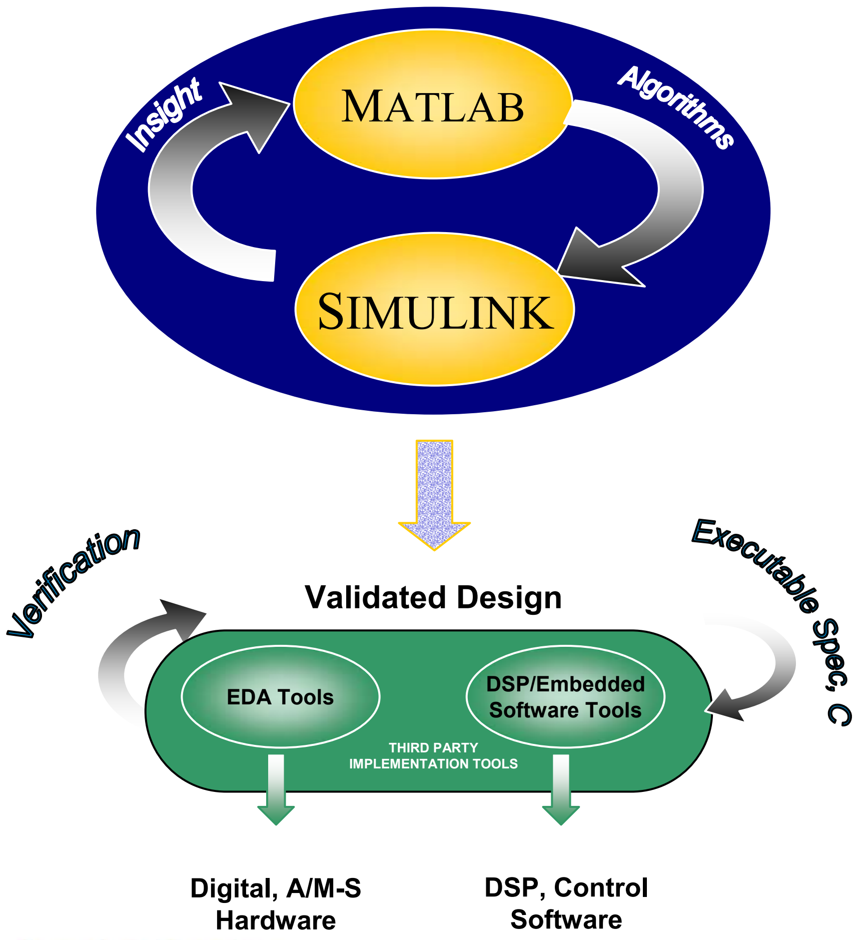
Model Different Components



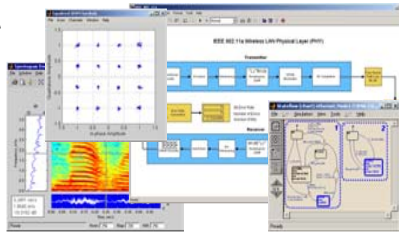

Fractional-N Frequency Synthesis
Digital Video Broadcasting-Terrestrial
2k Mode. Nonhierarchical Transmission

- Analog/Mixed-Signal
  - PLLs, data converters
  - Continuous time, variable-step ODE solvers
- DSP Baseband
  - Discrete time, fast frame-based processing. Bit-true cycle accurate.
- MAC layer/Data Link Layer
  - Simple protocols, acknowledgement schemes
  - Reactive or event driven state machines
  - With Stateflow

Use a Validated Design



- Create validated design
- Use as reference or executable specification to test low-level designs against
- Provide clear specifications
- Detect design flaws early
- Reduce design risk and time-to-market



Motorola's Wireless Subscriber Systems Group

- Challenge
  - Mixed-signal Phase-Locked Loop (PLL) design
  - Cycle-to-cycle jitter and loop locking sensitivity
  - SPICE/Verilog
  - 100 μsecs: 2 hours
- Solution
  - Simulink
  - Faster development time and simulation time
  - 100 μsecs : 2.5 mins
  - Sub-picosecond resolution

“The Simulink models exceeded our project specifications for required simulation speed. Accurate simulations can now be measured in minutes rather than hours or days.”

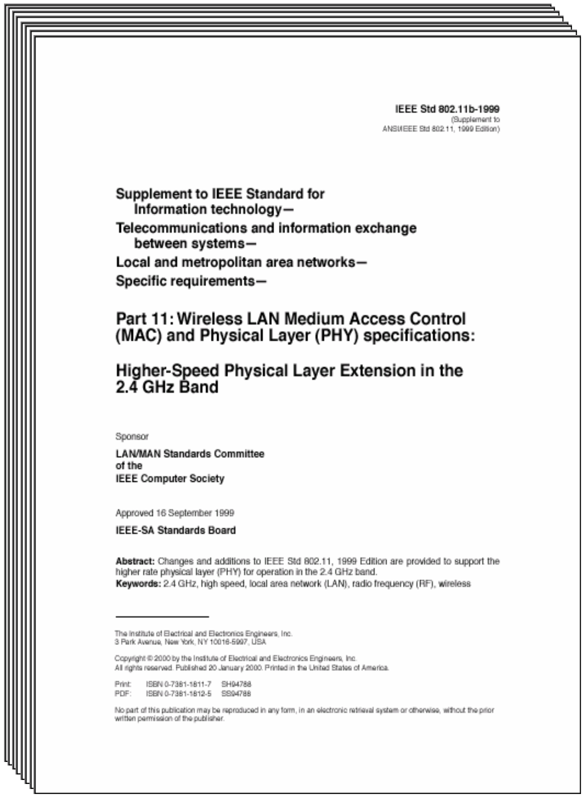
Yuan Yuan, Motorola

SOPC WORLD 2002

The MathWorks

PHY Design Case Study: IEEE 802.11b

- IEEE Standard Document
  - 600 pages (11 + b appendix)
  - 4 modes (1, 2, 5.5 and 11Mbps)
- Components
  - Framing and CRC
  - Long/short preamble and sync
  - Modulation and spreading
  - Filtering
  - Channel number selection (1-11)
  - RF subsystems, Tx power, mask, power-on ramp



IEEE Std 802.11b-1999
(Supplement to
ANSI/IEEE Std 802.11, 1999 Edition)

Supplement to IEEE Standard for Information technology—
Telecommunications and information exchange between systems—
Local and metropolitan area networks—
Specific requirements—

Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications:

Higher-Speed Physical Layer Extension in the 2.4 GHz Band

Sponsor

LAN/MAN Standards Committee
of the
IEEE Computer Society

Approved 16 September 1999

IEEE-SA Standards Board

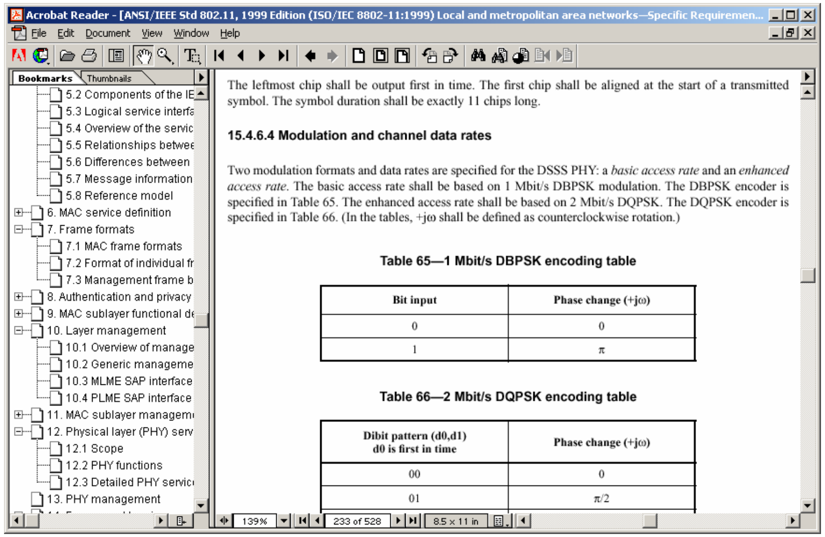
Abstract: Changes and additions to IEEE Std 802.11, 1999 Edition are provided to support the higher rate physical layer (PHY) for operation in the 2.4 GHz band.
Keywords: 2.4 GHz, high speed, local area network (LAN), radio frequency (RF), wireless

The Institute of Electrical and Electronics Engineers, Inc.
3 Park Avenue, New York, NY 10016-5997, USA

Print: ISBN 0-7381-1811-7 SH94788
PDF: ISBN 0-7381-1812-5 SS94788

1Mbps Mode

- Modulation
  - DBPSK
- Spread
  - 11 chips Barker sequence per symbol
- Pulse shaping
  - 4/8 samples per chip
  - Root raised cosine
- Channel
  - AWGN



The leftmost chip shall be output first in time. The first chip shall be aligned at the start of a transmitted symbol. The symbol duration shall be exactly 11 chips long.

15.4.6.4 Modulation and channel data rates

Two modulation formats and data rates are specified for the DSSS PHY: a *basic access rate* and an *enhanced access rate*. The basic access rate shall be based on 1 Mbit/s DBPSK modulation. The DBPSK encoder is specified in Table 65. The enhanced access rate shall be based on 2 Mbit/s DQPSK. The DQPSK encoder is specified in Table 66. (In the tables, +jω shall be defined as counterclockwise rotation.)

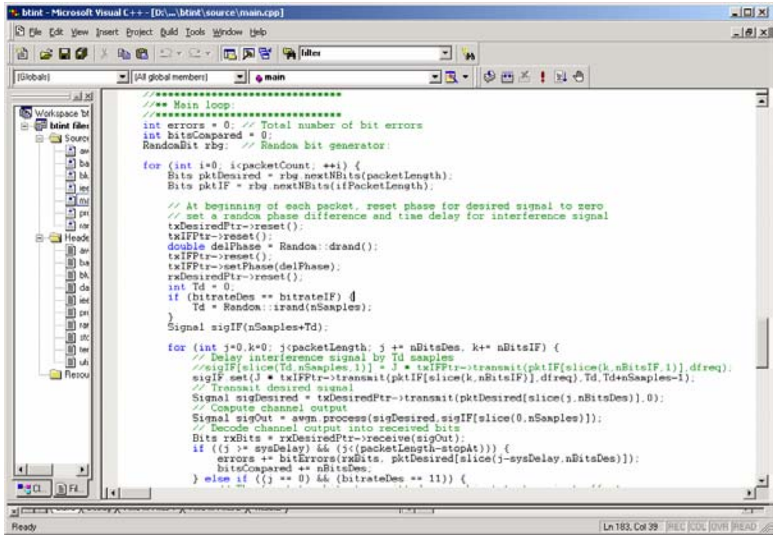
Table 65—1 Mbit/s DBPSK encoding table

Bit input	Phase change (+jω)
0	0
1	π

Table 66—2 Mbit/s DQPSK encoding table

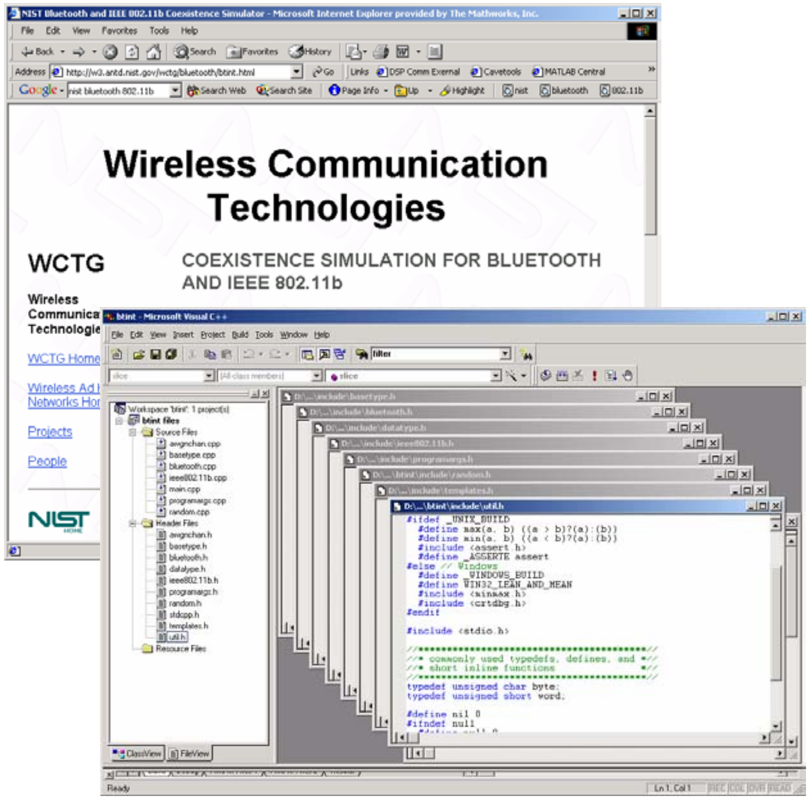
Dibit pattern (d0,d1) d0 is first in time	Phase change (+jω)
00	0
01	π/2

C/C++


btint - Microsoft Visual C++ - [D:\...\btint\source\main.cpp]
File Edit View Insert Project Build Tools Window Help
filter
[Globals]
[All global members]
main
Workspace 'bt
btint files
Source
Heade
Resou
Ready
Ln 183, Col 39

```
//*******************************
//** Main loop:
//*******************************
int errors = 0; // Total number of bit errors
int bitsCompared = 0;
RandomBit rbg;  // Random bit generator:

for (int i=0; i<packetCount; ++i) {
    Bits pktDesired = rbg.nextNBits(packetLength);
    Bits pktIF = rbg.nextNBits(ifPacketLength);

    // At beginning of each packet, reset phase for desired signal to zero
    // set a random phase difference and time delay for interference signal
    txDesiredPtr->reset();
    txIFPtr->reset();
    double delPhase = Random::drand();
    txIFPtr->reset();
    txIFPtr->setPhase(delPhase);
    rxDesiredPtr->reset();
    int Td = 0;
    if (bitrateDes == bitrateIF) {
        Td = Random::irand(nSamples);
    }
    Signal sigIF(nSamples+Td);

    for (int j=0,k=0; j<packetLength; j += nBitsDes, k+= nBitsIF) {
        // Delay interference signal by Td samples
        //sigIF[slice(Td,nSamples,1)] = J * txIFPtr->transmit(pktIF[slice(k,nBitsIF,1)],dfreq);
        sigIF.set(J * txIFPtr->transmit(pktIF[slice(k,nBitsIF)],dfreq),Td,Td+nSamples-1);
        // Transmit desired signal
        Signal sigDesired = txDesiredPtr->transmit(pktDesired[slice(j,nBitsDes)],0);
        // Compute channel output
        Signal sigOut = awgn.process(sigDesired,sigIF[slice(0,nSamples)]);
        // Decode channel output into received bits
        Bits rxBits = rxDesiredPtr->receive(sigOut);
        if ((j >= sysDelay) && (j<(packetLength-stopAt))) {
            errors += bitErrors(rxBits, pktDesired[slice(j-sysDelay,nBitsDes)]);
            bitsCompared += nBitsDes;
        } else if ((j == 0) && (bitrateDes == 11)) {
```

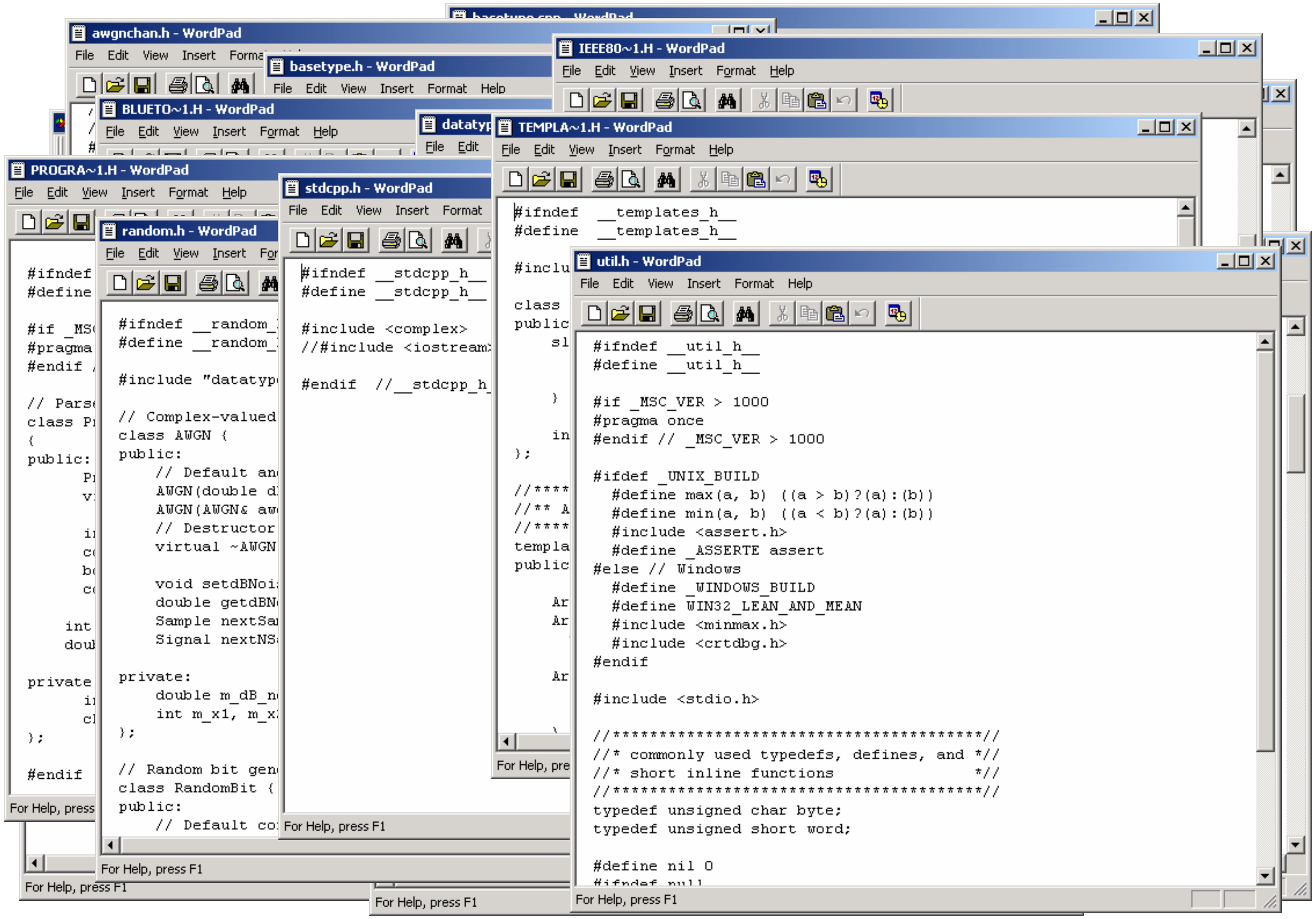
NIST 802.11b and Bluetooth C/C++ Code


NIST Bluetooth and IEEE 802.11b Coexistence Simulator - Microsoft Internet Explorer provided by The Mathworks, Inc.
Wireless Communication Technologies
WCTG
COEXISTENCE SIMULATION FOR BLUETOOTH AND IEEE 802.11b
WCTG Home
Projects
People
btint - Microsoft Visual C++
D:\...\btint\include\util.h

```
#ifdef _UNIX_BUILD
  #define max(a, b) ((a > b)?(a):(b))
  #define min(a, b) ((a < b)?(a):(b))
  #include <assert.h>
  #define _ASSERTE assert
#else // Windows
  #define _WINDOWS_BUILD
  #define WIN32_LEAN_AND_MEAN
  #include <minmax.h>
  #include <crtdbg.h>
#endif

#include <stdio.h>

//*************************************//
//* commonly used typedefs, defines, and *//
//* short inline functions              *//
//*************************************//
typedef unsigned char byte;
typedef unsigned short word;

#define nil 0
#ifndef null
```

- w3.antd.nist.gov/wctg/bluetooth/btint.html
- Bluetooth and 802.11b
- 802.11b
  - Random bits
  - 1 or 11Mbps modulation and spreading
  - Filtering
- Bluetooth
  - Random bits
  - Modulation
  - Filtering

C/C++ Code: 17 Files, 1500-2000 lines



```
#ifndef __templates_h__
#define __templates_h__
```

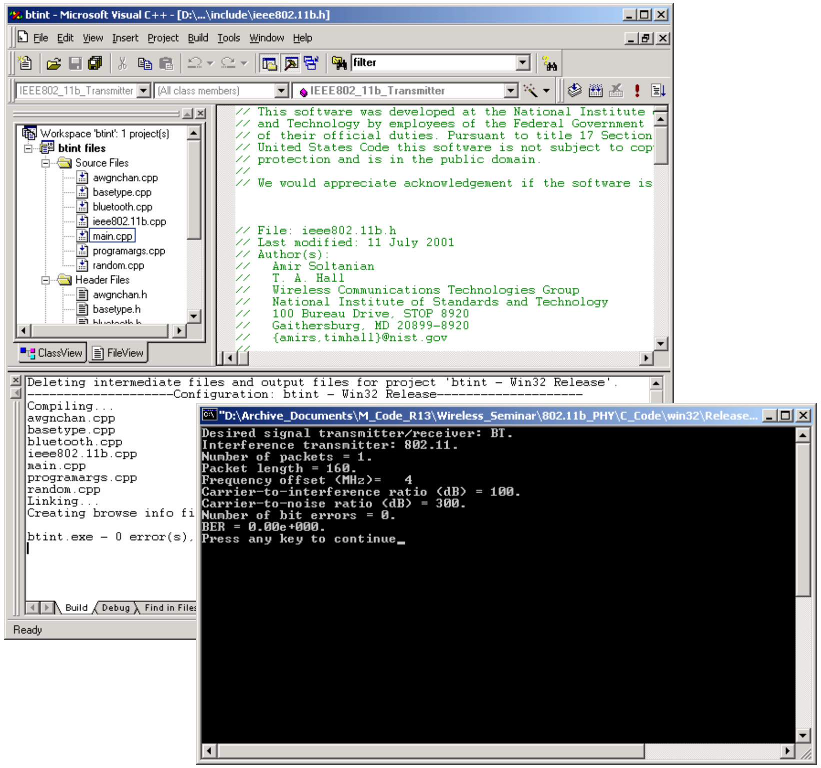
```
#ifndef __stdcpp_h__
#define __stdcpp_h__

#include <complex>
//#include <iostream

#endif //__stdcpp_h
```

```
#ifndef __random_
#define __random_

#include "datatyp

// Complex-valued
class AWGN {
public:
    // Default an
    AWGN(double d
    AWGN(AWGN& aw
    // Destructor
    virtual ~AWGN

    void setdBNoi
    double getdBN
    Sample nextSa
    Signal nextNS

private:
    double m_dB_n
    int m_x1, m_x
};

// Random bit gen
class RandomBit {
public:
    // Default co
```

```
#ifndef __util_h__
#define __util_h__

#if _MSC_VER > 1000
#pragma once
#endif // _MSC_VER > 1000

#ifdef _UNIX_BUILD
  #define max(a, b)  ((a > b)?(a):(b))
  #define min(a, b)  ((a < b)?(a):(b))
  #include <assert.h>
  #define _ASSERTE assert
#else // Windows
  #define _WINDOWS_BUILD
  #define WIN32_LEAN_AND_MEAN
  #include <minmax.h>
  #include <crtdbg.h>
#endif

#include <stdio.h>

//*******************************************//
//* commonly used typedefs, defines, and   *//
//* short inline functions                 *//
//*******************************************//
typedef unsigned char byte;
typedef unsigned short word;

#define nil 0
#ifndef null
```

C/C++ Code: Build, Run, Debug, and Change

- Build and run
  - Bug: File I/O
- Speed
  - 7 secs for 100 packets
- Time to create
  - 400 days @ 5 lines/day
  - 40 days @ 50 lines/day
- Removing Bluetooth
  - Difficult. Many implicit dependencies



```
// This software was developed at the National Institute
// and Technology by employees of the Federal Government
// of their official duties. Pursuant to title 17 Section
// United States Code this software is not subject to cop
// protection and is in the public domain.
//
// We would appreciate acknowledgement if the software is



// File: ieee802.11b.h
// Last modified: 11 July 2001
// Author(s):
//   Amir Soltanian
//   T. A. Hall
//   Wireless Communications Technologies Group
//   National Institute of Standards and Technology
//   100 Bureau Drive, STOP 8920
//   Gaithersburg, MD 20899-8920
//   {amirs,timhall}@nist.gov
```

```
Deleting intermediate files and output files for project 'btint - Win32 Release'.
--------------------Configuration: btint - Win32 Release--------------------
Compiling...
awgnchan.cpp
basetype.cpp
bluetooth.cpp
ieee802.11b.cpp
main.cpp
programargs.cpp
random.cpp
Linking...
Creating browse info fi

btint.exe - 0 error(s),
```

```
Desired signal transmitter/receiver: BT.
Interference transmitter: 802.11.
Number of packets = 1.
Packet length = 160.
Frequency offset (MHz)=   4
Carrier-to-interference ratio (dB) = 100.
Carrier-to-noise ratio (dB) = 300.
Number of bit errors = 0.
BER = 0.00e+000.
Press any key to continue_
```

```
D:\>btint -c 100 -d 802.11 -i BT -EbNo 6
```

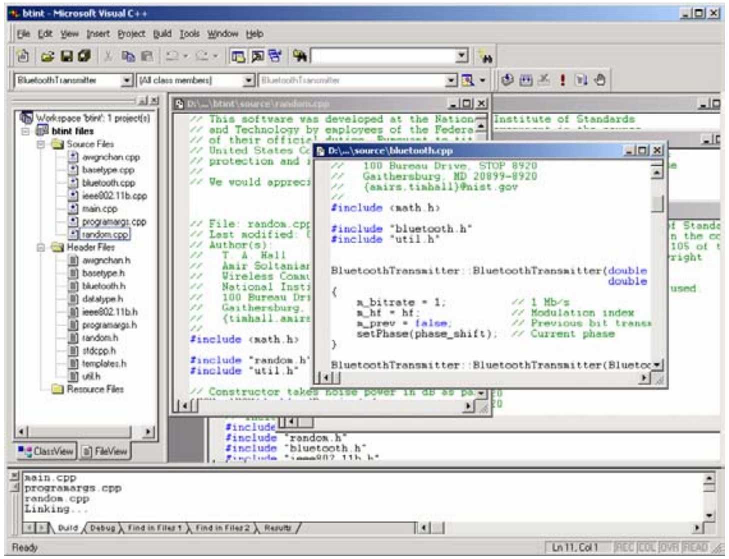
S/W Demonstration

SOPC WORLD 2002

The MathWorks

C/C++ Pros and Cons

- Pros
  - Ubiquitous
  - Fast to execute
  - Data type options
- Cons
  - No canned DSP/Comm functions
  - Can't visualize signals
  - Too low-level, lots housekeeping
  - Error prone design entry, implicit interdependencies
  - Slow to iterate, debug and make changes



```
// This software was developed at the Nation
// and Technology by employees of the Federa
// of their official
// United States Co
// protection and
//
// We would appreci

// File: random.cpp
// Last modified:
// Author(s):
//    T. A. Hall
//    Amir Soltanian
//    Wireless Comm
//    National Insti
//    100 Bureau Dri
//    Gaithersburg,
//    {timhall.amirs
//
#include <math.h>

#include "random.h"
#include "util.h"

// Constructor takes noise power in dB as pa
```

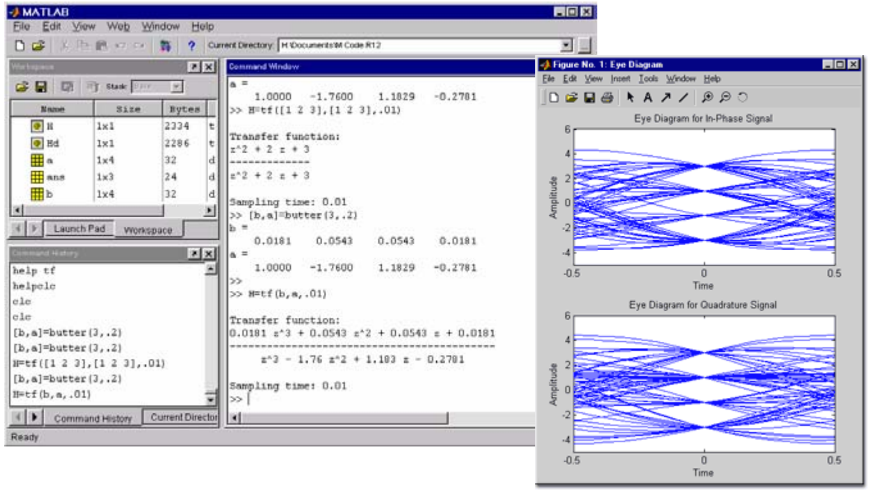
```
//    100 Bureau Drive, STOP 8920
//    Gaithersburg, MD 20899-8920
//    {amirs.timhall}@nist.gov
//
#include <math.h>

#include "bluetooth.h"
#include "util.h"


BluetoothTransmitter::BluetoothTransmitter(double
                                           double
{
    m_bitrate = 1;             // 1 Mb/s
    m_hf = hf;                 // Modulation index
    m_prev = false;            // Previous bit transm
    setPhase(phase_shift);     // Current phase
}

BluetoothTransmitter::BluetoothTransmitter(Bluetoc
```

```
Institute of Standards

f Standa
n the co
105 of t
right

used
```

```
#include
#include "random.h"
#include "bluetooth.h"
#include "ieee802.11b.h"
```

```
main.cpp
programargs.cpp
random.cpp
Linking...
```

MATLAB


MATLAB
File Edit View Web Window Help
Current Directory: H:\Documents\M Code R12
Workspace
Stack:
Launch Pad
Workspace
Command History
Current Director
Ready
Command Window
Figure No. 1: Eye Diagram
File Edit View Insert Tools Window Help
Eye Diagram for In-Phase Signal
Eye Diagram for Quadrature Signal
Amplitude
Time

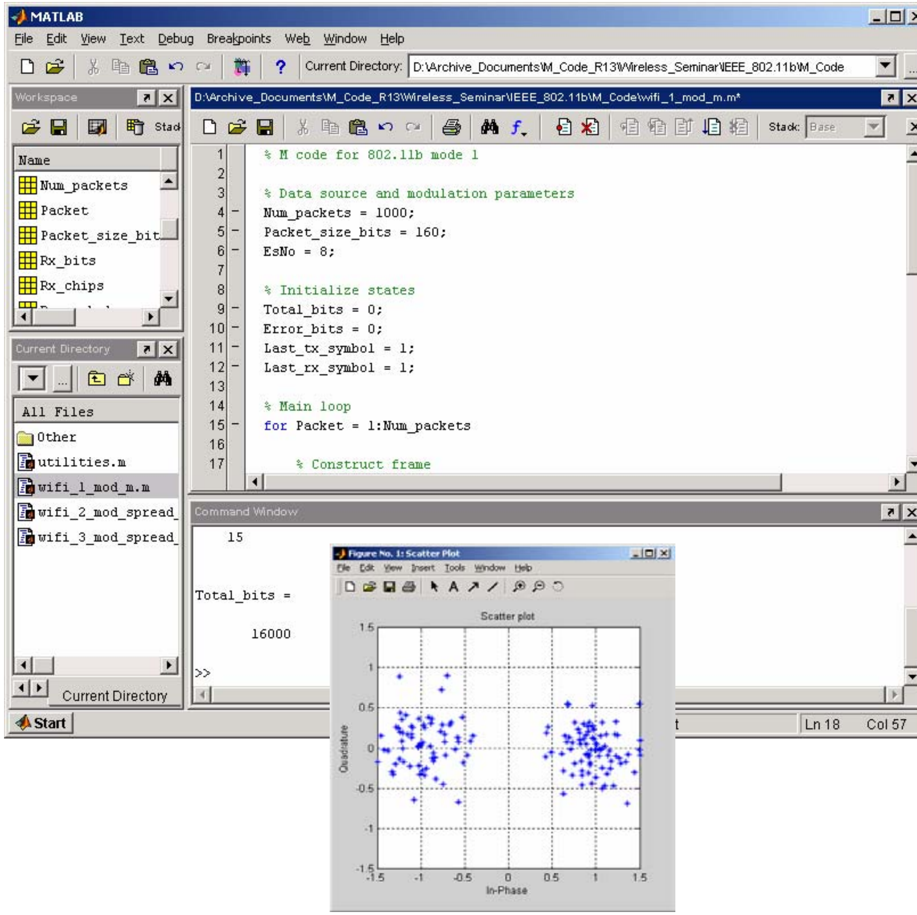
Name	Size	Bytes	
H	1x1	2334	t
Hd	1x1	2286	t
a	1x4	32	d
ans	1x3	24	d
b	1x4	32	d

```
help tf
helpclc
clc
clc
[b,a]=butter(3,.2)
[b,a]=butter(3,.2)
H=tf([1 2 3],[1 2 3],.01)
[b,a]=butter(3,.2)
H=tf(b,a,.01)
```

```
a =
    1.0000   -1.7600    1.1829   -0.2781
>> H=tf([1 2 3],[1 2 3],.01)

Transfer function:
z^2 + 2 z + 3
-------------
z^2 + 2 z + 3

Sampling time: 0.01
>> [b,a]=butter(3,.2)
b =
    0.0181    0.0543    0.0543    0.0181
a =
    1.0000   -1.7600    1.1829   -0.2781
>>
>> H=tf(b,a,.01)

Transfer function:
0.0181 z^3 + 0.0543 z^2 + 0.0543 z + 0.0181
-------------------------------------------
     z^3 - 1.76 z^2 + 1.183 z - 0.2781

Sampling time: 0.01
>> 
```

802.11b: M Code

- Using MATLAB
  - Generating bits
  - Symbols and noise
  - `scatterplot(Rx_symbols)`
- 802.11 Tx lines of code
  - One single file 52 lines
  - Modulation (3 lines)
  - Spreading (1 line)
- Speed
  - 2 secs for 100 packets
- Find the bug


MATLAB
File Edit View Text Debug Breakpoints Web Window Help
Current Directory: D:\Archive_Documents\M_Code_R13\Wireless_Seminar\IEEE_802.11b\M_Code
Workspace
Name
Num_packets
Packet
Packet_size_bit
Rx_bits
Rx_chips
Current Directory
All Files
Other
utilities.m
wifi_1_mod_m.m
wifi_2_mod_spread_
wifi_3_mod_spread_
D:\Archive_Documents\M_Code_R13\Wireless_Seminar\IEEE_802.11b\M_Code\wifi_1_mod_m.m*
Stack: Base
Command Window
Start
Ln 18 Col 57
Figure No. 1: Scatter Plot
Scatter plot
Quadrature
In-Phase

```
% M code for 802.11b mode 1

% Data source and modulation parameters
Num_packets = 1000;
Packet_size_bits = 160;
EsNo = 8;

% Initialize states
Total_bits = 0;
Error_bits = 0;
Last_tx_symbol = 1;
Last_rx_symbol = 1;

% Main loop
for Packet = 1:Num_packets

    % Construct frame
```

```
  15


Total_bits =

     16000

>>
```

S/W Demonstration: Build and show

SOPC WORLD 2002

The MathWorks

MATLAB Vs C/C++: Spreading and Upsampling

- **C Code**

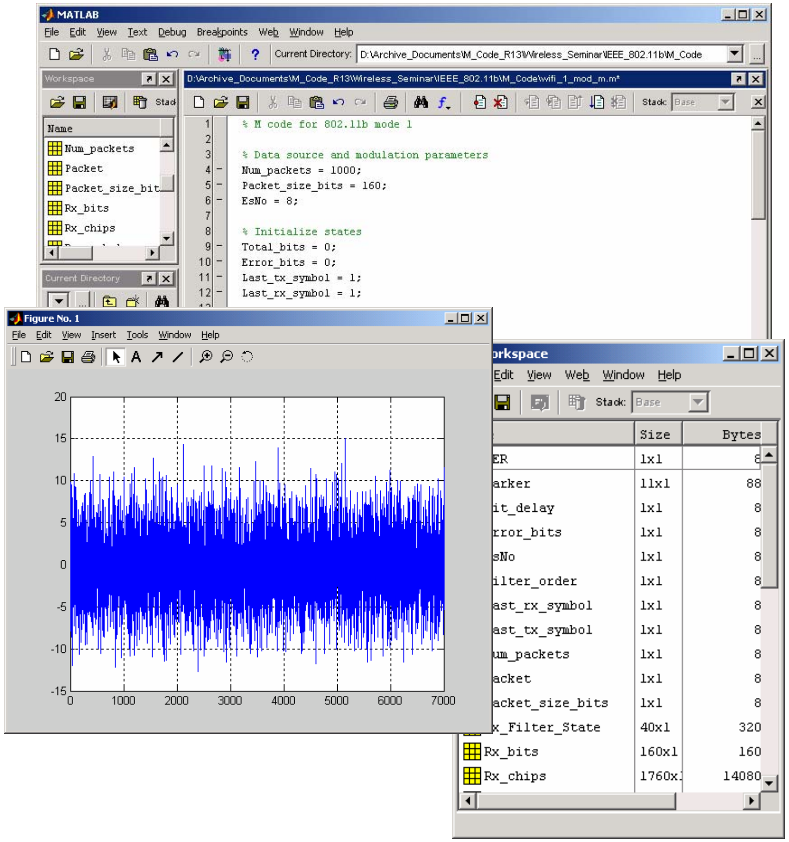
```
Bits spread=addChips(diffOut[slice(i,1)]);

Bits
IEEE802_11b_Transmitter::addChips(const Bits& input) {
    Bits spreadOut(input.size()*Ns,false);
    for (int i=0;i<input.size();++i){
        for(int j=0; j<11; ++j) {
            spreadOut[i*Ns+4*j]= m_chip[j]^input[i];
        }
    }
   return spreadOut;
}
```

- **M Code**

```
   Tx_chips=reshape(Barker*Tx_symbols',[],1);
   Tx_samples(1:Samples_per_chip:end)=Tx_chips;
```

MATLAB Pros and Cons

- Pros
  - Interactive
  - Easy signal visualization
  - Canned common functions
  - Faster development
- Cons
  - Limited data types
  - Limited low-level control
  - Less memory efficient
  - Slower to execute for scalars, loops before R13


MATLAB
File Edit View Text Debug Breakpoints Web Window Help
Current Directory: D:\Archive_Documents\M_Code_R13\Wireless_Seminar\IEEE_802.11b\M_Code
Workspace
Name
Num_packets
Packet
Packet_size_bit
Rx_bits
Rx_chips
Current Directory
D:\Archive_Documents\M_Code_R13\Wireless_Seminar\IEEE_802.11b\M_Code\wifi_1_mod_m.m*
Stack: Base
Figure No. 1
File Edit View Insert Tools Window Help
20
15
10
5
0
-5
-10
-15
0
1000
2000
3000
4000
5000
6000
7000
orkspace
Edit View Web Window Help
Stack: Base

```
% M code for 802.11b mode 1

% Data source and modulation parameters
Num_packets = 1000;
Packet_size_bits = 160;
EsNo = 8;

% Initialize states
Total_bits = 0;
Error_bits = 0;
Last_tx_symbol = 1;
Last_rx_symbol = 1;
```

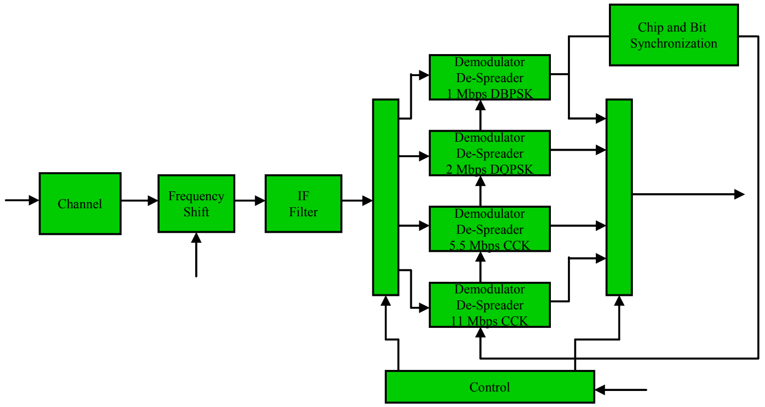
	Size	Bytes
ER	1x1	8
arker	11x1	88
it_delay	1x1	8
rror_bits	1x1	8
sNo	1x1	8
ilter_order	1x1	8
ast_rx_symbol	1x1	8
ast_tx_symbol	1x1	8
um_packets	1x1	8
acket	1x1	8
acket_size_bits	1x1	8
x_Filter_State	40x1	320
Rx_bits	160x1	160
Rx_chips	1760x1	14080

SOPC WORLD 2002

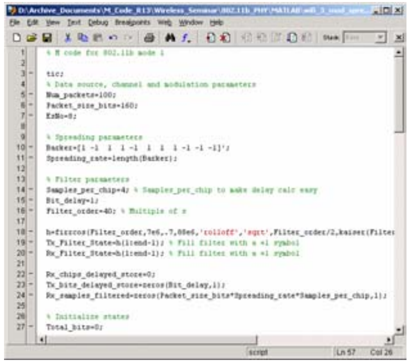
The MathWorks

Limitations of C and M for System Design

- No architecture information
  - Can only model a pipeline
  - Can't describe a real system
- No timing information
  - Can only model uniform Fs
  - Difficult to model delays
  - Must manually handle state
  - Can't model A/M-S
  - Difficult to model Rx algorithms


Channel
Frequency Shift
IF Filter
Demodulator De-Spreader 1 Mbps DBPSK
Demodulator De-Spreader 2 Mbps DQPSK
Demodulator De-Spreader 5.5 Mbps CCK
Demodulator De-Spreader 11 Mbps CCK
Chip and Bit Synchronization
Control

=

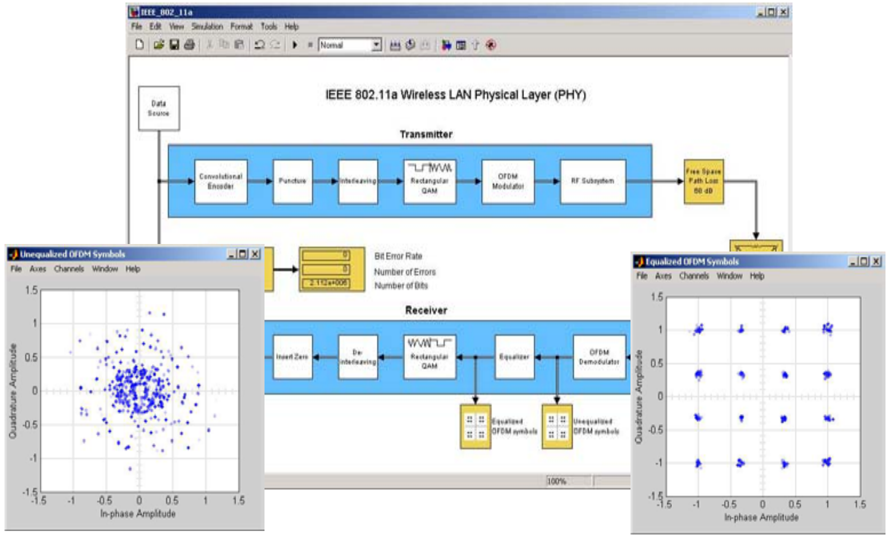

D:\Archive_Documents\M_Code_R13\Wireless_Seminar\802.11b_PHY\MATLAB\
File Edit View Text Debug Breakpoints Web Window Help
Stack:
script
Ln 57 Col 26

```
% M code for 802.11b mode 1

tic;
% Data source, channel and modulation parameters
Num_packets=100;
Packet_size_bits=160;
EsNo=8;

% Spreading parameters
Barker=[1 -1  1  1 -1  1  1  1 -1 -1 -1]';
Spreading_rate=length(Barker);

% Filter parameters
Samples_per_chip=4; % Samples_per_chip to make delay calc easy
Bit_delay=1;
Filter_order=40; % Multiple of s

h=firrcos(Filter_order,7e6,.7,88e6,'rolloff','sqrt',Filter_order/2,kaiser(Filte
Tx_Filter_State=h(1:end-1); % Fill filter with a +1 symbol
Rx_Filter_State=h(1:end-1); % Fill filter with a +1 symbol

Rx_chips_delayed_store=0;
Tx_bits_delayed_store=zeros(Bit_delay,1);
Rx_samples_filtered=zeros(Packet_size_bits*Spreading_rate*Samples_per_chip,1);

% Initialize states
Total_bits=0;
```

SOPC WORLD 2002

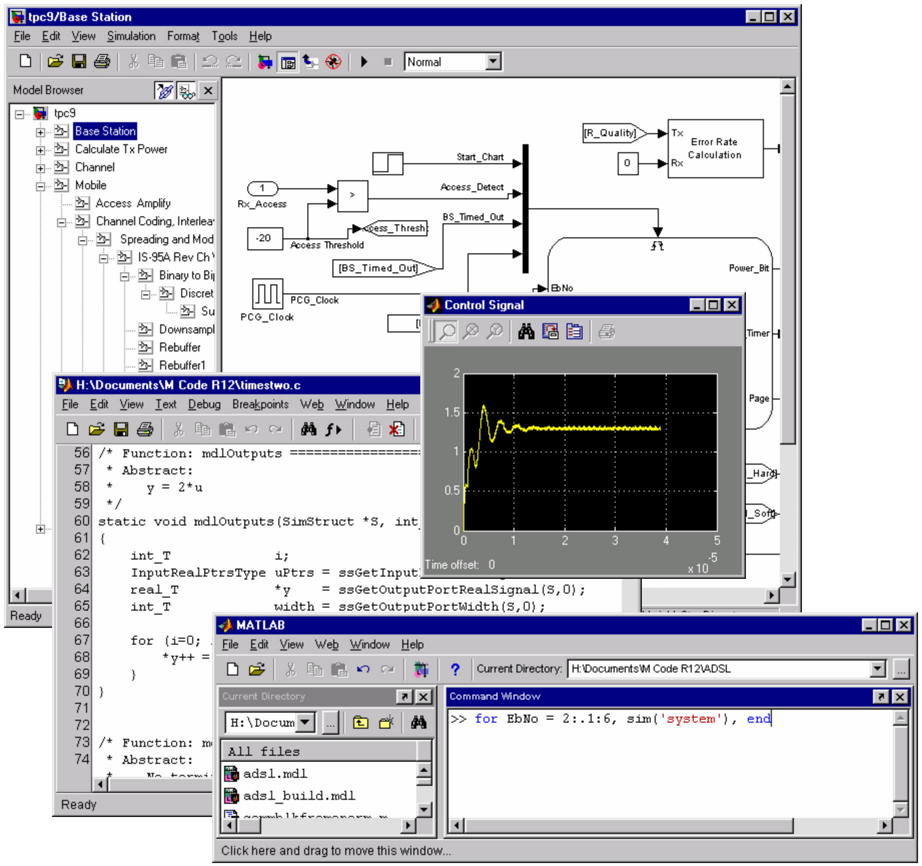
The MathWorks

Simulink



ALTERA

Simulink

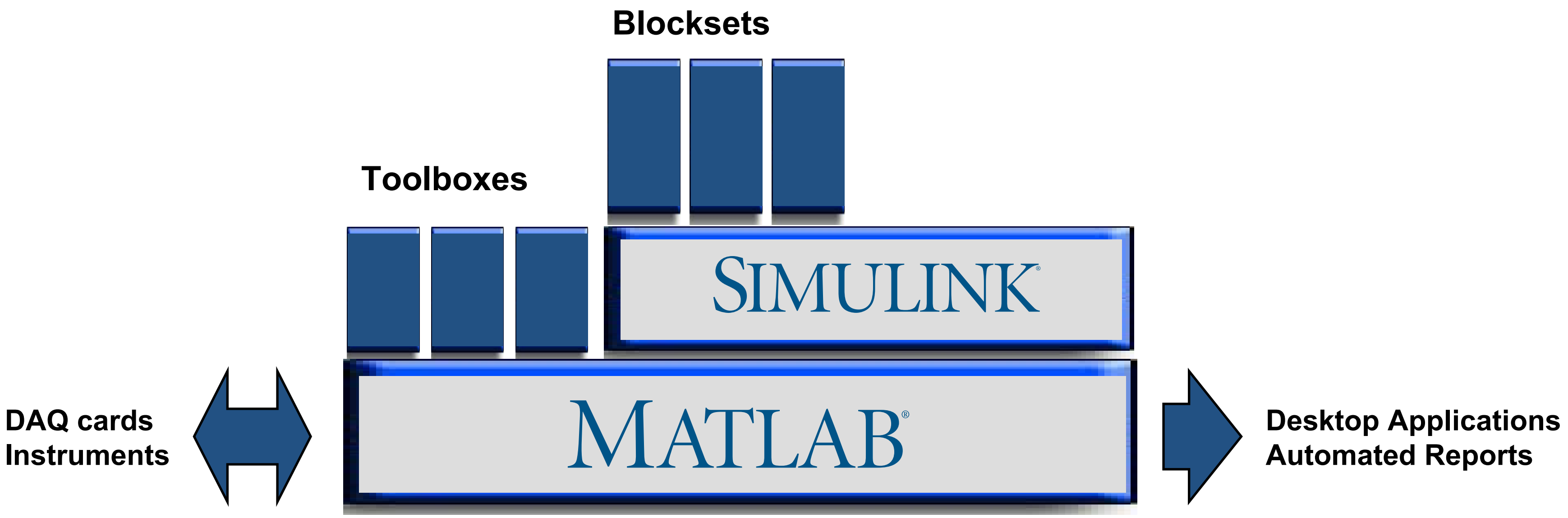


```
/* Function: mdlOutputs =================
 * Abstract:
 *     y = 2*u
 */
static void mdlOutputs(SimStruct *S, int_
{
    int_T             i;
    InputRealPtrsType uPtrs = ssGetInput
    real_T            *y    = ssGetOutputPortRealSignal(S,0);
    int_T             width = ssGetOutputPortWidth(S,0);

    for (i=0;
        *y++ =
    }
}


/* Function: m
 * Abstract:
```

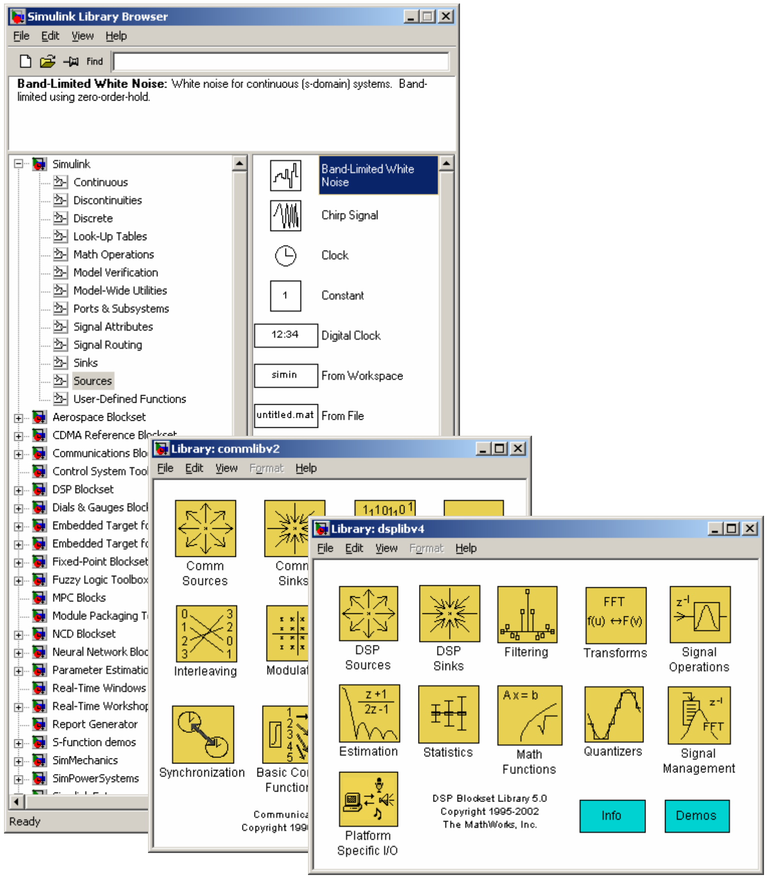
```
>> for EbNo = 2:.1:6, sim('system'), end
```

- Hierarchical block diagram design and simulation tool
- Digital, analog/mixed signal and event driven
- Visualize Signals
- Co-develop with C code
- Integrated with MATLAB

Simulink in The MATLAB Environment



The Simulink Block Libraries



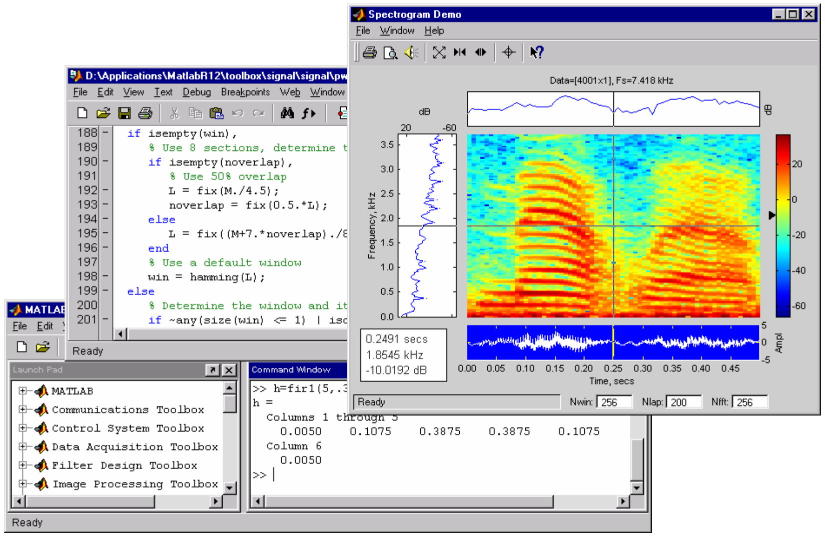
Band-Limited White Noise: White noise for continuous (s-domain) systems. Band-limited using zero-order-hold.

- Simulink
  - Sources and sinks
  - Continuous and Discrete
  - Math, Non-Linear
  - Look-up tables, user functions
  - Subsystems, verification
- DSP Blockset
  - Sub libraries
- Communications Blockset
  - Sub libraries
- Fixed-Point Blockset
- Power-Systems Blockset
- Incremental development

S/W Demonstration: Build

The MathWorks products for DSP and Communications

- Accelerating engineering design and discovery
- MATLAB for algorithm development and analysis
- Simulink for system-level design

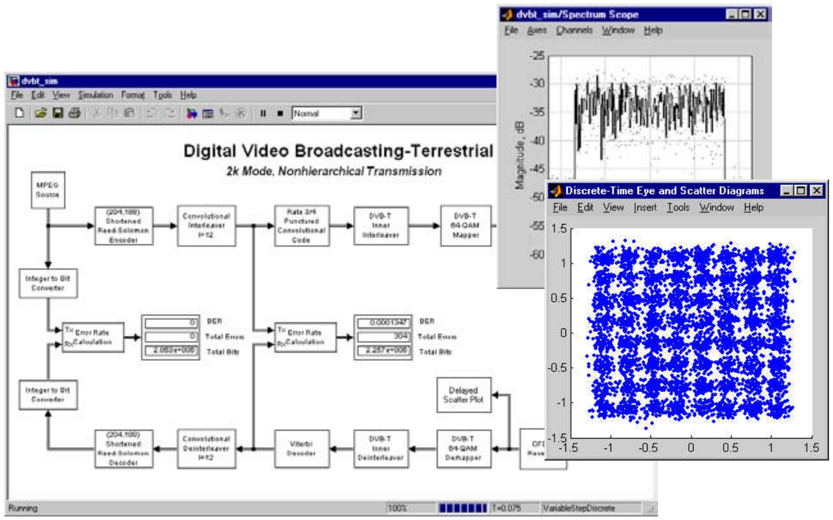


```
if isempty(win),
   % Use 8 sections, determine t
   if isempty(noverlap),
      % Use 50% overlap
      L = fix(M./4.5);
      noverlap = fix(0.5.*L);
   else
      L = fix((M+7.*noverlap)./8
   end
   % Use a default window
   win = hamming(L);
else
   % Determine the window and it
   if ~any(size(win) <= 1) | isc
```

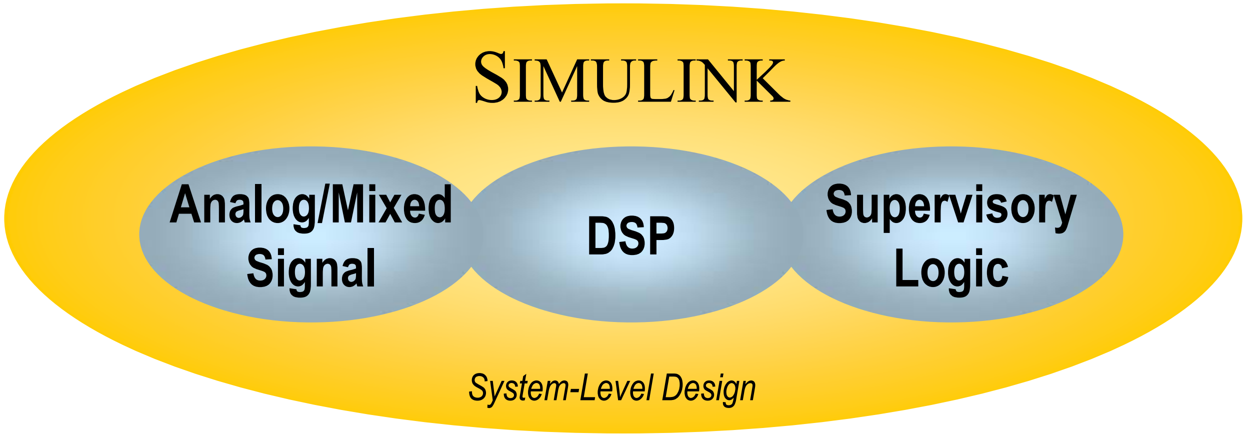
```
MATLAB
Communications Toolbox
Control System Toolbox
Data Acquisition Toolbox
Filter Design Toolbox
Image Processing Toolbox
```

```
>> h=fir1(5,.3
h =
  Columns 1 through 5
    0.0050    0.1075    0.3875    0.3875    0.1075
  Column 6
    0.0050
>> |
```

```
0.2491 secs
1.8545 kHz
-10.0192 dB
```

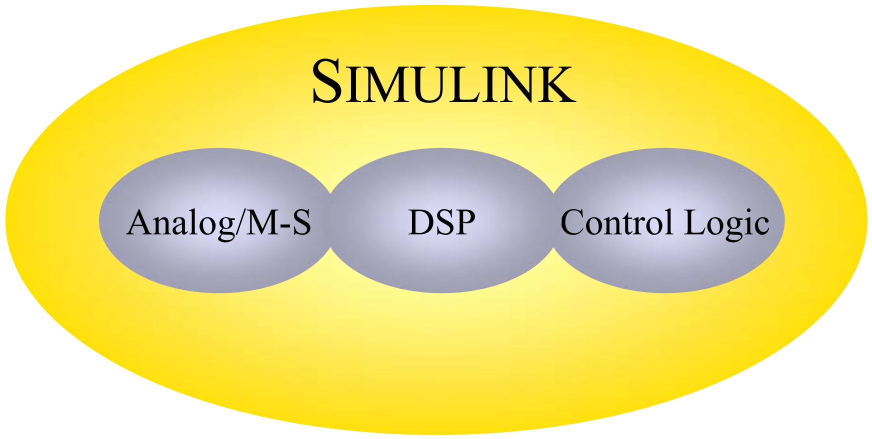

dvbt_sim
File Edit View Simulation Format Tools Help
Digital Video Broadcasting-Terrestrial
2k Mode, Nonhierarchical Transmission
MPEG Source
(204,188) Shortened Reed-Solomon Encoder
Convolutional Interleaver I=12
Rate 3/4 Punctured Convolutional Code
DVB-T Inner Interleaver
DVB-T 64-QAM Mapper
Integer to Bit Converter
Tx Error Rate Rx Calculation
BER
Total Errors
Total Bits
Delayed Scatter Plot
(204,188) Shortened Reed-Solomon Decoder
Convolutional Deinterleaver I=12
Viterbi Decoder
DVB-T Inner Deinterleaver
DVB-T 64-QAM Demapper
Running
dvbt_sim/Spectrum Scope
File Axes Channels Window Help
Magnitude, dB
Discrete-Time Eye and Scatter Diagrams
File Edit View Insert Tools Window Help

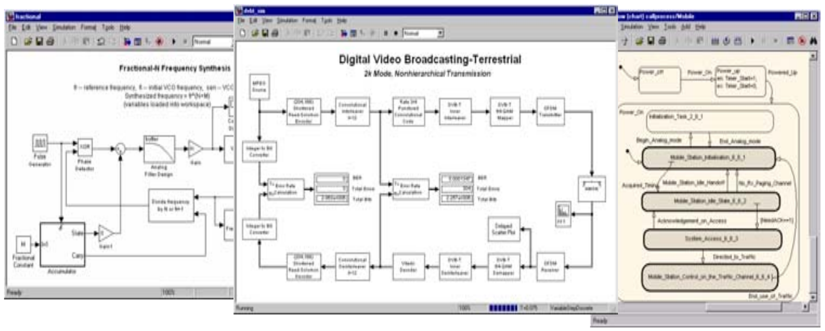
MathWorks Integrated Design Solution



Common tool for all design teams
Simulate component interactions
Test behavior of whole system
No re-design necessary

Modeling system components




Fractional-N Frequency Synthesis
Digital Video Broadcasting-Terrestrial
2k Mode, Nonhierarchical Transmission

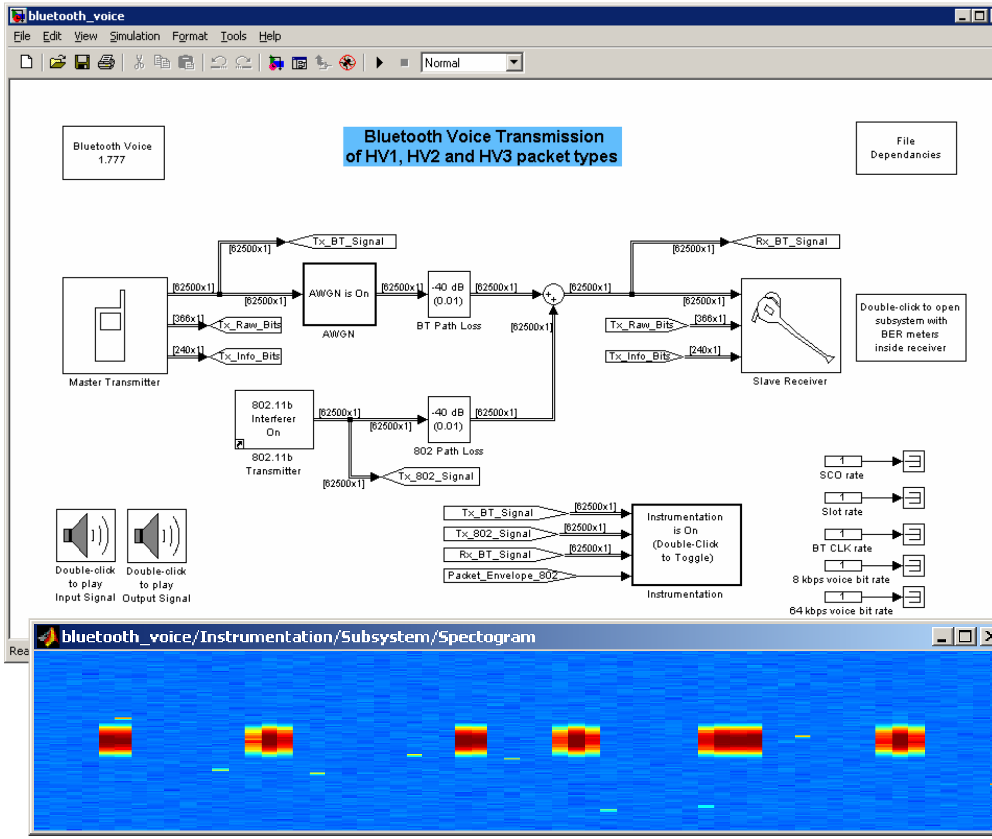
- Analog/Mixed-Signal
  - E.g. PLLs, data converters
  - Continuous time, variable-step ODE solvers
- DSP
  - E.g. Baseband processing, speech processing
  - Discrete time, fast frame-based processing. Bit-true cycle accurate.
- Control Logic
  - E.g. MAC layer, acknowledgement schemes
  - Reactive or event driven state machines

SOPC WORLD 2002

The MathWorks

End-to-end systems

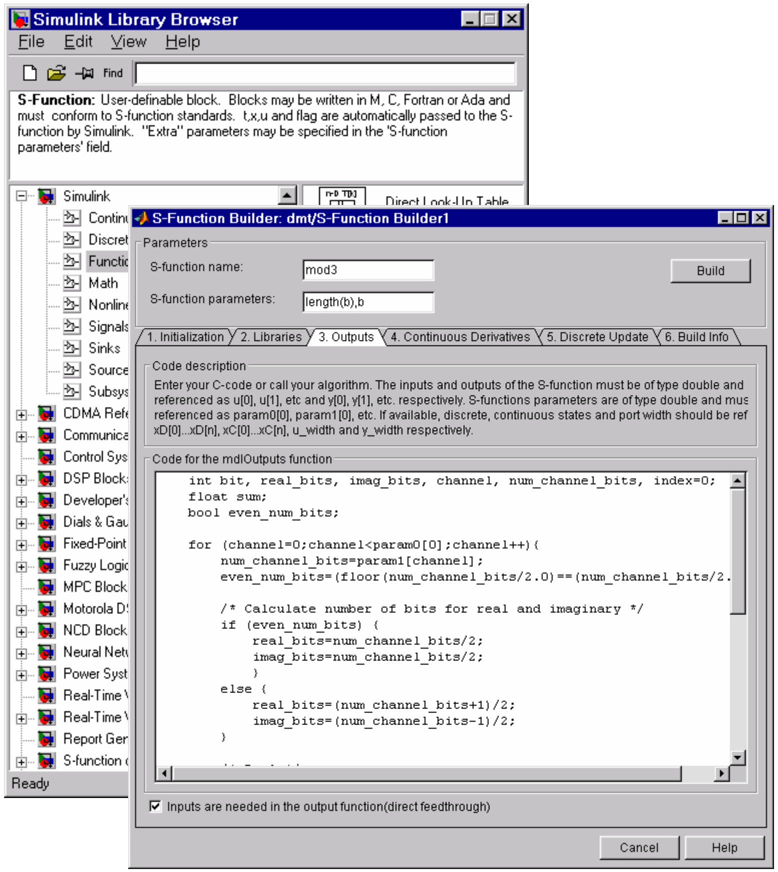
- Digital and analog operations such as coding, interleaving and modulation
- Channel models, error coding, sources, sinks
- Multiple rates for frames, symbols, bit and sample rates
- Synchronization
- Performance testing
- Analog, digital, hybrid, and event-driven simulation



SOPC WORLD 2002

The MathWorks

Co-develop with C Code



S-Function: User-definable block. Blocks may be written in M, C, Fortran or Ada and must conform to S-function standards. t,x,u and flag are automatically passed to the S-function by Simulink. "Extra" parameters may be specified in the 'S-function parameters' field.

Parameters

S-function name: mod3

S-function parameters: length(b),b

Code description

Enter your C-code or call your algorithm. The inputs and outputs of the S-function must be of type double and referenced as u[0], u[1], etc and y[0], y[1], etc. respectively. S-functions parameters are of type double and mus referenced as param0[0], param1[0], etc. If available, discrete, continuous states and port width should be ref xD[0]...xD[n], xC[0]...xC[n], u_width and y_width respectively.

Code for the mdlOutputs function

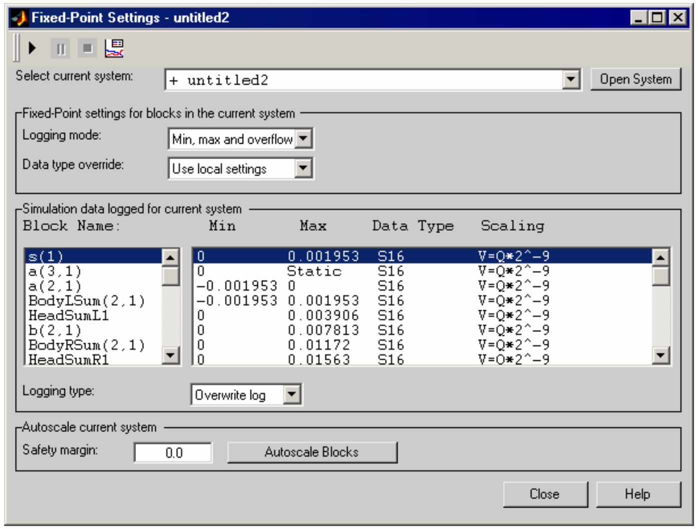
```
    int bit, real_bits, imag_bits, channel, num_channel_bits, index=0;
    float sum;
    bool even_num_bits;

    for (channel=0;channel<param0[0];channel++){
        num_channel_bits=param1[channel];
        even_num_bits=(floor(num_channel_bits/2.0)==(num_channel_bits/2.

        /* Calculate number of bits for real and imaginary */
        if (even_num_bits) {
            real_bits=num_channel_bits/2;
            imag_bits=num_channel_bits/2;
            }
        else {
            real_bits=(num_channel_bits+1)/2;
            imag_bits=(num_channel_bits-1)/2;
        }
```

[✓] Inputs are needed in the output function(direct feedthrough)

- S-function block in Simulink
  - API to specify outputs, state, parameters and sample-times
- S-function builder in Simulink
  - GUI to enter C Code
  - Predefined variables for input, output and states

Fixed Point Simulation

- New integrated fixed point in core Simulink (R13)
- User-definable data types
- Analysis tools
  - log min, max, overflows block-by-block
- Floating point override options



Fixed-Point Settings - untitled2

Select current system: + untitled2

Fixed-Point settings for blocks in the current system

Logging mode: Min, max and overflow

Data type override: Use local settings

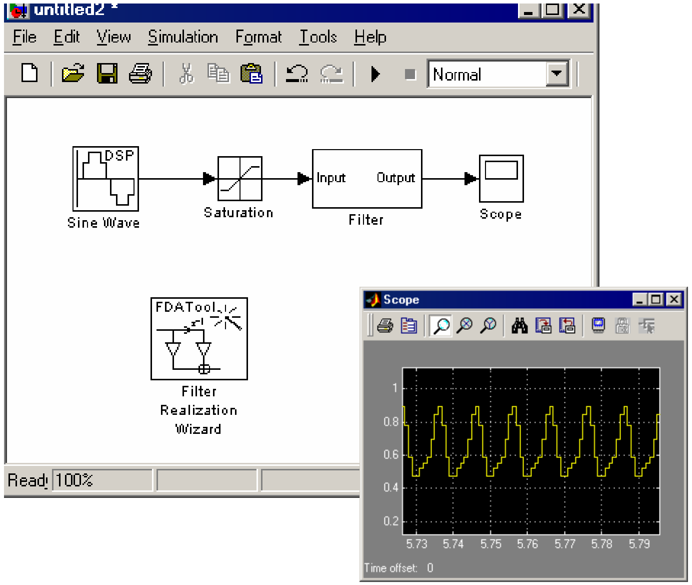
Simulation data logged for current system

Block Name:	Min	Max	Data Type	Scaling
s(1)	0	0.001953	S16	V=Q*2^-9
a(3,1)	0	Static	S16	V=Q*2^-9
a(2,1)	-0.001953	0	S16	V=Q*2^-9
BodyLSum(2,1)	-0.001953	0.001953	S16	V=Q*2^-9
HeadSumL1	0	0.003906	S16	V=Q*2^-9
b(2,1)	0	0.007813	S16	V=Q*2^-9
BodyRSum(2,1)	0	0.01172	S16	V=Q*2^-9
HeadSumR1	0	0.01563	S16	V=Q*2^-9

Logging type: Overwrite log

Autoscale current system

Safety margin: 0.0

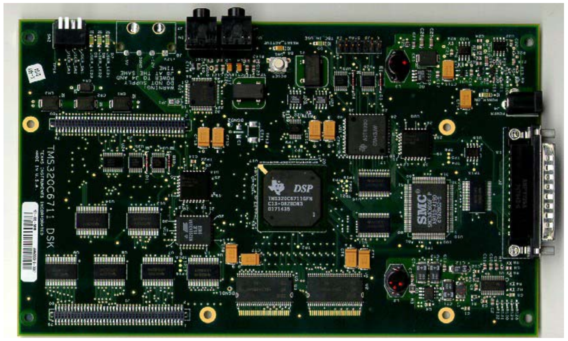


SOPC WORLD 2002

The MathWorks

Links to implementation

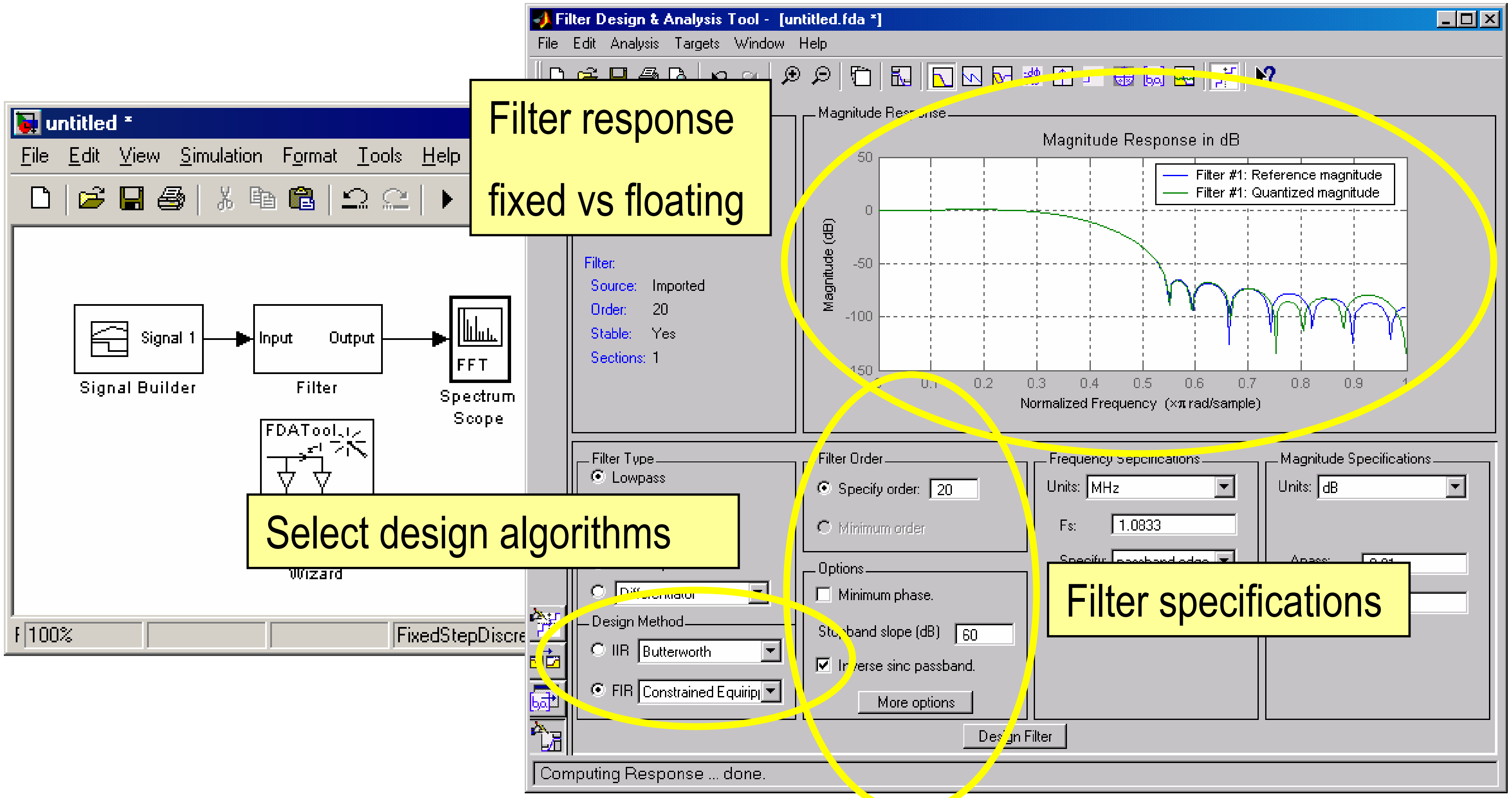
- Real-Time Workshop
  - Automatically generates ANSI C from Simulink
  - Customizable code
  - Rapid Prototyping



- Embedded Targets
  - TI C6000
  - Motorola MPC555

- Altera DSP Builder
  - Bit-true and cycle-true Simulink library for common functions
  - Automatic HDL code generation from a Simulink model
  - Available from Altera

FIR Filter Design: demo



Filter:
Source: Imported
Order: 20
Stable: Yes
Sections: 1

Filter Type: [✓ Lowpass]

Design Method: IIR Butterworth [✓ FIR Constrained Equirip]

Filter Order: [✓ Specify order: 20] Minimum order

Options:
- Minimum phase. ☐
- Stopband slope (dB): 60
- Inverse sinc passband. ✓

Frequency Specifications: Units: MHz; Fs: 1.0833

Magnitude Specifications: Units: dB

Computing Response ... done.

Filter response

fixed vs floating

Select design algorithms

Filter specifications

SOPC WORLD 2002

The MathWorks

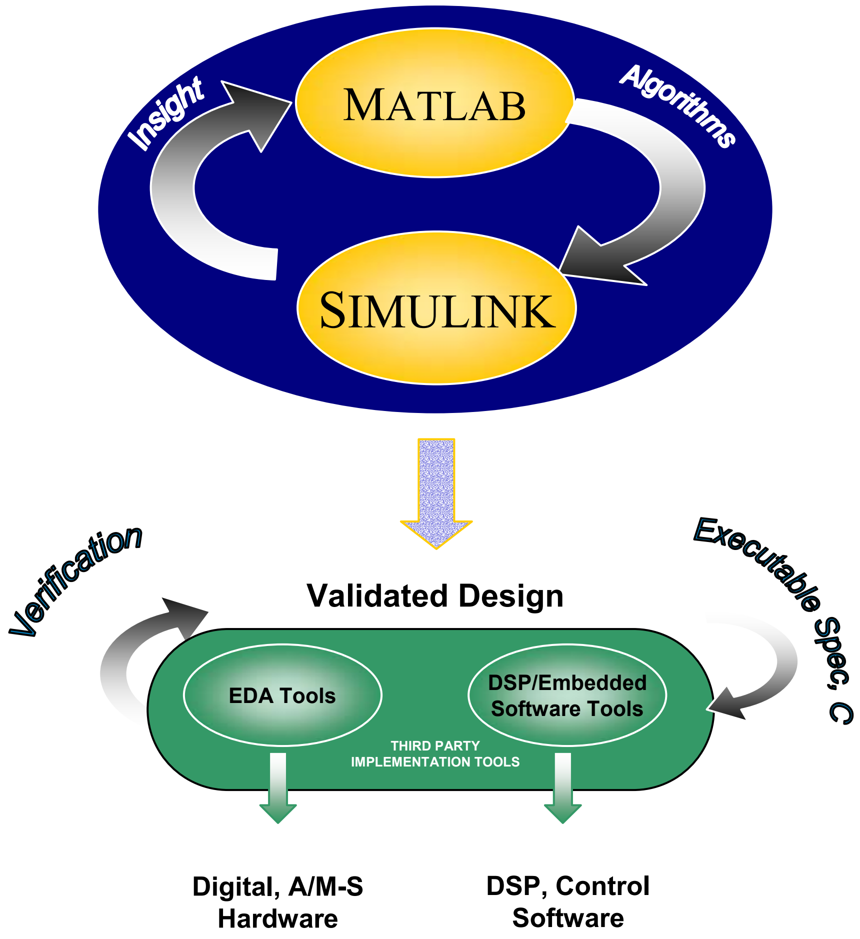
SOPC WORLD 2002

The MathWorks

Summary

ALTERA

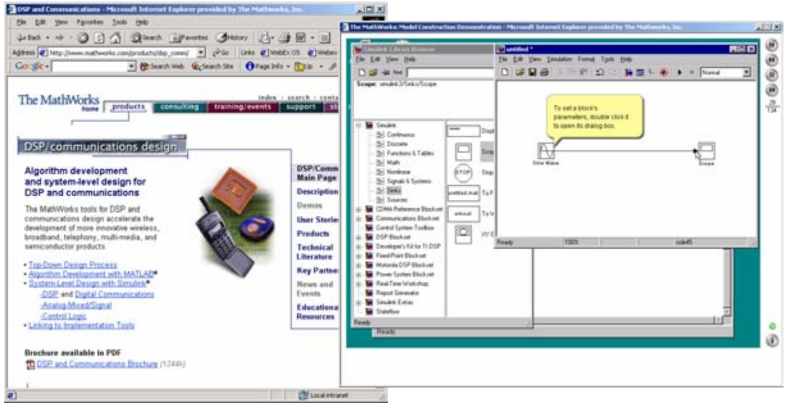
MATLAB and Simulink



- Create validated design
- Use as reference or executable specification to test low-level designs against
- Provide clear specifications
- Detect design flaws early
- Reduce risk and time-to-market

Further Information on Products and Services

- Product information and animated demonstrations
  - www.mathworks.com/products/dsp_comm
- Events
  - www.mathworks.com/dsp_events
  - Regular on-line software demonstrations



Algorithm development and system-level design for DSP and communications

The MathWorks tools for DSP and communications design accelerate the development of more innovative wireless, broadband, telephony, multi-media, and semiconductor products.

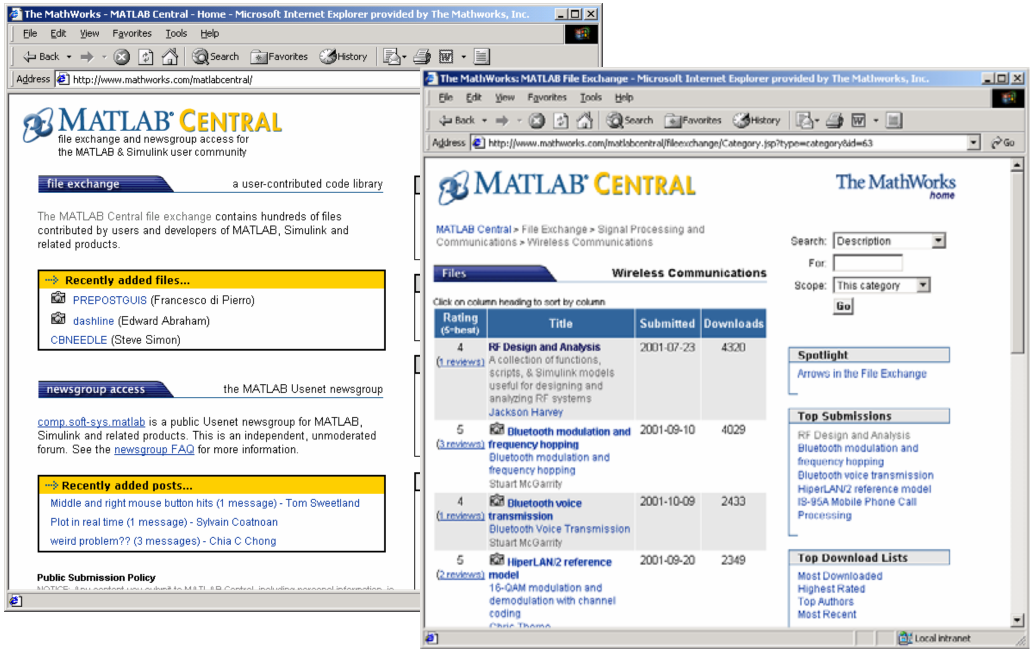
- Top-Down Design Process
- Algorithm Development with MATLAB®
- System-Level Design with Simulink®
  - -DSP and Digital Communications
  - -Analog Mixed/Signal
  - -Control Logic
- Linking to Implementation Tools

Brochure available in PDF

DSP and Communications Brochure (1244k)

MATLAB Central

- www.mathworks.com/matlabcentral
- MathWorks and user contributed models



The MathWorks - MATLAB Central - Home - Microsoft Internet Explorer provided by The Mathworks, Inc.

Address http://www.mathworks.com/matlabcentral/

MATLAB® CENTRAL
file exchange and newsgroup access for
the MATLAB & Simulink user community

file exchange — a user-contributed code library

The MATLAB Central file exchange contains hundreds of files contributed by users and developers of MATLAB, Simulink and related products.

Recently added files...
- PREPOSTGUIS (Francesco di Pierro)
- dashline (Edward Abraham)
- CBNEEDLE (Steve Simon)

newsgroup access — the MATLAB Usenet newsgroup

comp.soft-sys.matlab is a public Usenet newsgroup for MATLAB, Simulink and related products. This is an independent, unmoderated forum. See the newsgroup FAQ for more information.

Recently added posts...
- Middle and right mouse button hits (1 message) - Tom Sweetland
- Plot in real time (1 message) - Sylvain Coatnoan
- weird problem?? (3 messages) - Chia C Chong

Public Submission Policy

The MathWorks: MATLAB File Exchange - Microsoft Internet Explorer provided by The Mathworks, Inc.

Address http://www.mathworks.com/matlabcentral/fileexchange/Category.jsp?type=category&id=63

MATLAB® CENTRAL — The MathWorks home

MATLAB Central > File Exchange > Signal Processing and Communications > Wireless Communications

Files — Wireless Communications

Click on column heading to sort by column

Rating (5=best)	Title	Submitted	Downloads
4 (1 reviews)	**RF Design and Analysis** A collection of functions, scripts, & Simulink models useful for designing and analyzing RF systems Jackson Harvey	2001-07-23	4320
5 (3 reviews)	**Bluetooth modulation and frequency hopping** Bluetooth modulation and frequency hopping Stuart McGarrity	2001-09-10	4029
4 (1 reviews)	**Bluetooth voice transmission** Bluetooth Voice Transmission Stuart McGarrity	2001-10-09	2433
5 (2 reviews)	**HiperLAN/2 reference model** 16-QAM modulation and demodulation with channel coding	2001-09-20	2349

Search: Description
For:
Scope: This category
Go

Spotlight
- Arrows in the File Exchange

Top Submissions
- RF Design and Analysis
- Bluetooth modulation and frequency hopping
- Bluetooth voice transmission
- HiperLAN/2 reference model
- IS-95A Mobile Phone Call Processing

Top Download Lists
- Most Downloaded
- Highest Rated
- Top Authors
- Most Recent