MathWorks

MathWorks
Accelerating the pace of engineering and science

4月 - 北京 · 上海 · 深圳

2015 MATLAB 巡回研讨会

技术融合的时代

运用MATLAB加速嵌入式算法开发

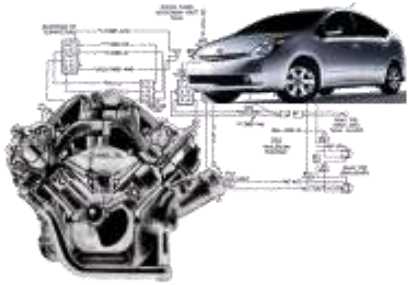
MathWorks China
应用工程师
单博

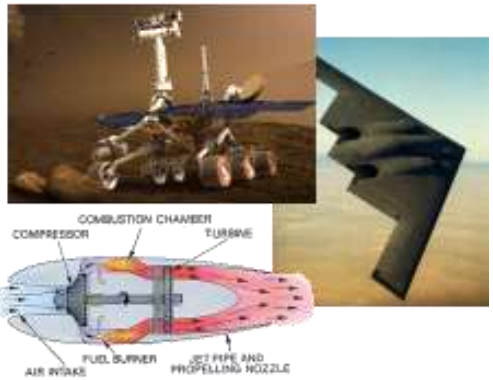
主要内容

- 嵌入式应用仿真
- 浮点模型转化为定点模型的工作流程
  - 自动收集数据并建议数据类型
  - 推导数据类型
- 将定点模型自动转换为嵌入式C代码
  - 代码生成与优化
  - 验证 Polyspace
  - 硬件连接
- 国内典型用户案例分析

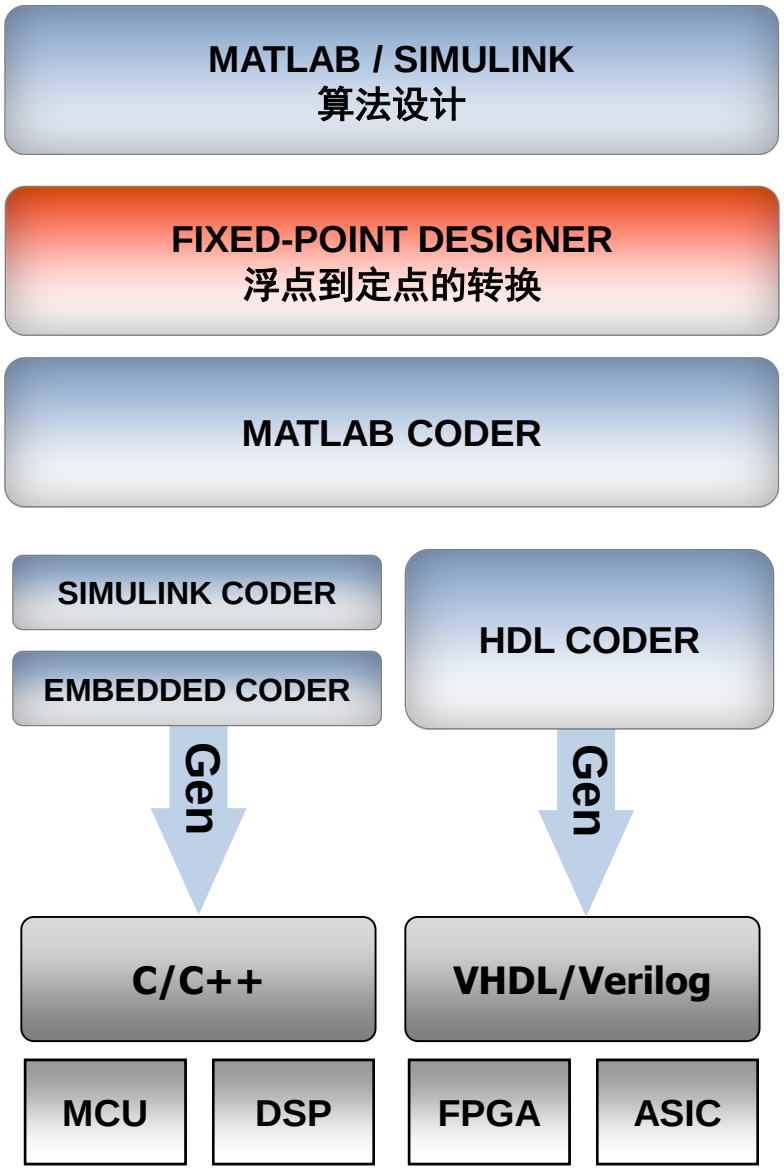
嵌入式系统

- 手机、pad
- 家电
- 网络设备
- 工业控制
- 仪器仪表
- 医疗
- 汽车
- 飞机
- 机器人
- 物联网





嵌入式算法的设计挑战

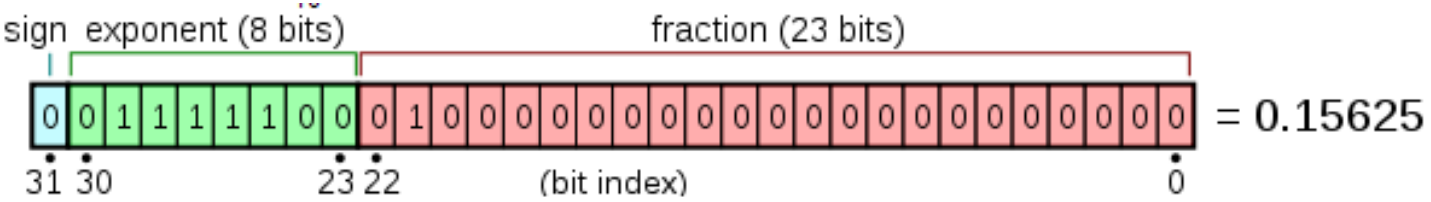


设计狮

定点化过程会占据总设计时间超过25%+

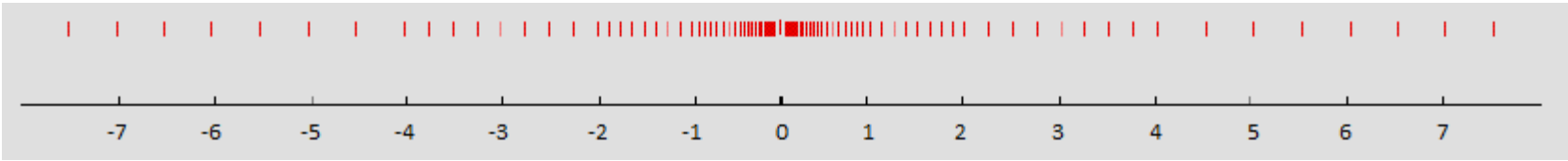
What is floating-point?

- Characterized by SIGN bit, MANTISSA(Fraction) and EXPONENT IEEE 754 Single Precision format (Normalized): 32 bits word size(single)



$$value = (-1)^{sign}(1 + \sum_{i=1}^{23} b_{-i}2^{-i}) \times 2^{(e-127)}$$

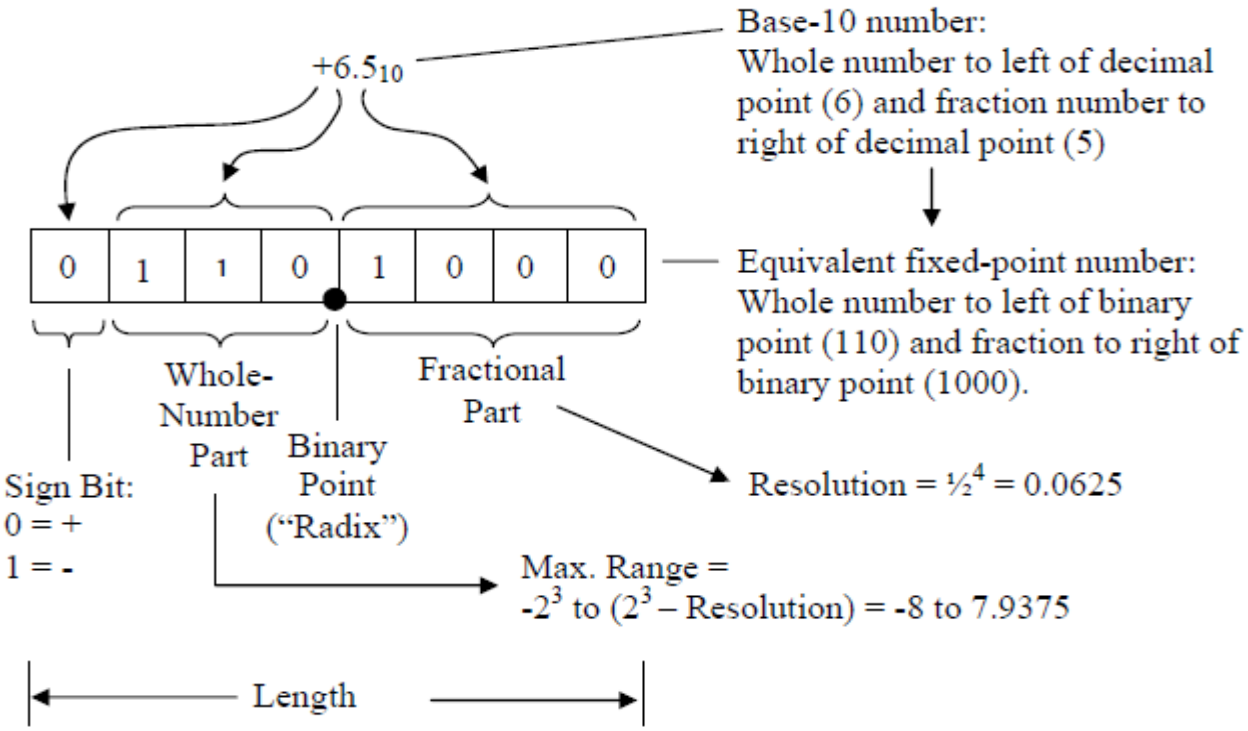
- Use of three separate fields increases both range (Exponent size) and precision (Fraction size) of floating point numbers

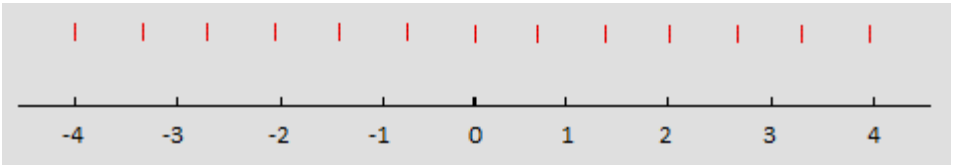


Above picture from http://en.wikipedia.org/wiki/Single_precision_floating-point_format

What is fixed-point?

- Characterized by single WORD with fixed RADIX Point
- Use fractional numbers without floating point
- For a fixed size, trade-off between Precision and Range





嵌入式目标器件

- DSP/MCU (TI, Analog Devices, etc.)
  - 定点DSP比浮点DSP便宜很多
  - 定点DSP的功耗更低
  - 定点DSP的时钟频率更高
  - 固定字长

- FPGA (Xilinx, Altera, etc.)
  - 在定点FPGA实现中，字长每增加1个比特都意味着消耗更多的片上资源和功耗
  - 设计师可以改变字长

Example: 定点C手工实现

```
void differentialEq( void )
{
   /* Implements a fixed point first order difference equation    */

   int Prod;
   long Accum;
   static short lastVal=0;
   short a=0x7eb8; // 0.99 in s16,15
   short oneminusa=0x0148; // .01 in s16,15
   short temp;

   Prod = gAlg_in1 * gAlg_in1;
   temp = Prod >> 15;
   Accum = a*lastVal + oneminusa*temp;

   gAlg_out1 = (short)(Accum >> 15);
   lastVal = gAlg_out1;
}
```

把变量转换成整形

需要很多注释帮助理解代码

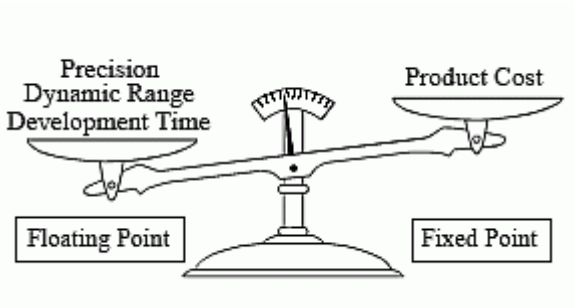
跟踪二进制小数点位置

没有饱和或取整

DSP/MCU/FPGA上需要做的定点工作

- 芯片做整数运算，不是定点运算.
- 定点工作需要你自己做.
- 加减法:
  - 你需要对齐二进制小数点(>> or <<)，然后相加.
- 乘法:
  - 整数相乘，然后你解释乘积的二进制小数点.
- 除法:
  - 你很可能用不了“/”. 你要自己写个函数或者调用库函数.
- 开方:
  - 不能用标准库函数，你要自己写一个函数.
- 取整/饱和:
  - 你想要？你自己做.

Fixed Point Tradeoffs



Consideration	Floating Point	Fixed Point	Fixed Point with MathWorks Tools
RAM/ROM消耗	↑	↓	↓
执行速度	↓	↑	↑
硬件功耗	↑	↓	↓
硬件成本	↑	↓	↓
开发时间	↓	↑	↓
实现复杂度	↓	↑	↓
出错率	↓	↑	↓

定点的弊端及解决方案

弊端1

开发时间更长

解决方案

采用仿真、自动化验证和自动化量化工具将会缩短开发时间和减少开发资源

弊端2

因为实现复杂经常引入错误

解决方案

采用快速原型，在环测试，产品级代码生成手段能明显地帮助减少错误

弊端3

动态范围变小引入量化误差

解决方案

合理的选择定标（小数点位置）和字长（硬件限制）能减少量化误差

MATLAB/Simulink 定点化工作流程

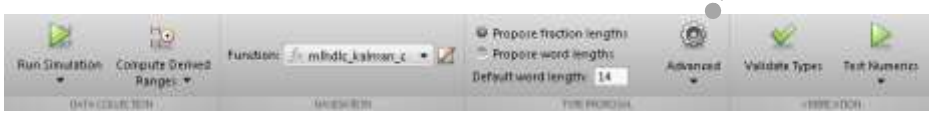
已有代码
(浮点)

已有测试激励
(定点)

定点转换的
准备工作

Make Project

Set Input Type



收集取值范围

自动建议数据类型

生成定点MATLAB代码

测试

满足需求?

Yes

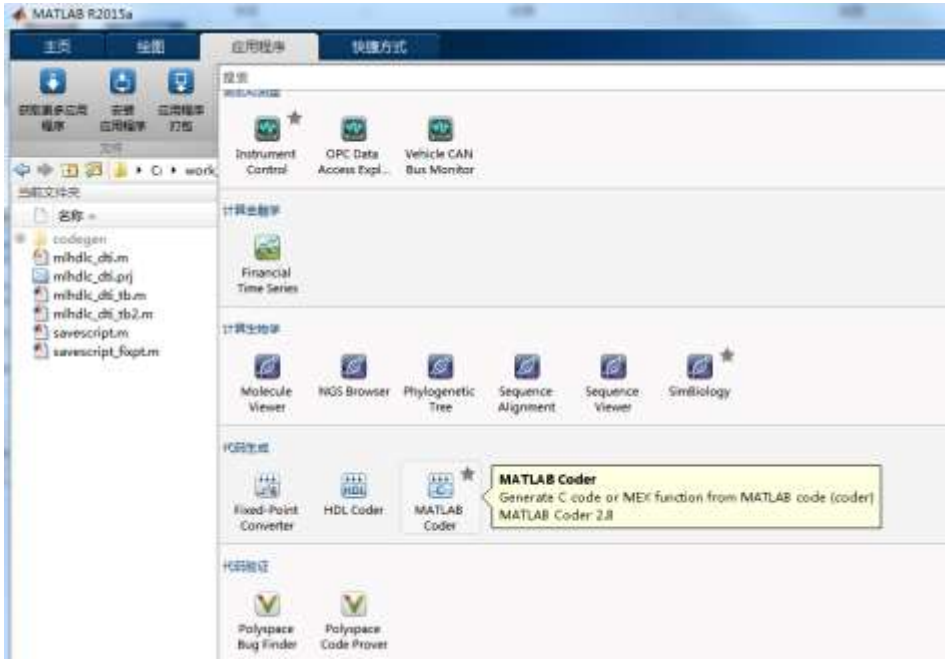
No

生成代码

例子

- 工作流程: 浮点模型-> 定点模型-> C
- 仿真 vs 推导

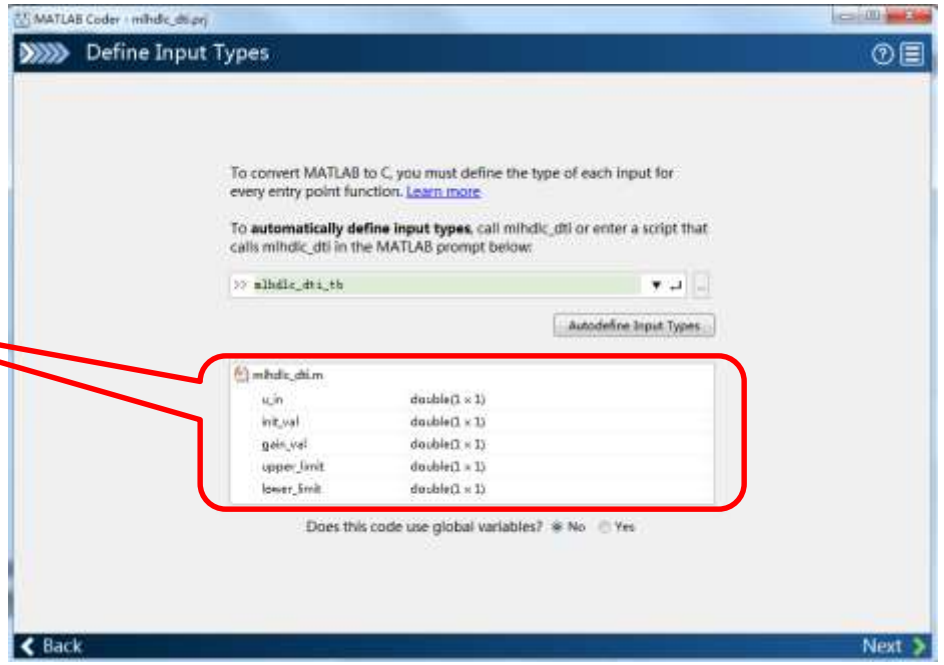
1 打开MATLAB Coder APP



2 使能Fixed-point conversion
选择被测m文件
- 定点转换
- 生成C代码

例子

3 选择testbench文件
- 自动识别被测文件的输入输出数据类型



实现约束
- 多形性
- 内存分配
- 矩阵行列处理
- 定点数据类型

```
function a= foo(b,c)
a = b * c;
```

c

```
double foo(double b, double c)
{
        return b*c;
}
```

```
void foo(const double b[15],
         const double c[30], double a[18])
{
  int i0, i1, i2;
for (i0 = 0; i0 < 3; i0++) {
   for (i1 = 0; i1 < 6; i1++) {
     a[i0 + 3 * i1] = 0.0;
     for (i2 = 0; i2 < 5; i2++) {
       a[i0 + 3 * i1] += b[i0 + 3 * i2] * c[i2 + 5 * i1];
     }
   }
  }
}
```

例子

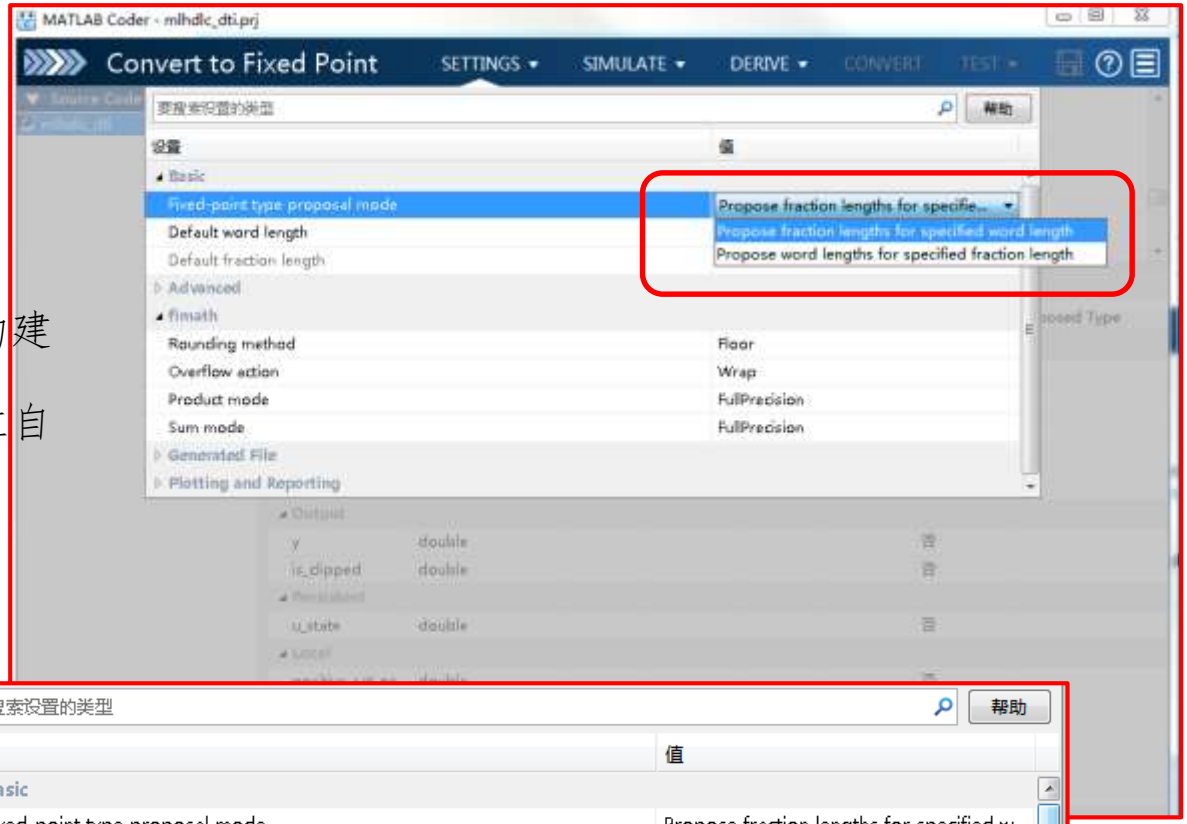
4 定点化配置

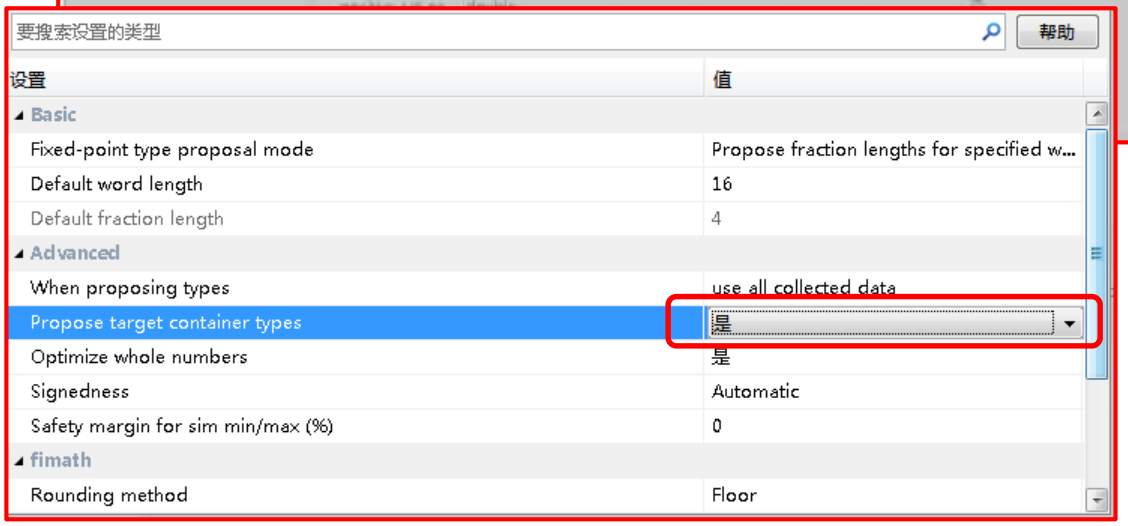
定点化模式

- C代码生成：固定字长自动建议小数位
- HDL代码生成：固定小数位自动建议字长

以目标器件类型

- 字长为8bit的整数倍

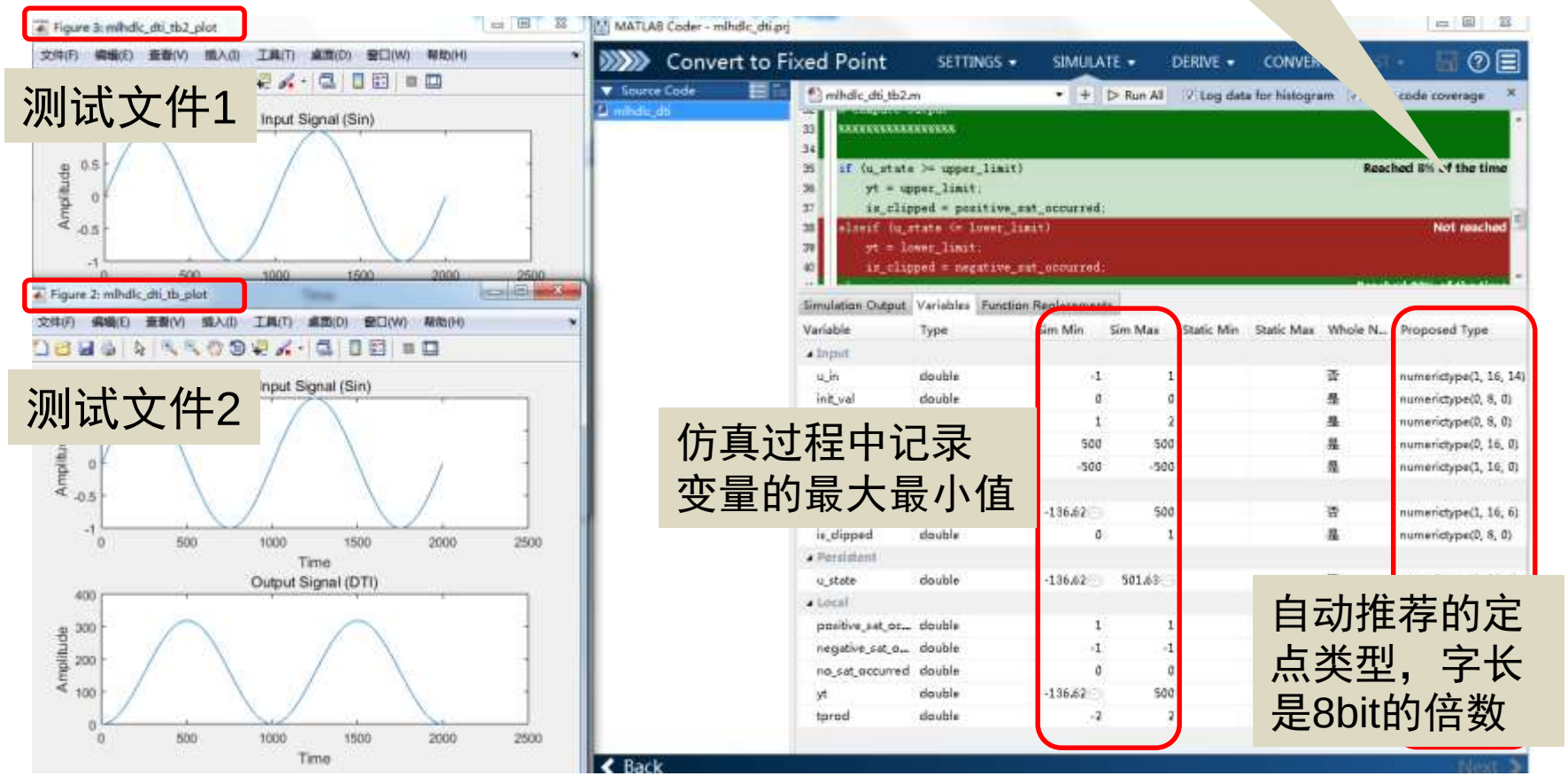




例子：用仿真结果自动建议定点类型

多个测试文件批量仿真

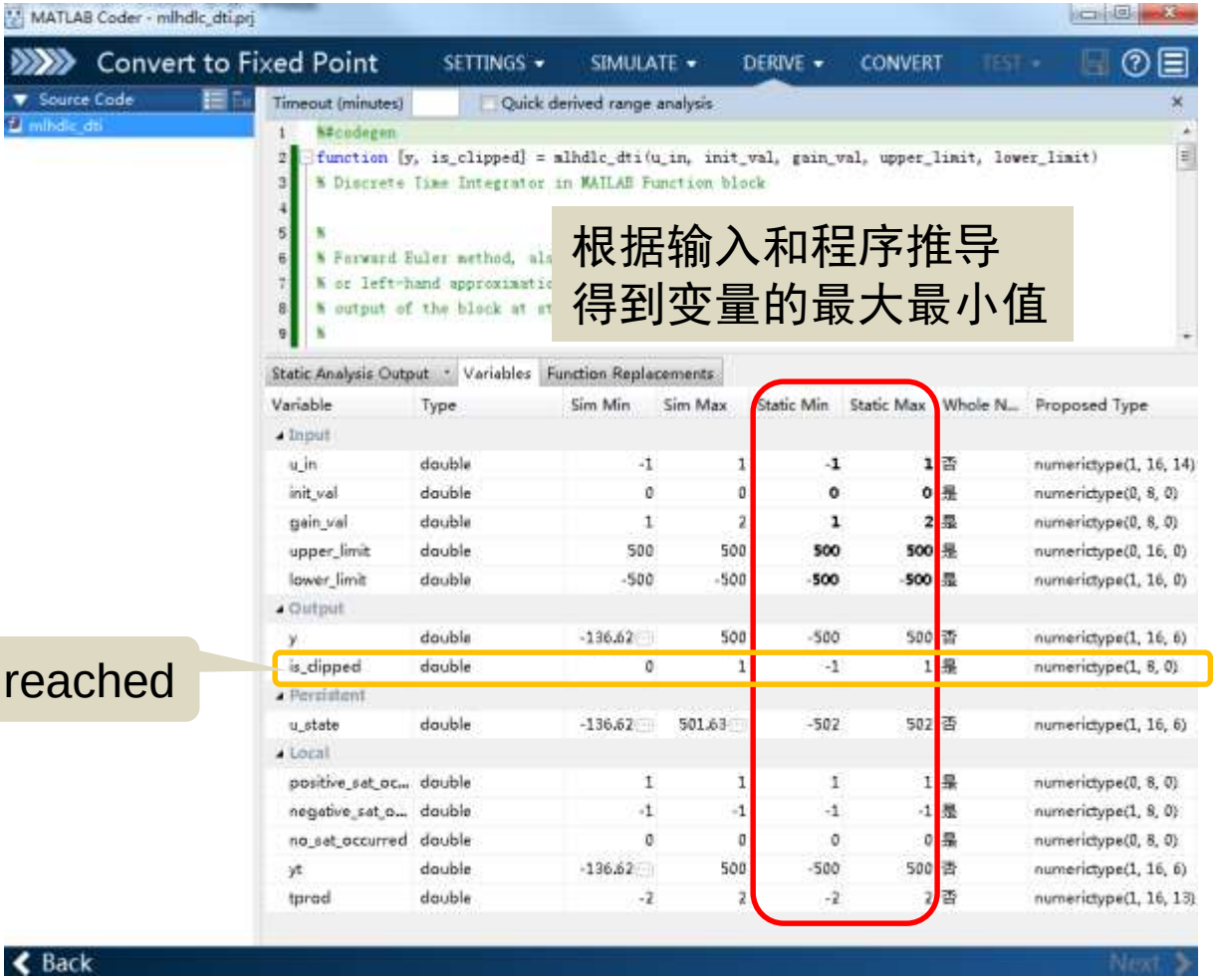
测试代码覆盖率
用颜色区分



仿真过程中记录
变量的最大最小值

自动推荐的定点类型，字长是8bit的倍数

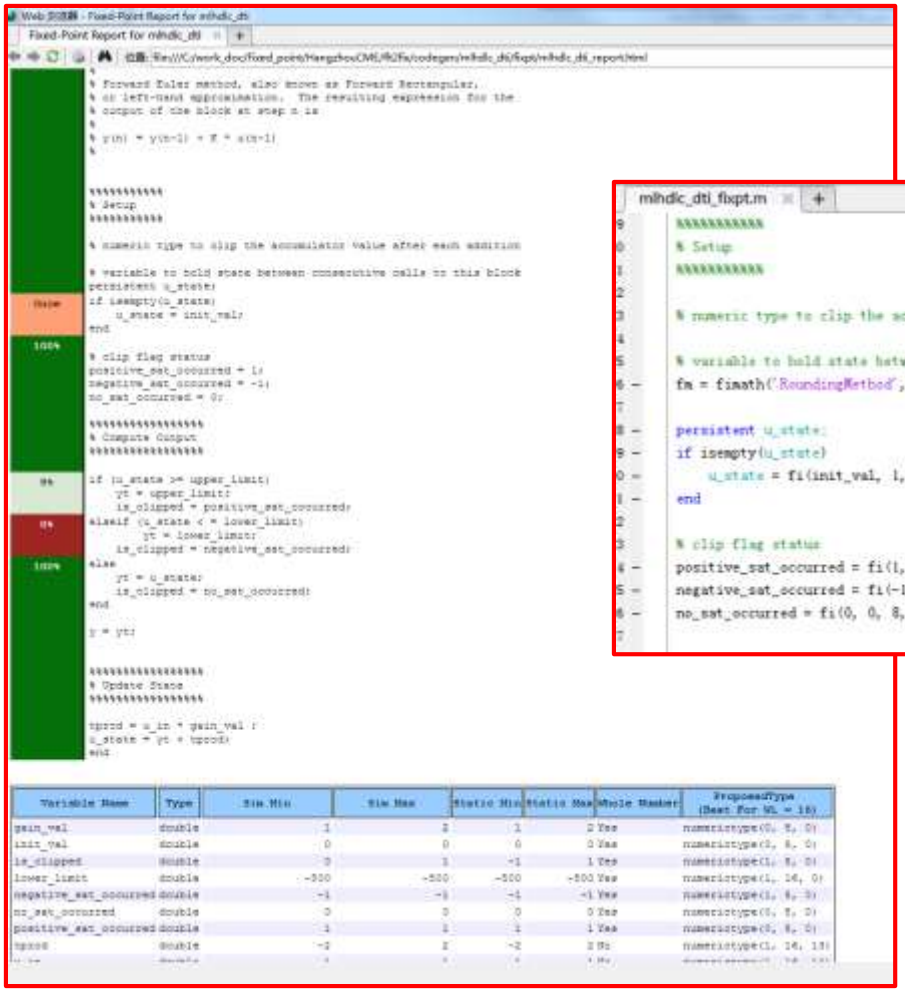
例子：推导并自动建议定点类型



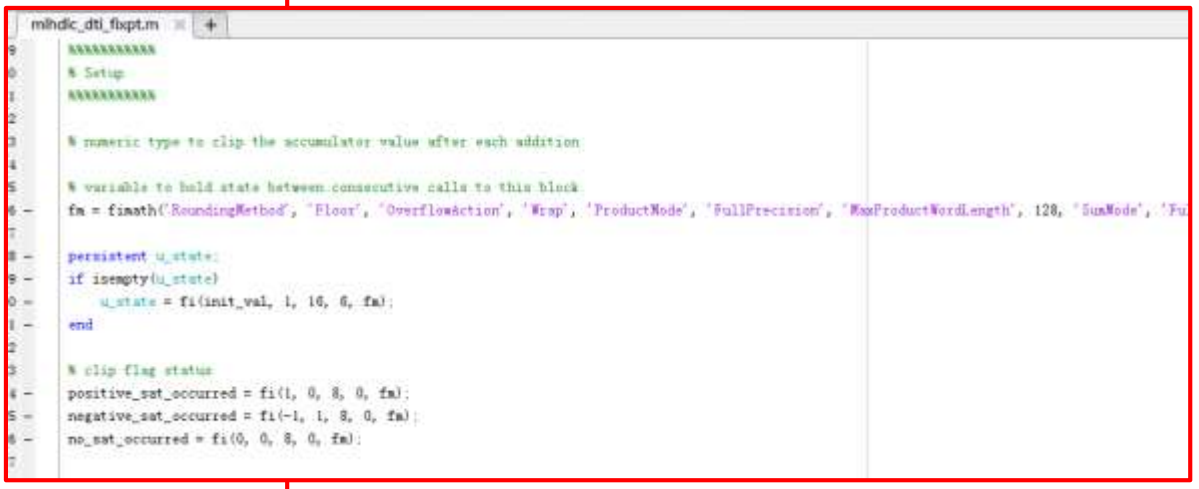
Not reached

例子：接受定点化

自动生成定点转换报告

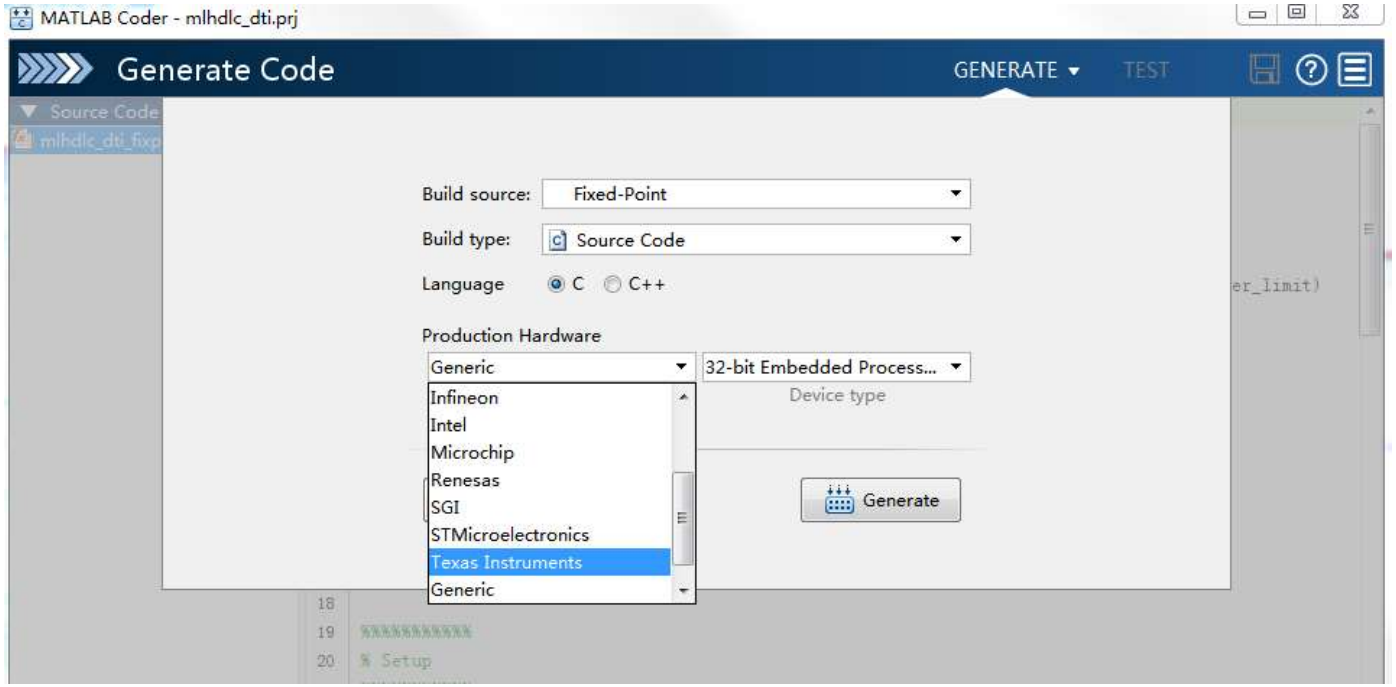


自动生成定点的m文件
用于后续系统定点仿真

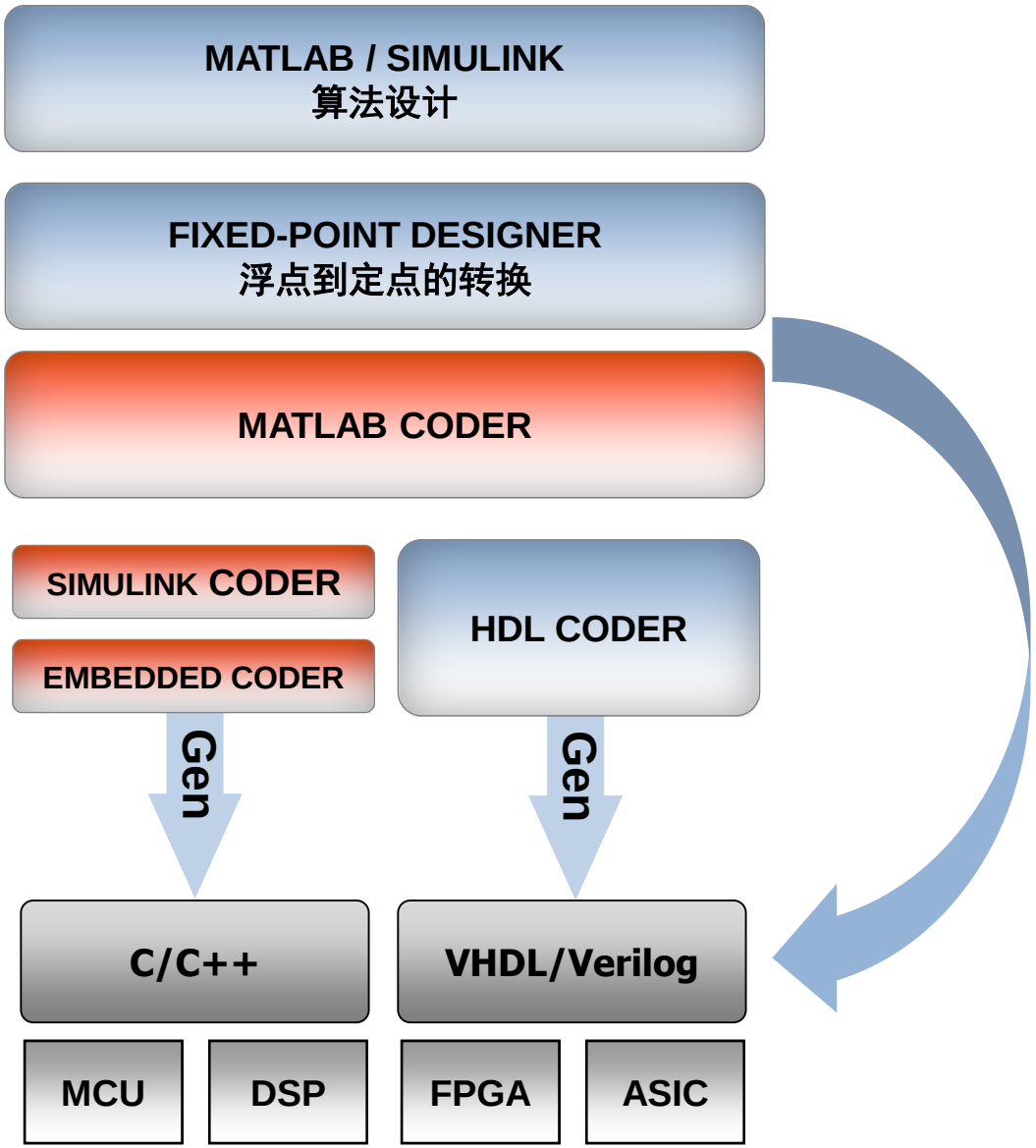


例子：自动生成C代码

选择目标硬件器件
生成C源代码
知识产权归客户



嵌入式算法的设计挑战



设计狮

程序猿

手工编码

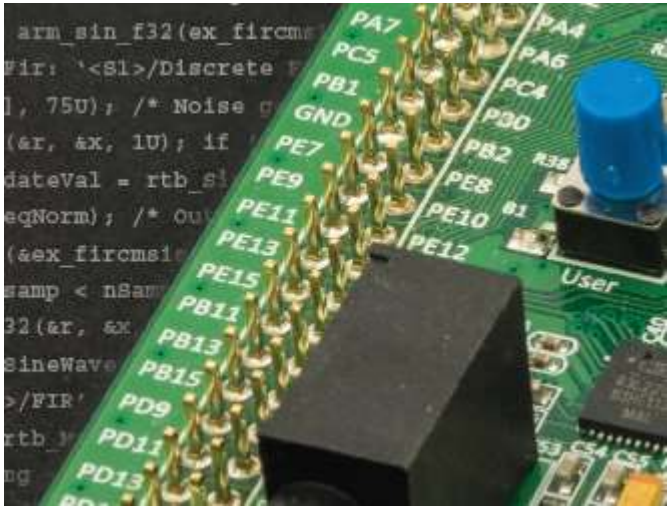
- 费时
- 手写代码失误高
- 不易于保持 MATLAB 参考代码与 C 代码的一致性
- 不易于在开发阶段修改需求

嵌入式代码生成工具
Embedded Coder

- 从 MATLAB, Simulink, 和Stateflow 生成C/C++ code
- 针对嵌入式系统优化
- 可配置到多种器件
  - 快速原型开发板
  - 用于汽车控制器的MCUs
  - 用于信号处理系统的DSPs
  - 用于消费电子的ARMs
- 支持行业标准
  - DO-178, IEC 61508, ISO 26262, and EN 50128
  - AUTOSAR, ASAP2, and MISRA-C

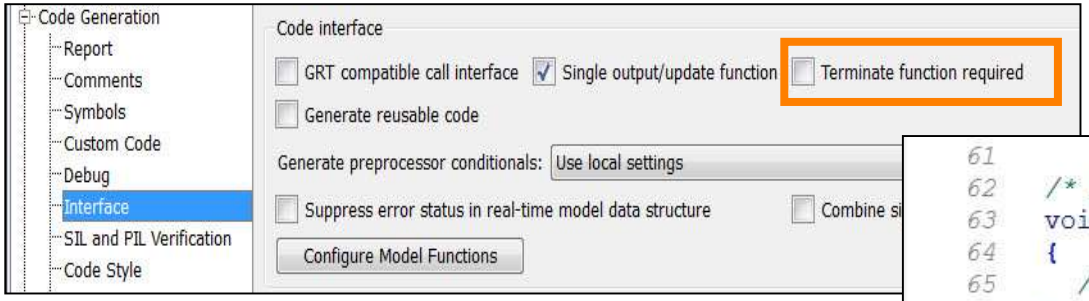
主要特性

- 用data dictionary管理存储类型，类型定义，别名
- 针对处理器优化
  - ARM Cortex-A and M
  - Intel IPP and Power Architecture SIMD
  - Customizable for any device
- 支持多速率、多任务、多核执行，包含或不包含RTOS
- 软件在环、处理器在环测试
  - 模型和代码结果比对
  - 代码剖析
  - 可集成代码覆盖率工具
- 定制注释、代码报告，支持需求文档、模型、代码之间的双向跟踪



移除不需要的代码或数据支持

- Removing initialization code *(Optimization pane)*
- Removing termination code *(Interface pane)*
- Removing data support *(Interface pane)*
- Disabling MAT-File Logging *(Interface pane)*
- Conditional Input Branch Execution *(Optimization pane)*



No termination function

```
61
62  /* Model initialize function */
63  void optim_example_initialize(void)
64  {
65    /* (no initialization code required) */
66  }
67
68  /*
69   * File trailer for generated code.
70   *
71   * [EOF]
72   */
73
```

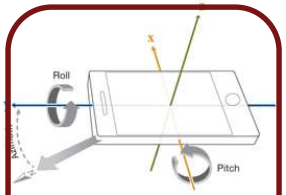
MATLAB /Simulink支持目标硬件

低成本硬件、免费硬件支持包


Arduino®


Raspberry Pi™


LEGO® NXT



iphone


BeagleBoard


PandaBoard


Gumstix® Overo


Microsoft Kinect


Zynq


TI C6000


USRP SDR


iRobot Create

http://www.mathworks.cn/hardware-support/

代码验证

Polyspace C / C ++产品系列

- Polyspace Bug Finder
  - 快速找到嵌入式软件中的错误
  - 检查代码符合MISRA和JSF编码规则
  - 供软件工程师日常使用
- Polyspace Code Prover
  - 证明在给定的运行条件下，那些代码是 安全可靠的
  - 不需要穷举执行程序就可发现代码的 运行时缺陷, 边界条件 和 冗余代码
  - 深度剖析代码的运行时行为和变量的数据范围
  - 证明软件内存共享的安全性
  - 采用数学分析的方法 – 不会漏报或误报任何运行时软件缺陷

自动代码生成已成为行业趋势

User Story

通信与信号处理领域新特性

新产品

Antenna Toolbox(15a)
Vision HDL Toolbox(15a)
LTE System Toolbox(14a)
Phased Array System Toolbox(11a)

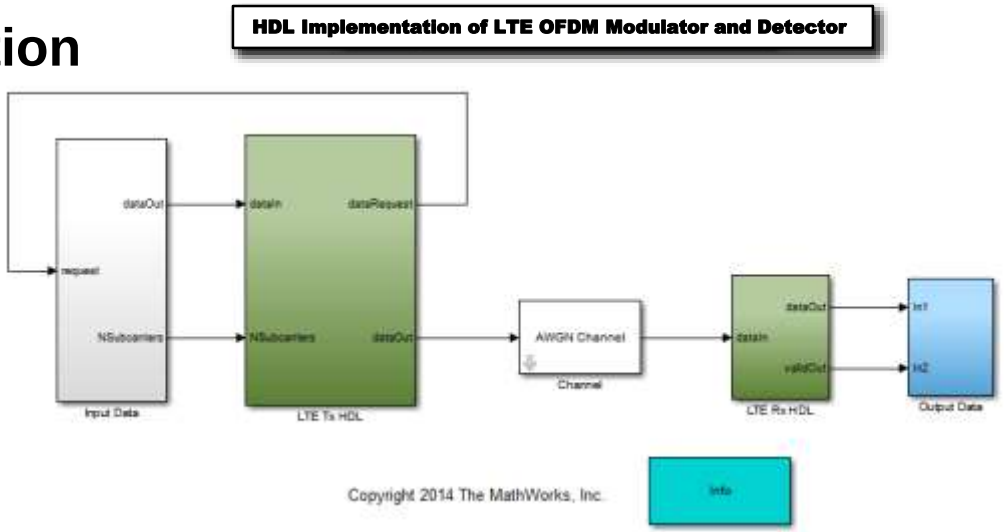
大调整

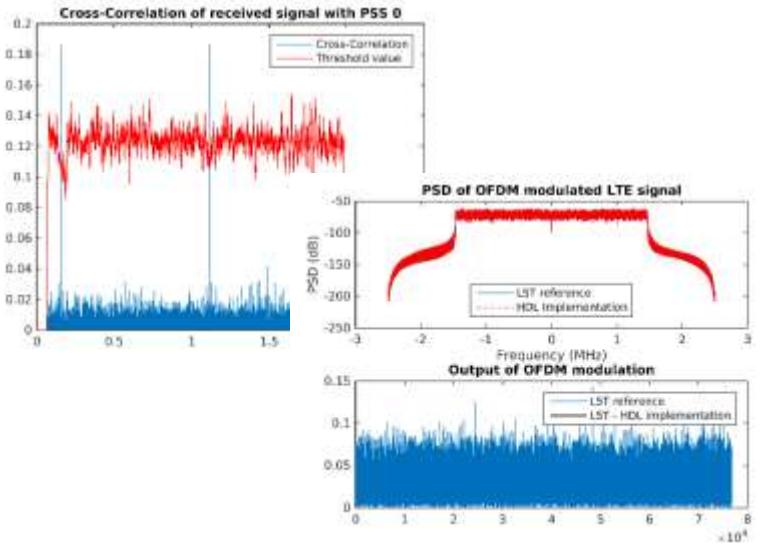
Fixed-point Designer(13a)
HDL Coder(12a)
Embedded Coder(11a)
Communication System Toolbox(11a)
DSP System Toolbox(11a)
Computer Vision System Toolbox(10a)

HDL Implementation of LTE OFDM Modulator and Detector

HDL optimized implementation with golden reference verification

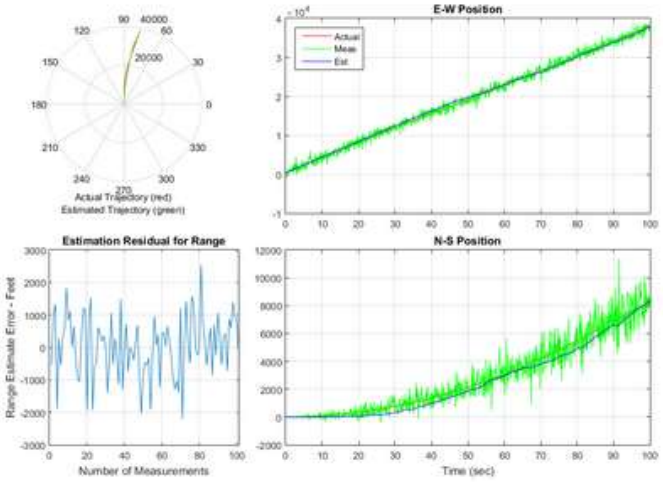
- Implementation of an LTE OFDM Modulator and Detector optimized for HDL code generation
- Verification of transmitter HDL implementation against a golden reference waveform using LTE System Toolbox
- Initial receiver detection and synchronization stages implemented in HDL coder



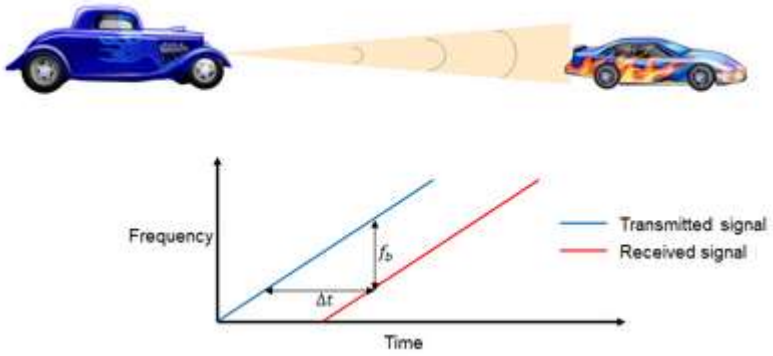


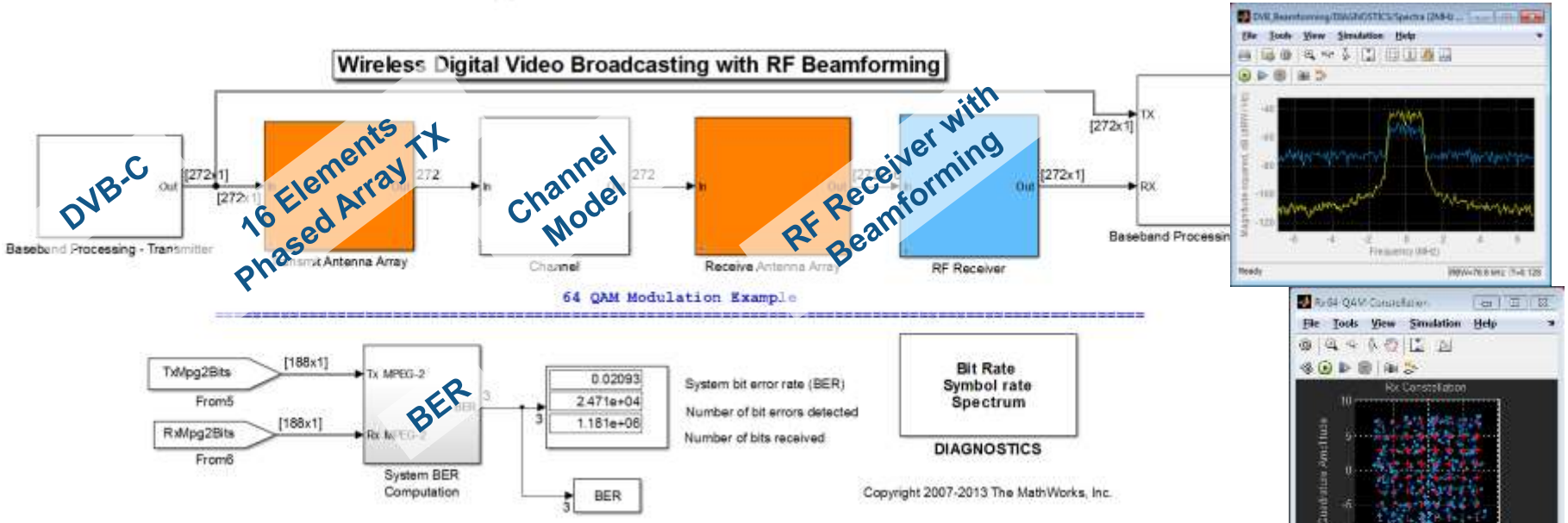
阵列信号处理

Radar tracking



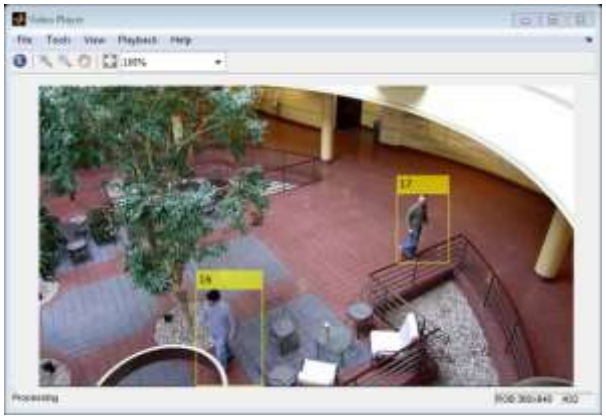
Automotive Adaptive Cruise Control



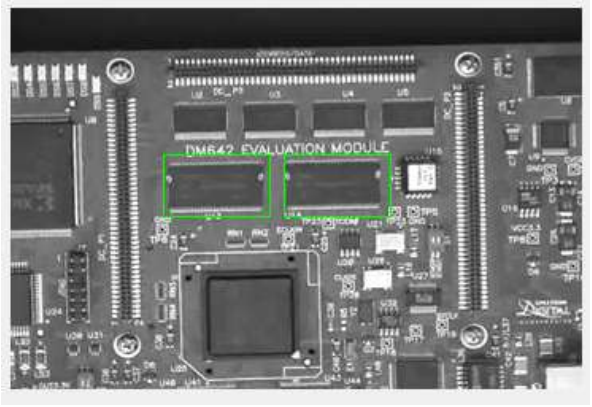


图像处理

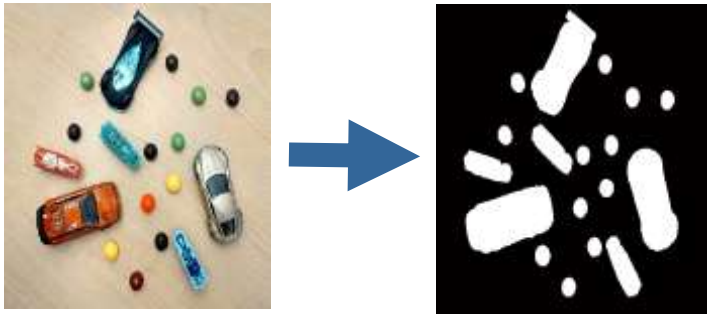
多目标跟踪



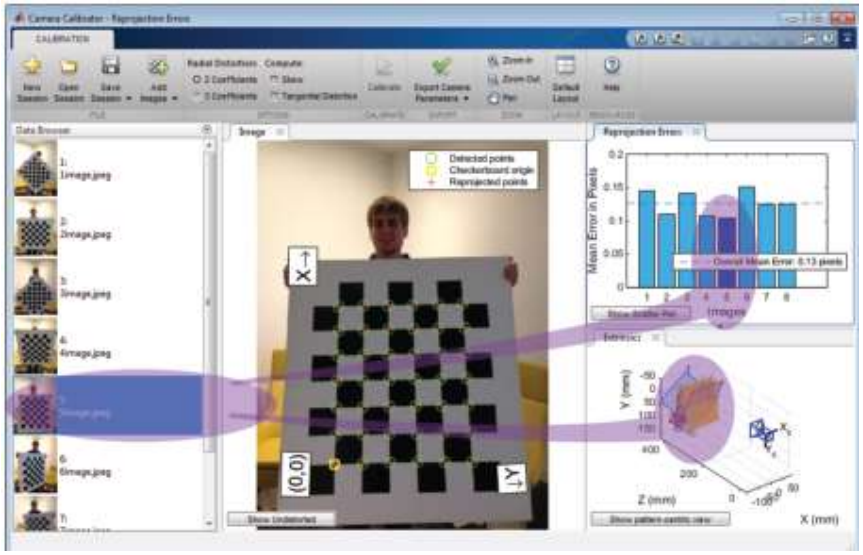
模式匹配



物体分割



相机校准APP



人脸识别与跟踪