Fixed-Point Toolbox 2

Design and execute fixed-point algorithms and analyze fixed-point data

Fixed-Point Toolbox provides fixed-point data types and arithmetic in MATLAB®. You can use it to design fixed-point algorithms using MATLAB syntax and execute these algorithms at compiled C-code speed. You can reuse these fixed-point algorithms in Simulink® and pass fixed-point data to and from Simulink models to facilitate the simulation, implementation, and analysis of fixed-point systems. You can also generate test sequences for verifying fixed-point software and hardware implementations.

Using Fixed-Point Data Types

Together, Fixed-Point Toolbox and MATLAB provide an environment for developing, implementing, and verifying algorithms for fixed-point designs. The toolbox supports arbitrary word lengths from 2 to 65,535 bits for fixed-point data types, enabling you to capture bit-true fixed-point behaviors in the MATLAB workspace.

Fixed-Point Arithmetic

Fixed-Point Toolbox uses MATLAB syntax to support fixed-point arithmetic. The toolbox provides the following functions and operators:

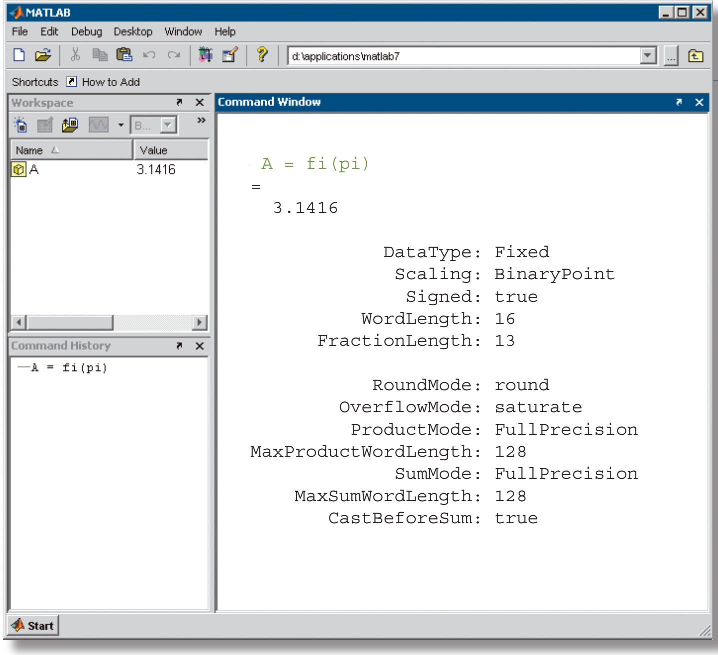
- Basic arithmetic operations, including addition, subtraction and multiplication
- A division function
- A square-root function
- Relational, logical, and bitwise operators
- Statistical functions, such as `min` and `max`
- User-selected overflow and rounding modes to govern all arithmetic

The toolbox lets you use four modes for performing fixed-point arithmetic:

- Fully automatic mode, enabling you to use up to 65,535 bits for the sum or product
- Fully specified mode, letting you indicate the word length and binary point location for the sum or product
- Two partially automatic modes, in which the binary point location is set automatically but you can specify the word length of the sum or product

KEY FEATURES

- Support for fixed-point data types in MATLAB
- Fixed-point arithmetic and logic operators
- Support for exchanging fixed-point data between MATLAB and Simulink
- Tools for floating-to-fixed-point conversion, including data logging and data-type override
- Ability to accelerate execution speeds of fixed-point algorithms in the MATLAB workspace



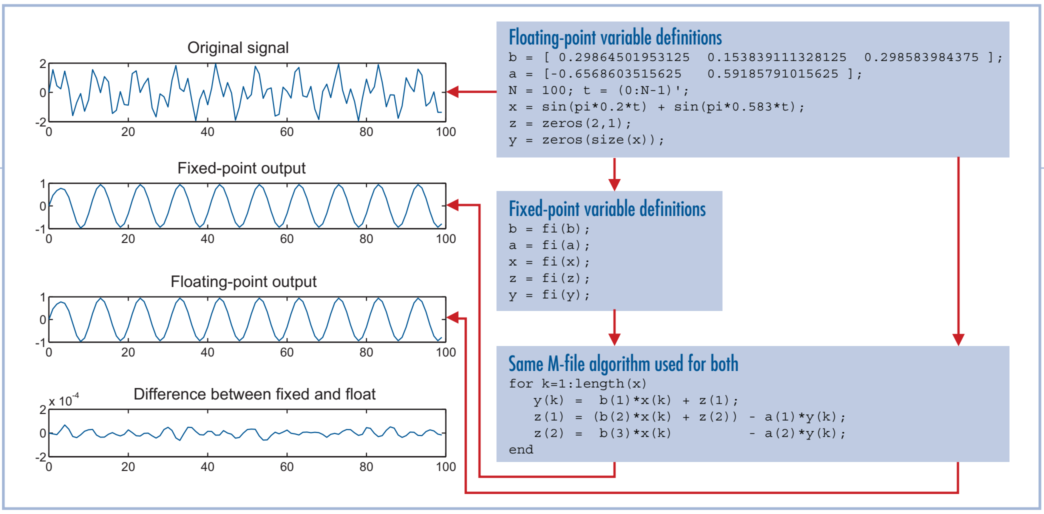
```
—A = fi(pi)
```

```
 A = fi(pi)
=

  3.1416

              DataType: Fixed
               Scaling: BinaryPoint
                Signed: true
            WordLength: 16
        FractionLength: 13

             RoundMode: round
          OverflowMode: saturate
           ProductMode: FullPrecision
  MaxProductWordLength: 128
               SumMode: FullPrecision
      MaxSumWordLength: 128
         CastBeforeSum: true
```

Fixed-point representation of π is created with a default 16-bit word length and best-precision fraction length. Other properties of the fixed-point object can also be defined.

The MathWorks

Accelerating the pace of engineering and science



Floating-point variable definitions

```
b = [ 0.29864501953125  0.153839111328125  0.298583984375 ];
a = [-0.6568603515625   0.59185791015625 ];
N = 100; t = (0:N-1)';
x = sin(pi*0.2*t) + sin(pi*0.583*t);
z = zeros(2,1);
y = zeros(size(x));
```

Fixed-point variable definitions

```
b = fi(b);
a = fi(a);
x = fi(x);
z = fi(z);
y = fi(y);
```

Same M-file algorithm used for both

```
for k=1:length(x)
   y(k) =  b(1)*x(k) + z(1);
   z(1) = (b(2)*x(k) + z(2)) - a(1)*y(k);
   z(2) =  b(3)*x(k)         - a(2)*y(k);
end
```

Plot of the algorithm output in both 16-bit fixed point and double-precision floating point. Using Fixed-Point Toolbox, you can develop data-type-independent algorithms and run them with fixed-point variables.

Using the built-in plotting and visualization functions in MATLAB, you can visualize fixed-point data objects created in Fixed-Point Toolbox and analyze the results of your fixed-point design.

Fixed-Point Data Exchange Between MATLAB and Simulink

The fixed-point operations in Fixed-Point Toolbox complement the fixed-point capabilities in Simulink (available separately). You can use fixed-point data objects to pass fixed-point data between the MATLAB workspace and Simulink models. Fixed-Point Toolbox also supports the design, analysis, and implementation of fixed-point digital filters with Filter Design Toolbox (also available separately). You can quantize filter coefficients and other data with Fixed-Point Toolbox and then use Filter Design Toolbox functions to construct an appropriate filter structure.

Floating-to-Fixed-Point Conversion

Fixed-Point Toolbox provides analysis tools for efficiently converting a design from floating-point to fixed-point representation. The data logging tools let you record minimum or maximum data values and pinpoint when overflow or underflow occurs during fixed-point operations. The data-type override tools let you analyze a fixed-point algorithm by switching the data type of variables between floating-point or fixed-point. With this analysis, you can observe the dynamic range of variables involved in your M-code and ensure that the algorithm behaves consistently in floating-point and fixed-point representations.

Accelerating the Execution Speed of Fixed-Point Algorithms in MATLAB

The `emlmex` function of Fixed-Point Toolbox lets you significantly accelerate the execution speed of fixed-point algorithms. With `emlmex` you can generate a compiled MEX version of your M-code, provided that the M-code is composed from a defined subset of MATLAB language. This MEX version runs at compiled C-code speed, much faster than the original M-code.

Required Products

MATLAB

Related Products

Filter Design Toolbox. Design and analyze fixed-point, adaptive, and multirate filters

Signal Processing Blockset. Design and simulate signal processing systems and devices

Signal Processing Toolbox. Perform signal processing, analysis, and algorithm development

Simulink® Fixed Point. Design and simulate fixed-point systems

For more information on related products, visit *www.mathworks.com/products/fixed*

Platform and System Requirements

For platform and system requirements, visit *www.mathworks.com/products/fixed* ■

Sample Functions

Construct a fixed-point object

Perform basic fixed-point arithmetic

Perform relational, logical, and bitwise operations

Convert to/from binary, hex, and native integers

Estimate the dynamic range of data using double overrides and data logging

Resources

VISIT
www.mathworks.com

TECHNICAL SUPPORT
www.mathworks.com/support

ONLINE USER COMMUNITY
www.mathworks.com/matlabcentral

DEMOS
www.mathworks.com/demos

TRAINING SERVICES
www.mathworks.com/training

THIRD-PARTY PRODUCTS AND SERVICES
www.mathworks.com/connections

WORLDWIDE CONTACTS
www.mathworks.com/contact

E-MAIL
info@mathworks.com

The MathWorks
Accelerating the pace of engineering and science

91206v01 03/07