

白皮书

在 FPGA 上部署 5G NR 无线通信：一套完整的 MATLAB 与 Simulink 工作流程

简介

算法创新推动无线通信技术进步, 从而促进个人连接、航天和卫星通信、高可靠性自动驾驶系统和物联网 (IoT) 的发展。设计、实现及测试这些系统需要跨多个学科密切合作。

将算法模型部署到 FPGA 硬件可以进行无线测试和验证。直接从系统级算法和模型自动生成 HDL 代码, 无需工程师依赖规范文档或手动构建和编写代码。

本白皮书介绍了将 MATLAB® 算法和 Simulink® 模型直接转换为适用于 FPGA 的 HDL 的工作流程, 内容包括:

- 使用 MATLAB 和 5G Toolbox™ 进行符合 5G NR 标准的算法建模
- 从基于帧的 MATLAB 算法过渡到数据流的 Simulink 实现
- 基于 Fixed-Point Designer™ 和目标硬件特性的定点实现
- 通过使用经过验证的知识产权 (IP) 加速设计
- 生成 HDL 并在目标硬件上部署, 在本例中为 Xilinx® Zynq® 设备

我们将通过一个 5G NR 小区搜索设计来说明该过程。MathWorks 创建此设计的初衷是为了满足关键客户需求。该设计现在已经成为 Wireless HDL Toolbox™ 提供的参考应用。此工作流程中还将使用 5G Toolbox 和 HDL Coder™。

在硬件中实现无线通信算法

通信工程师可以使用 MATLAB 和 Simulink 以及附加工具箱, 在算法级别开发、仿真和优化其应用。但是, 为了将这些算法实现到原型设计或生产部署需要跨多种角色进行协调, 这极大地增加了项目的复杂性。

不同的技能和工作环境

图 1 显示了在硬件中构建无线通信应用所需的技能。工程师通常只在这些领域中的一个或两个领域拥有专业知识, 因此项目通常涉及跨角色和跨部门协调。

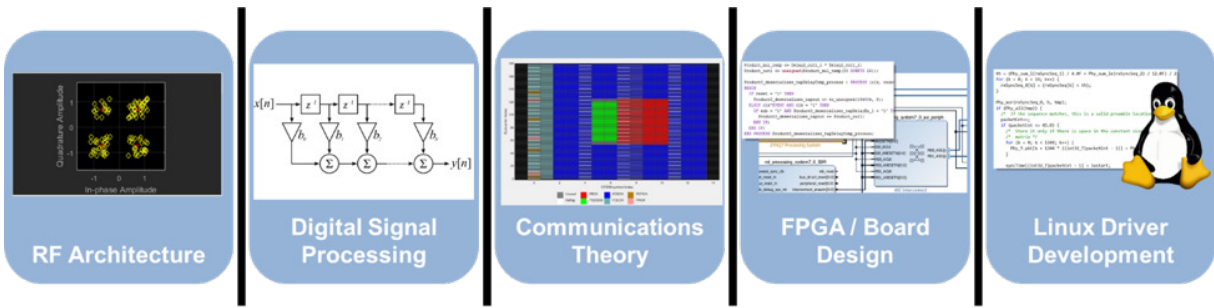


图 1. 在硬件上设计无线通信应用原型或部署所需的技能和工作环境。

硬件设计大不一样

对 FPGA 或 ASIC 器件上的硬件进行编程，与编写将被编译为在通用处理器上运行的一组指令的软件差别很大。数字硬件是固定资源的预构建电路，可在数据流上运行。虽然硬件可以提高速度，但需要一些专业知识来进行构建。其中的一些大致差异有：

- 数据流：您可以将任何大小或维数的数据加载到软件中。但是，硬件是流经 1 和 0 的电路。这意味着必须调整算法以适应长时间的数据流处理，并包括一些管理数据流的逻辑。
- 并行：硬件电路可以并行处理数据流，从而加快处理速度，但需要协调下行操作的并行路径时序，以确保数据如期到达。
- 定点数据类型：浮点数学通常比定点需要的运算更多，数据字长更宽。由于硬件是一组固定的资源，并且大多数团队更愿意使用最小的芯片，因此，定点广泛用于硬件设计。但是，即使对于有经验的硬件设计人员来说，从浮点算法转换为定点硬件就绪实现的过程也颇具挑战性。
- 管理资源使用和时钟周期：通常，初次尝试以 FPGA 为目标会导致这样的结果：要么不适合设备的资源，要么运行速度慢于预期。了解如何优化这两个正交参数正是硬件设计学习道路上的一大难关。
- 与软件和其他硬件器件交互：设备上片上系统 (SoC) 将至少和一个 FPGA 处理器相结合。硬件和软件通过具有预定地址的内存位置来回传送数据。访问这些寄存器以及其他外部内存和芯片 I/O，需要了解器件的架构。

成功的 FPGA 硬件定位需要根据上面列出的参数调整算法以有效运算。通信/DSP 工程师与其硬件设计方密切合作可获得最佳效果。只通过规范文档和综合报告进行沟通的工程师无法在 FPGA 硬件上有效地实现创新算法；这需要密切的合作和共同的设计探索环境。

工作流程：打通算法与硬件设计鸿沟的桥梁

适合工作的工具

MATLAB 适用于开发数学算法、操纵大型数据集、探索数学和可视化数据。但硬件实现需要时序、并行和自定义定点字长。这些构造原生于专用硬件描述语言 (HDL), 如 VHDL[®] 和 Verilog[®], 但这些 HDL 与 MATLAB 没有任何关联。这要求算法设计人员在书面规范中描述其意图, 由设计人员在编写详细的 HDL 和验证测试平台时进行解释。

Simulink 是一个视觉设计和仿真环境, 与 MATLAB 紧密结合, 并将算法设计桥接到硬件实现。通过内置时序, 您可利用 Simulink 设计和仿真并行架构, 并在设计过程中实现数据类型的传播可视化。Simulink 还可以对嵌入式软件和模拟电路进行建模, 从而在硬件中实现之前, 对整个系统执行总体仿真。做好实现准备后, 您可以使用 HDL Coder 和 Embedded Coder[®] 生成可合成的 HDL 和嵌入式 C 代码, 以部署在目标设备上。

出于这个原因, 许多团队使用 Simulink 作为算法与硬件设计之间的协作环境, 如图 2 所示。

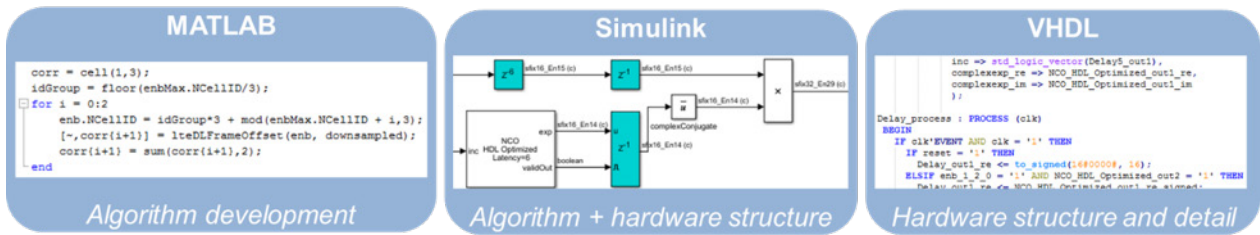


图 2. 算法和硬件设计环境。算法开发人员使用 MATLAB; 硬件设计人员使用 VHDL 或 Verilog。Simulink 缩短了这一差距, 使这两个角色能够实现更高质量的合作。

设计概述

为了说明此工作流程, 我们将使用 5G NR 无线设计为例, 但所讨论的规则适用于任何无线设计。

NR HDL 小区搜索设计可检测及解调 5G NR 同步信号块 (SSB)。用户设备 (UE) 使用主同步信号和辅助同步信号, 从 gNodeB (gNB) 基站获取小区标识和帧时序。

图 3 显示了设计架构的大体结构图。数字下变频器 (DDC) 模块先修正频率偏移, 再由 PSS 模块返回最强的主同步信号 (PSS)。使用最强 PSS 执行正交频分复用 (OFDM) 解调。最后, 在相应的资源单元中搜索辅助同步信号 (SSS)。

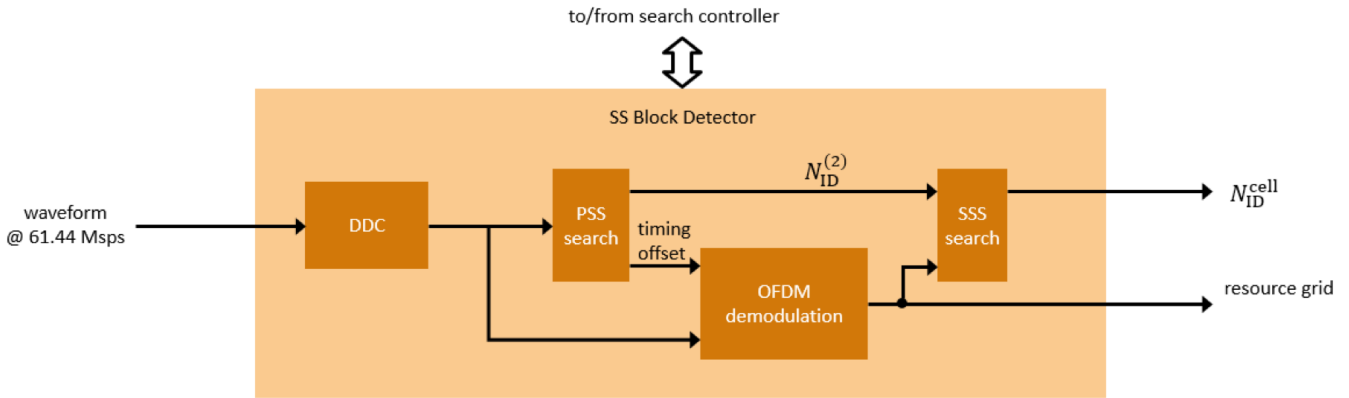


图 3. NR HDL 小区搜索硬件架构的大体结构图。

在将这些算法用于流硬件时做出了某些考虑和权衡, 其中部分将在下面的工作流程组件描述中做重点介绍。

工作流程概述

从 MATLAB 算法到 FPGA 有很多途径。图 4 所示的自上而下优化方法可帮助团队以明确、敏捷的方式将其无线设计部署到 FPGA。

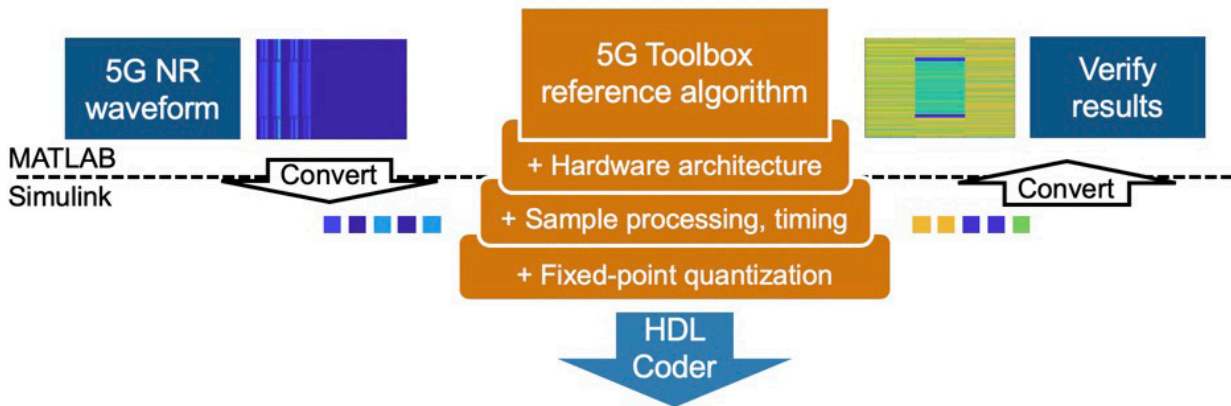


图 4. 自上而下从无线算法到 FPGA 部署优化工作流程。

这些优化步骤无需遵循图中所示的顺序；优化排序因应用和您的个人偏好而异。使用此方法可在与 MATLAB 保持连接的同时逐步过渡到 Simulink。鉴于 Simulink 采用内置时间概念，因此添加基于采样的流处理不失为在 Simulink 中实施优化的好方法。应根据前一阶段或原始算法结果验证这些优化步骤中的每一步。

图 4 中的蓝色框表示测试平台单元：生成输入波形并验证结果。其中包含多个层，视工作流程阶段而定。在持续优化设计的过程中，测试平台将使用前一阶段作为参考进行比较，因此前一阶段成为测试平台的一部分。务必充分验证每个阶段，以便在出现缺陷时识别并修复缺陷，进而使用已知良好的版本验证下游优化。

参考算法

无线通信设计通常始于在 MATLAB 中进行算法开发和测试。这个 MATLAB 算法便成为“黄金参考”，其余的过程以此为参照继续进行。在传统的工作流程中，算法将作为功能规范文档的来源。在此工作流程中，算法将既是进一步优化的起点，也是验证下游优化的可执行模型。

5G NR 无线标准复杂深奥，但您可以通过使用 5G Toolbox 中符合标准的预先构建功能来加快算法开发过程。这些算法与 Communications Toolbox™ 的无线功能相结合，可让您快速构建已知良好的 5G 通信模块，从而专注于开发独特的算法。

NR HDL 小区搜索设计采用 PSS 搜索、OFDM 解调和 SSS 搜索步骤（如随 5G Toolbox 提供的 [NR 同步程序](#) 示例所示）。

硬件架构

要设计实现算法，首先要进行分区。在 5G Toolbox 示例 [NR 同步程序](#) 中，用于波形生成、信道传播、接收器和检查结果的 MATLAB 代码均合并到一个脚本。要实现接收器（如图 4 所示），需要对接收器功能进行分区并定义输入和输出。

Wireless HDL Toolbox 参考应用包含 MATLAB 版本的分区接收器：[NR HDL Cell Search MATLAB Reference](#)。在此版本中，执行高速信号处理任务的接收器功能被划分到各自的 MATLAB 函数，进而部署到具有输入和输出的硬件，如图 5 中的橙色框所示。图 5 中的蓝色框将成为硬件测试平台。搜索控制功能可作为软件实现，用于片上系统 (SoC) 设备上的处理器。在此参考应用中，该功能将作为测试平台的一部分在 MATLAB 中实现。此类硬件-软件分区（通过软件控制硬件加速）是针对无线应用场合的一种常见方法。

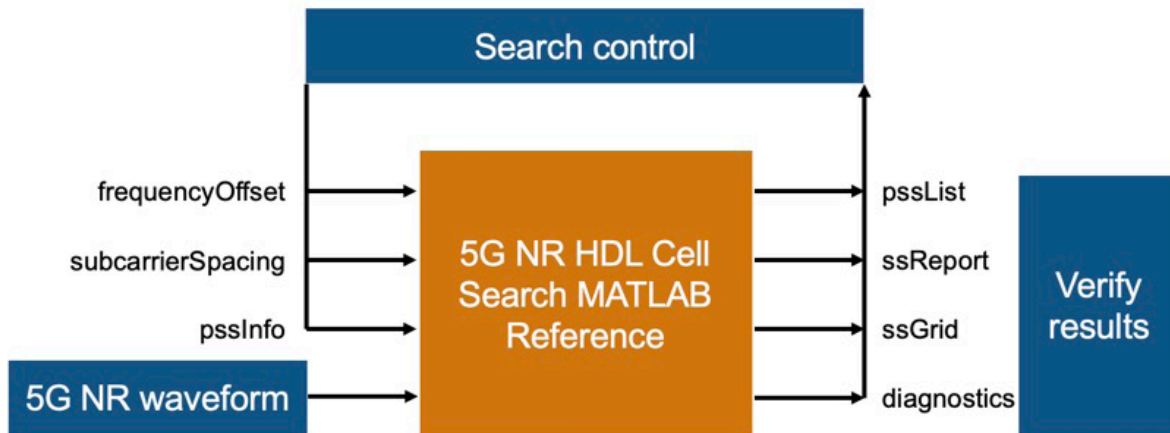


图 5. 通过软件和测试平台对硬件功能进行分区。

设计硬件时，首先要修正从搜索控制传递的频率偏移并降低采样率。这样，软件可以搜索一系列粗频偏。硬件执行 PSS 关联并测量残差细频偏，将结果传回搜索控制，进而确定最强峰值。接着，将此信息及细频偏值传回硬件，在其中执行 OFDM 解调和 SSS 检测。

截至目前为止，所有处理均针对整个波形开展，即“基于帧”的处理。设计硬件时应着眼于真实模式实现场景，此时信号会源源不断地流入。接下来，调整算法以适应样本流处理。

基于采样的处理和时序

调整算法以使用流样本数据时，需要思维方式上进行一些大的转变。您不仅需要更改算法的工作原理，还需要管理数据流。这是您需要做出的最大改变，但却是针对硬件的必要改变。由于采用内置采样时间概念，使用 Simulink 设计这种工作模式往往更方便。

在 NR HDL 小区搜索设计中，检测 PSS 需要将已知序列与输入信号相关联。在纯数学算法中，相关性十分直观：将一个信号滑过另一个信号，对样本的结果求和，并在总和中确定足够的峰值。

在 PSS 检测的情况下，三个小区标识 PSS 中的每一个都需要与输入信号互相关。PSS 是可能存储在本地片上 RAM 中的静态值，并且输入信号是流入的，因此一个信号在另一个信号上的滑动自然发生。但是，我们仍然需要缓冲输入流的窗口以执行互相关计算。这些全部在 PSS 相关器的离散 FIR 滤波器 HDL 优化模块中处理，如图 6 所示。这些模块可以自由选择采用串行架构还是并行架构，并自动根据参数设置更新仿真延迟。

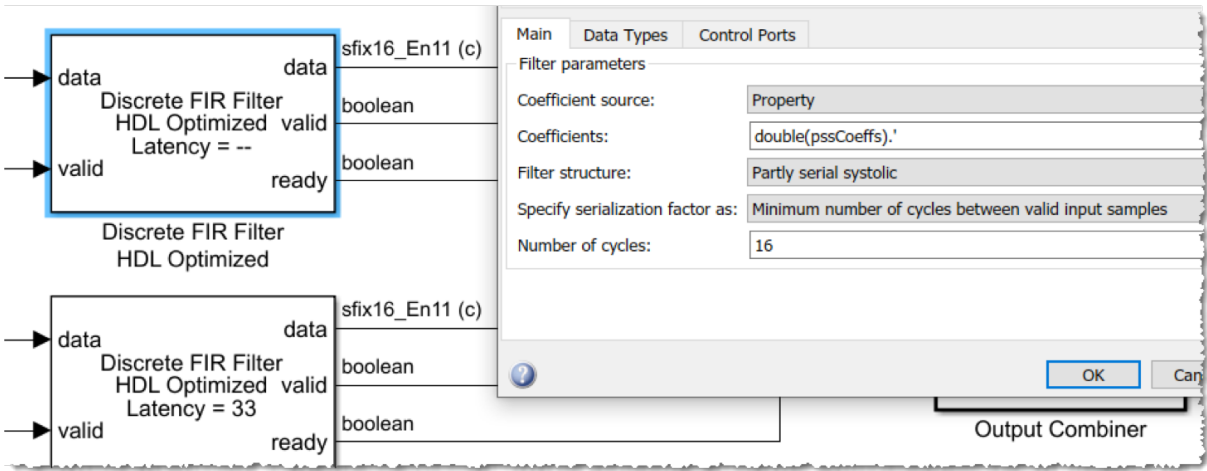


图 6. 配置离散 FIR 滤波器 HDL 优化模块。

在硬件中，我们可以并行执行三个 PSS 相关，检测峰值，然后将结果传递到控制软件。为确定峰值，我们会检查各相关结果是否高于平均信号功率对应的阈值。如果超过阈值，我们可以在本地时间窗口中搜索真正的峰值。图 7 说明了这种架构，其中绘制了 PSS 相关结果和平均信号功率。

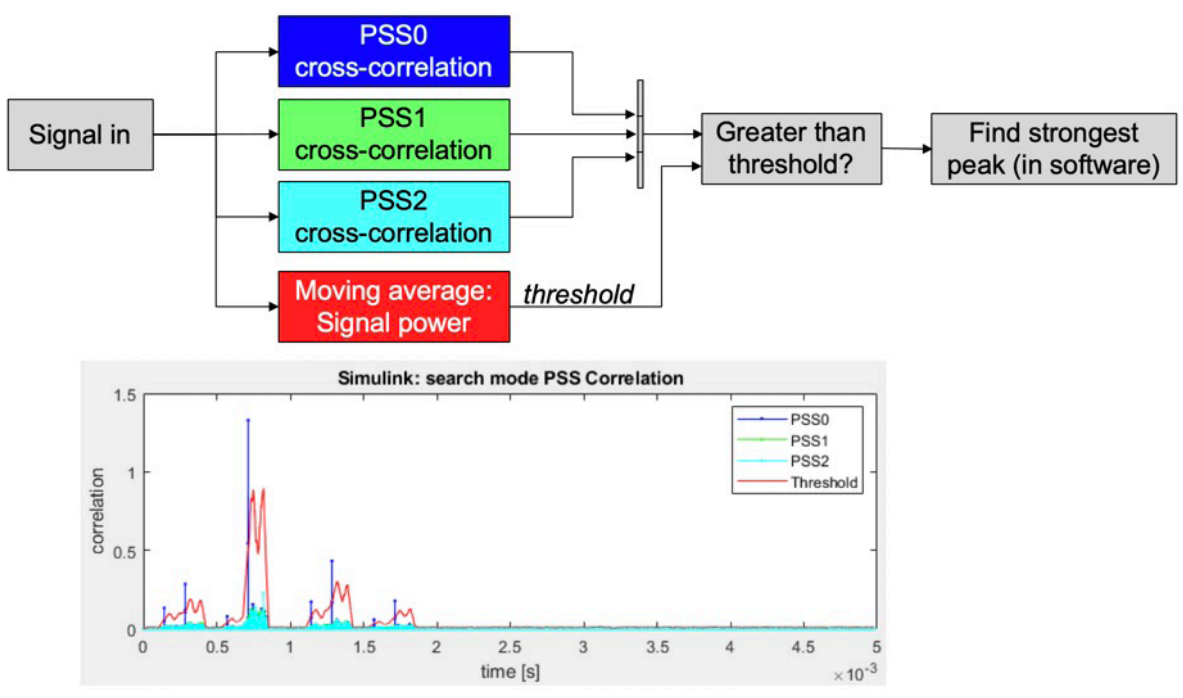


图 7. PSS 检测流程图 (上图)、PSS 互相关结果图和移动平均阈值 (下图)。

计算连续数据流的平均信号功率所需采用的方法，有别于计算 MATLAB 中的全局平均值所采用的方法。这也需要滑动窗口方法，也是图 7 中的阈值图随时间变化的原因。测量平均信号功率的时间窗口与相关器相同。此方法不仅可以提供所需的检测性能，而且占用的存储空间较小。这种设计折中需要算法开发人员与硬件工程师密切合作。

图 8 显示了移动平均值算法。延迟模块（如“ z^{-1} ”）将映射到硬件中的寄存器以存储值。“ z^{-128} ”模块将 FIFO 缓冲区与输入采样并行插入，允许减去之前 129 个时钟周期的采样，以便持续存储最近的 128 个移动平均值样本。图 8 中的蓝色寄存器用作工作流中的一级。由于数学运算会耗费硬件时间，因此将它们分解以减少各阶段之间的计算量会缩短时钟周期，从而增加最大时钟频率。您可以让 HDL Coder 自动插入工作流；但是，最好在设计中对正确数量的延迟进行建模，以便在模拟过程中确保并行路径在生成 HDL 之前正确同步。请注意图 8 中并行 validIn 信号与延迟的匹配方式，这样当该信号用于启用的延迟时，将可到达正确的时钟周期。如果让 HDL Coder 将流水线延迟作为优化（如自适应流水线）的一部分，请确保在“优化”设置中启用“平衡延迟”（默认设置）。

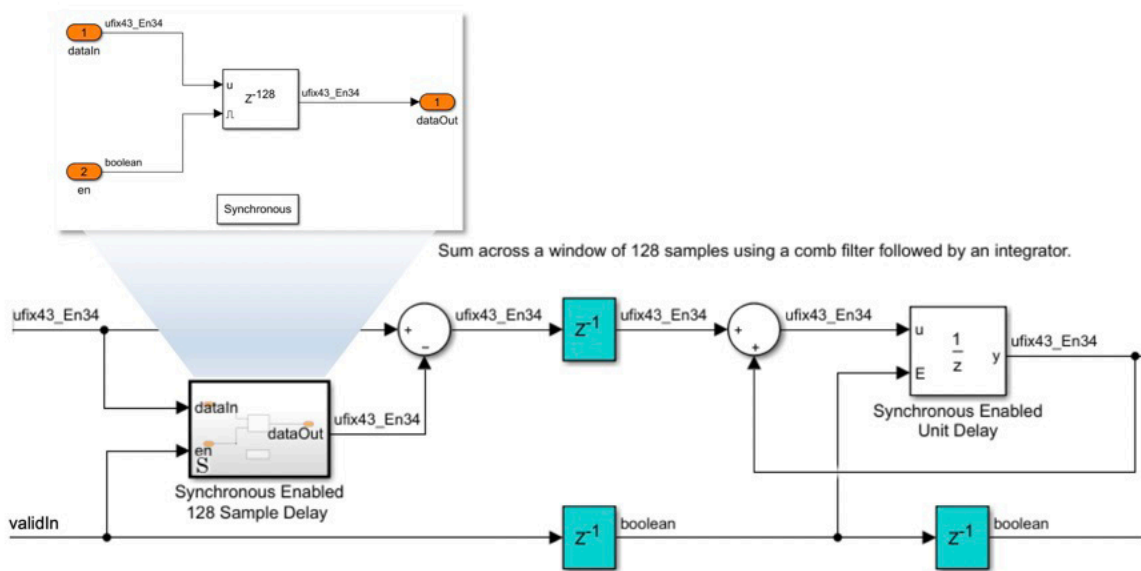
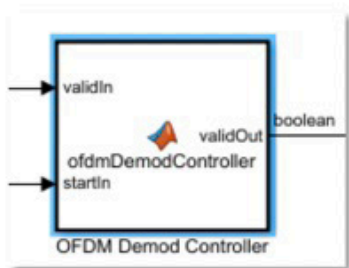


图 8. Simulink 中的移动平均算法，展示了硬件实现细节。

随着数据连续流动，算法需要知道诸如分组之类的序列何时开始和结束，或者是否被中断。这就要求硬件中的控制信号至少由“valid”信号组成，但通常包括“enable”、“start”和“end”。在 NR HDL 小区搜索设计中，OFDM 解调器使用 MATLAB 函数模块根据 startIn 和 validIn 信号生成有效信号，如图 9 所示。



```

Editor - Block: nrhdSSBlockDetection/SS Block Core/OFDM Demodulate/OFDM Demod Controller
SS Block Core/OFDM Demodulate/OFDM Demod Controller
function validOut = ofdmDemodController(validIn, startIn)
1
2
3     persistent req
4
5     ofdmCountWL = nextpow2(274*3);
6
7     if isempty(req)
8         req = struct(...
9             'valid', false, ...
10            'started', false, ...
11            'ofdmCount', fi(0,0,ofdmCountWL,0)...
12            );
13     end
14
15     validOut = req.valid;
16
17     next = req;
18
19     if startIn && validIn
20         next.valid(:) = 1;
21         next.started(:) = 1;
22         next.ofdmCount(:) = 0;
23     elseif validIn && req.started
24         next.valid(:) = 1;
25         if req.ofdmCount == 274*3-2
26             next.started(:) = 0;
27             next.ofdmCount(:) = 0;
28         else
29             next.ofdmCount(:) = req.ofdmCount + 1;
30         end
31     else
32         next.valid(:) = 0;
33     end
34
35     req = next;
36
37 end

```

图 9. OFDM 解调控制器 MATLAB 函数模块和相应的 MATLAB 代码。

前面描述的流式算法使用 FIFO 缓冲区（一种简单形式的内存）来临时存储来自数据流的窗口。非顺序存储或使用的数据必须使用可寻址的 RAM 资源或只读 LUT 资源。内存架构可以极大地影响硬件实现的性能和资源使用。图 10 显示了如何在 SSS 检测子系统中使用内存。MATLAB 函数模块控制状态：“闲置”；将 OFDM 解调符号写入 RAM；读取信号建立关联与 SSS 序列。SSS 序列是固定值，因而存储在 LUT 中，作为一种只读内存资源。

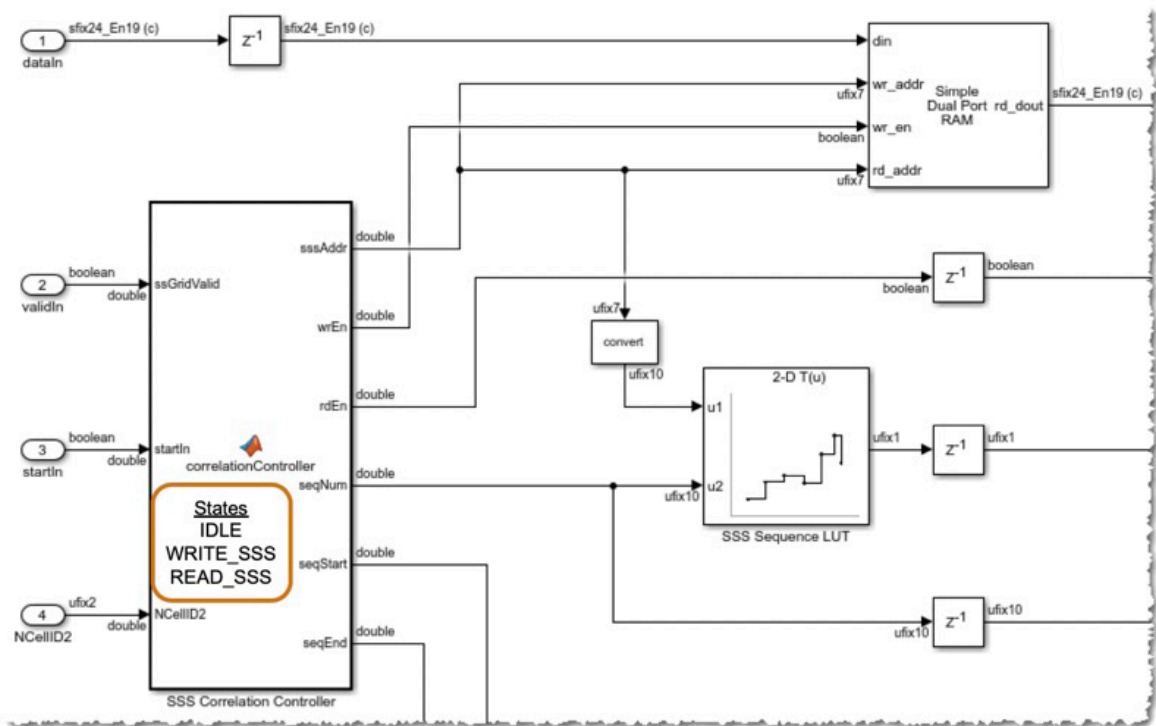


图 10. 使用 RAM 和 LUT 硬件资源作为 SSS 检测算法输入。

为了实现性能和资源使用目标，了解内存大小和硬件设备功能非常重要。在本例中，RAM 是双端口，意味着可以同时执行读写操作。地址行是 7 位，因而可以将数据存储在 128 个位置；数据位宽是 48 位，因而需要 6144 位 RAM，在大多数情况下将映射到设备模块 RAM。

当您准备对 Simulink 硬件实现进行仿真时，Simulink 将与 MATLAB 工作区进行交互。NR HDL 小区搜索设计的顶层使用 From Workspace (自工作区) 模块在 MATLAB 工作区结构“in”中获取基于帧的信号数据，并逐个样本地传输到硬件实现中。在 To Workspace (到工作区) 模块中收集输出，然后将输出写入结构“out”，供 MATLAB 使用。图 11 显示了 Simulink 图，其中部分 MATLAB 代码用于设置输入参数、在搜索模式下运行 Simulink 模型、读取结果，然后在解调模式下运行 Simulink 模型。但还有一点图 11 并未体现：还可以调用 MATLAB 参考算法验证数据流 Simulink 实现的结果。

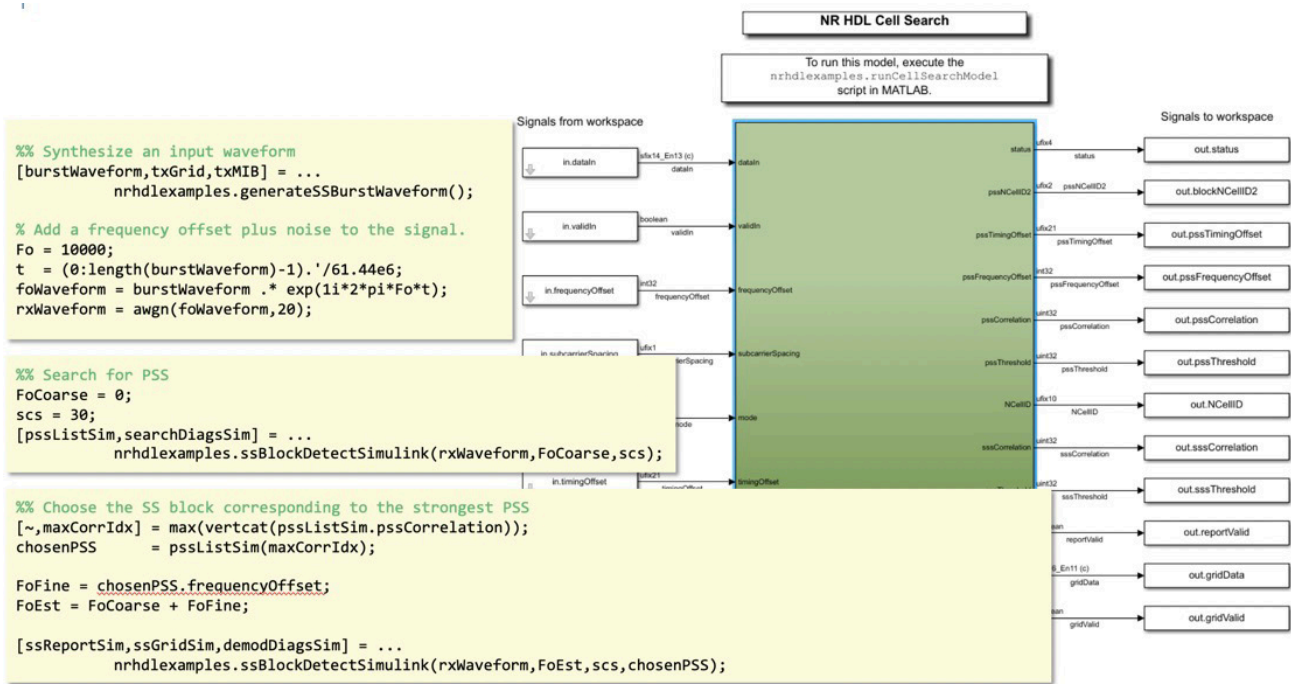


图 11. 用于 NR HDL 小区搜索参考应用的顶层 Simulink 图，其中 MATLAB 代码用于为输入合成符合 5G NR 标准的波形、在 PSS 搜索模式下从 MATLAB 调用 Simulink，以及将最强 PSS 传回 Simulink 模型以执行 SSS 检测。

利用经过验证的 IP 模块

实现流式算法的硬件架构时，相比承担从头开始设计的成本和风险，重复使用预构建模块通常更为现实。这些 IP 模块可以来自外部来源或公司内的其他项目。无论来源如何，都需遵守三个关键的使用原则：

1. **使用内置的可定制性而不是更改设计本身。** 这种可定制性通常采用参数的形式。例如，鉴于 PSS 相关器中采用离散 FIR 滤波器 HDL 优化模块（如图 6 所示），您可以重用 MATLAB 工作区系数、自定义 HDL 滤波器架构、实现操作串行化并配置其他参数。设置参数并构建设计之后，该模块将对此硬件实现进行仿真，包括特定设置延迟。此实现将经过优化并作为 HDL 下游生成。
2. **使用易于插入设计的 IP 模块。** NR HDL 小区搜索设计的 SS 模块内核子系统中的 OFDM 解调器模块说明了这一原则，如图 12 所示。此模块支持解调各种基于 OFDM 的标准：5G、LTE、无线局域网 (WLAN 802.11a/b/g/n/ac)、WiMAX (802.16 m 和 e)、数字视频广播 (DVB) 和数字音频广播 (DAB)。您可以根据自身设计需求快速设置循环前缀、OFDM 符号保护间隔和 FFT 实现参数。请注意，输入控制功能使用 startIn 信号确定何时重置解调器及驱动有效信号将样本传递到 OFDM 解调器。
3. **选择已经证明可以提供所需功能和性能的 IP 模块。** 这似乎是显而易见的，但如果您不遵守这一原则，您遇到的任何问题都将难以解决，因为您不熟悉设计。这就是 **NR HDL 小区搜索设计** 针对 Xilinx Zynq 硬件并使用无线信号现场测试的原因。

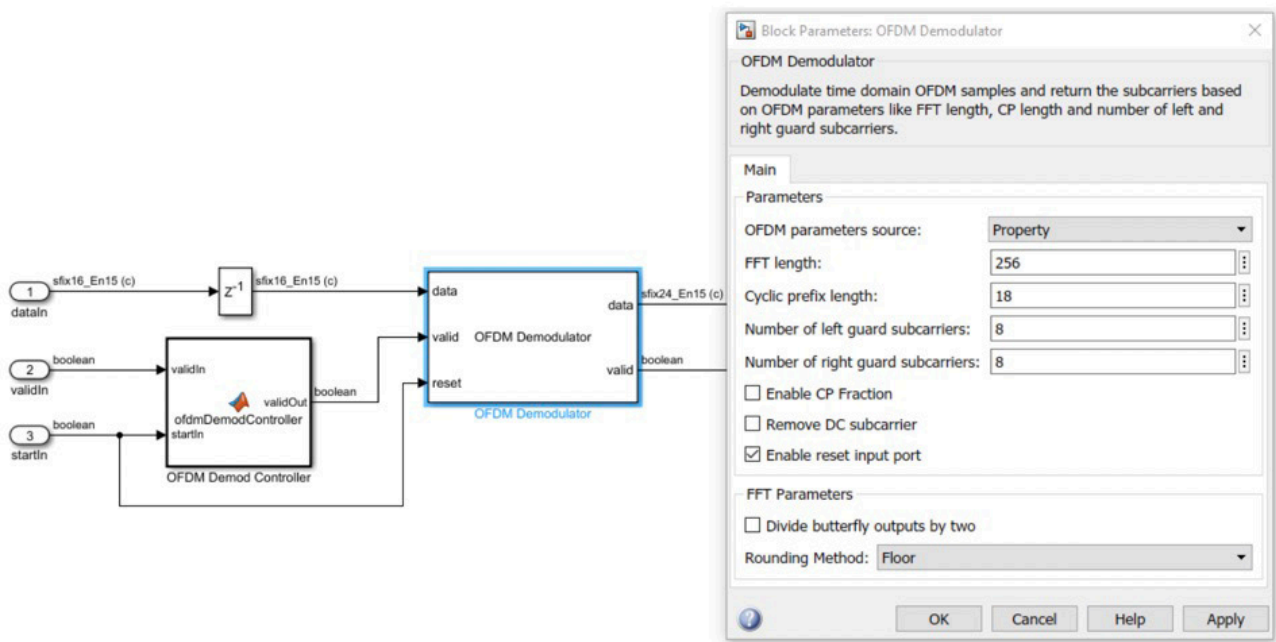


图 12. 插入设计环境的 IP 模块。

定点实现

大多数数字硬件设计使用定点数学来提升实现效率。准确度与效率恐难两全，存在反馈回路或高动态范围方程时尤为如此。HDL Coder 提供了原生浮点实现，可用于 LTE 中 OFDM 均衡所需的除法运算。您可以通过数据类型转换模块隔离这些运算，并生成单精度浮点运算，以实现可伸缩的准确度。

对于大多数设计，包括 5G NR 小区搜索设计在内，都可以做到足够准确的定点实现，并且您可以使用 Fixed-Point Designer 来管理该过程。有助于对定点理论有一个基本的了解，尤其是字长在不同的算术运算中如何增长。例如，乘积的字长将是输入字长的总和，您可以在 Simulink 中观察到，如图 13 所示。在所示的表示法中，sfix 表示带符号的定点，第一个数字是字长，第二个数字是精度位。因此，sfix16_En11 是一个带符号的定点数，其字长为 16，其中 11 位是精确位。当这两个 16 位数相乘时，会得出一个 32 位乘积。

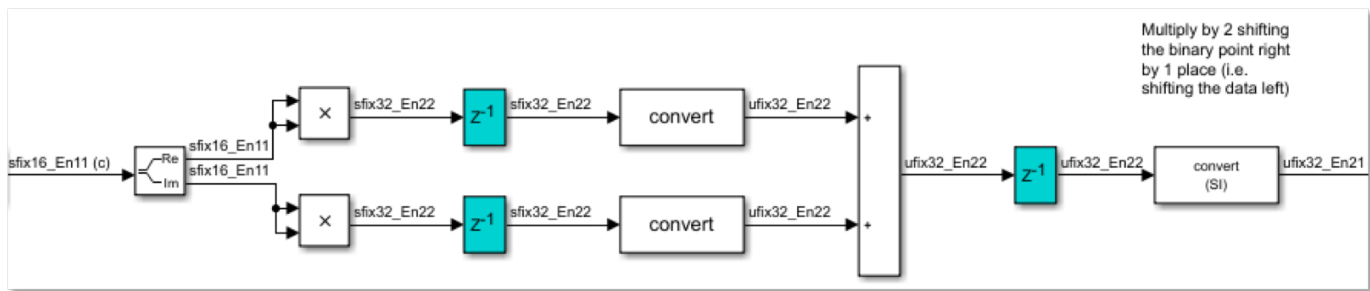


图 13. 通过算术运算进行定点字长传播。

乘数是执行定点转换时的关键因素，因为乘法运算需要占用大量硬件资源。大多数 FPGA 具有称为 DSP 切片的高效乘法资源，通常为 18x18 或 25x18。如果您的输入达到且大于乘数值，则乘法运算将在 DSP 切片中分割，使这些切片迅速耗尽。这又会造成效率与准确度难以两全的情况。

部署到 FPGA 硬件上

在您对定点实现的准确度感到满意之后，便可通过 HDL Coder 对过程进行定位。首先右键单击 SS Block Detection 子系统，然后选择 **HDL 代码 > HDL Workflow Advisor**。这个类似向导的界面将引导您完成生成 VHDL 或 Verilog HDL 以定位器件的步骤。HDL Coder 生成与器件无关的代码，其中包含与 Simulink 设计相对应的层次结构和信号名称，以及所有位级端口映射，如图 14 中的代码片段所示。

```

-----
--
-- Module: nrhd1SSBlockDetection_Magnitude_Squared_2
-- Source Path: nrhd1SSBlockDetection/SS Block Core/PSS Detection/Correlators/Magnitude Squared 2
-- Hierarchy Level: 5
--
-----
LIBRARY IEEE;
USE IEEE.std_logic_1164.ALL;
USE IEEE.numeric_std.ALL;

ENTITY nrhd1SSBlockDetection_Magnitude_Squared_2 IS
  PORT(
    clk          : IN    std_logic;
    reset        : IN    std_logic;
    enb          : IN    std_logic;
    dataIn_re    : IN    std_logic_vector(15 DOWNTO 0); -- sfix16_En11
    dataIn_im    : IN    std_logic_vector(15 DOWNTO 0); -- sfix16_En11
    validIn      : IN    std_logic;
    dataOut      : OUT   std_logic_vector(31 DOWNTO 0); -- ufix32_En24
    validOut     : OUT   std_logic
  );
END nrhd1SSBlockDetection_Magnitude_Squared_2;

ARCHITECTURE rtl OF nrhd1SSBlockDetection_Magnitude_Squared_2 IS
  -- Signals
  SIGNAL dataIn_re_signed      : signed(15 DOWNTO 0); -- sfix16_En11
  SIGNAL dataIn_im_signed      : signed(15 DOWNTO 0); -- sfix16_En11
  SIGNAL Delay3_out1_re        : signed(15 DOWNTO 0); -- sfix16_En11
  SIGNAL Delay3_out1_im        : signed(15 DOWNTO 0); -- sfix16_En11
  SIGNAL Product4_out1         : signed(31 DOWNTO 0); -- sfix32_En22
  SIGNAL Product5_out1         : signed(31 DOWNTO 0); -- sfix32_En22
  SIGNAL Delay1_out1           : signed(31 DOWNTO 0); -- sfix32_En22

```

图 14. 用于幅度平方子系统的生成 VHDL 片段。

HDL Coder 还可以驱动 FPGA 合成和部署过程,当然,这项指标针对您的供应商和器件。由于有太多可用于部署的器件和电路板,因此本白皮书无法涵盖所有详细信息,仅介绍常规定位类别:

部署原型板。如果希望使用现成的电路板,例如基于 Xilinx Zynq 的软件定义无线电 (SDR) 工具包,可下载[硬件支持包](#),以简化硬件和在 ARM® 处理器上运行的软件的设置及部署。[本示例](#)使用这项 NR HDL 小区搜索设计来说明针对基于 Xilinx Zynq 的电路板的步骤。

部署生产器件。通常,您可以与硬件团队一起制定自定义电路板定义。该定义指定电路板上的器件、电路板上的外设以及器件引脚如何映射到板级 I/O。运行 HDL Workflow Advisor 时,可以在步骤 1.1 中选择此电路板作为目标。然后,您可以将设计的 I/O 映射指定到任一寄存器,以便与软件或器件的引脚进行通信。

验证和调试

如“工作流程概述”部分所示,您的测试平台将随设计而演变。以下简要介绍测试平台的主要单元及其可能的演变过程。

激励

算法开发早期,您可以使用基于 MATLAB 的功能合成波形。在 NR HDL 小区搜索设计中,测试平台使用 5G Toolbox 功能合成符合标准的波形并通过噪声信道模型前向传输。无论激励波形的建模程度如何,执行现场测试时总会遇到不可预料的信号单元。因此,尽量捕捉实时无线波形是不错的选择(如针对 LTE 信号的[本示例](#)所示),以便先在 MATLAB 和 Simulink 中执行测试再延伸到硬件部分,硬件调试十分困难。

参考模型

优化设计的硬件细节、实现基于采样的流处理及定点量化时,您可以将结果与此前验证的算法版本进行比较。您可以随时将上述结果与原始参考算法结果进行比较,但比较更多类似设计版本的结果往往更加直观。一个很好的例子就是,一旦转换为流处理,比较流输出比转换回帧容易得多。在 NR HDL 小区搜索示例中,我们对 Simulink 实现模型与 MATLAB 硬件架构模型进行了比较,后者是前者的优化版本。

评估和调试

开发原始算法时,正确性评估通常包括检查波形、范围和频谱图。但是,当您进一步优化算法时,在验证过程中需衡量添加的实现优化是否与原始算法的功能相匹配。不同优化结果相互之间通常并不完全匹配,比较浮点版本与定点版本就是一个很好的例子。因此,您将需要确定如何评估算法是否通过了各项测试。在 NR HDL 小区搜索示例中,通过衡量相对均方误差比较 MATLAB 模型和 Simulink 模型的结果。误码率是另一种常用的无线通信技术。就及格/不及格衡量结果达成共识属于项目级决策。

在引入缺陷的步骤中, 往往更便于调试测试故障。如果一味等待后期发现缺陷, 所要面临的挑战将成倍增加, 因为您将需要遍历更多细节, 还要从设计决策中删除可能引发缺陷的步骤。一旦在 FPGA 上运行设计, 内部信号可视性将受到局限, 复杂度将进一步提升。

在继续运行并生成 HDL 或将其编入 FPGA 之前, 衡量**覆盖率**是评估设计功能测试情况的一种方式。同时, 这也有助于您评估测试是否在反复运行同一项功能, 从而浪费了仿真时间。您可以根据覆盖率分析反馈调整运行的测试。

减轻下游验证压力

在低层次细节层面进行验证和调试可能非常耗时且令人头痛, 因此尽早识别并修复缺陷至关重要。从寄存器传输级别 (RTL) 到设备实现, 增加的细节和功能越来越多, 因此引入缺陷的概率也越来越大。您可以充分利用在 MATLAB 和 Simulink 中的工作成果, 提高解决这些问题的效率。运用 **HDL 协同仿真**和 **FPGA 数据采集**等技术, 您不仅可以在 MATLAB 和 Simulink 中进行调试, 还能直接通过测试平台模型**生成验证组件**, 帮助生产验证团队构建自己的测试平台。

结果

我们将 NR HDL 小区搜索设计部署到连接了 Analog Devices® AD9361 RF 卡的 Xilinx Zynq-7000 All Programmable SoC ZC706 评估工具包。该工具包扫描了一个频率范围, 并通过实时无线信号检测到 PSS 和 SSS, 发现最强小区信号。将 OFDM 解调同步信号块资源网格从 SoC 传递到 MATLAB, 在 MATLAB 中使用 5G Toolbox 成功解码。

也许更重要的是, 我们从 MATLAB 算法开始, 添加硬件实现细节进行优化, 从同一环境自动生成并定位此器件。这种方法使我们能够轻松验证每次对硬件的连续优化, 以确保它提供与 MATLAB 算法相同的结果。此类仿真采用 5G NR 波形, 同时使用实时无线信号检测 PSS 和 SSS。

一名工程师很少具备完成这项工作的所有技能; 此处显示的协作环境通过召集多种角色的工程师综合运用这些技能, 便能够更快地进行设计、仿真和迭代。在系统级环境中尽早借助真实的刺激进行仿真, 有助于工程师在详细的硬件设计开始之前找到并修复功能和性能问题。可快速实现更改并进行仿真, 然后自动生成更新的代码以定位硬件。这样的敏捷性使工程师能够将精力集中在对产出的产品有更大影响的高阶解决方案上, 不致于陷入编写数千行 VHDL 而无暇抽身。

无论您的最终目标是原型设计还是生产部署, 此工作流程都可以加快将高质量无线通信算法部署到 FPGA 硬件的速度。

了解更多

观看有关 NR HDL 小区搜索设计的视频 (Wireless HDL Toolbox 提供的参考应用) : [5G NR HDL 小区搜索参考应用](#)

下载 Zynq SDR 硬件支持包: [Communications Toolbox 的 Zynq SDR 支持](#)

访问资源中心: [使用 MATLAB 进行无线通信设计](#)

探索快速入门服务

MathWorks 提供了许多方法来帮助您有效地学习如何使用这个工作流程, 从应用工程师入门, 到深入的现场培训课程和定制的咨询服务。

- [MATLAB 和 Simulink 培训](#)
 - [使用 MATLAB 设计 LTE 和 LTE Advanced 物理层系统](#)
 - [适用于 FPGA 的 DSP](#)
 - [使用 Simulink 在 Zynq 上开发软件无线电](#)
- [MathWorks 咨询服务](#)