

Avances en el arte del diseño de sistemas: Flujos de diseño de última generación para los sistemas de comunicaciones y procesamiento de señales

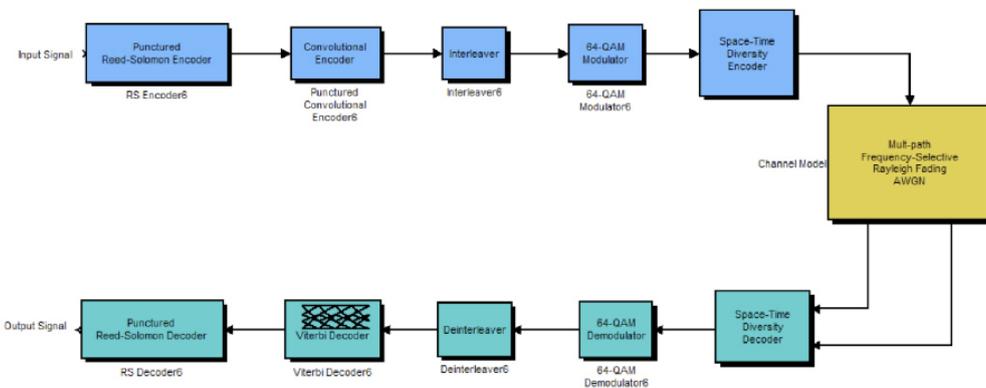
Artículo cedido por MATHWORKS



Autor Dr. Arun Mulpur,
Director de marketing
industrial para comu-
nicaciones, electróni-
ca y semiconductores
MathWorks

Las discontinuidades en el flujo de diseño resultan cada vez más perjudiciales y caras durante el desarrollo de tecnologías complejas de comunicaciones y procesamiento de señales. Al ser más cortos los ciclos de diseño, se magnifica el impacto de estas discontinuidades. En este artículo se describen algunas aplicaciones concretas (diseño de algoritmos, arquitectura de sistemas y diseño de hardware) que ilustran los recientes y significativos avances en cuanto a herramientas y métodos de modelización, simulación y generación de código. En el caso del diseño de algoritmos, ya hay disponibles en MATLAB varios cientos de objetos de sistema para el procesamiento de señales y las comunicaciones listas para usar que permiten un desarrollo más rápido de algoritmos complejos. Los arquitectos de sistemas pueden ahora modelizar componentes de sistemas de RF y banda base en un entorno unificado, así como llevar a cabo verdaderas simulaciones de frecuencia múltiple. En cuanto al diseño de hardware, diversos avances recientes, tales como el asistente de flujo de trabajo, el resaltado de rutas críticas, la canalización distribuida, la anotación de corrección y los informes de utilización de recursos, proporcionan un marco de trabajo integrado, para lograr iteraciones de diseño y prototipado más rápidas.

Figura 1. Diagrama de bloques que representa un sistema de comunicaciones típico



```

%% Load simulation parameters from a MAT file
clear
load mysystemobject.mat
%% Instantiate system objects to model the sequence of operations
% Transmitter
hRSEncoder = comm.RSEncoder(120,108);
hConvEncoder=comm.ConvolutionalEncoder(poly2trellis(7, [171 133]),...
'TerminationMethod','Truncated',...
'PuncturePatternSource','Property',...
'PuncturePattern', reshape([1 0 1 0 1;1 1 0 1 0]), 10, 1) );
hInterleaver= comm.BlockInterleaver('PermutationVector',...
interleave_pattern);
hModulator = comm.RectangularQAMModulator(64,...
'BitInput',true,...
'SymbolMapping','Custom',...
'CustomSymbolMapping',constellation_pattern,...
'NormalizationMethod','Average power');
hMIMOEncoder = comm.OSTBCEncoder;
%% Channel model
hAWGN=comm.AWGNChannel('NoiseMethod','Signal to noise ratio (Eb/No)',...
'EbNo',10);
  
```

Figura 2. Sistema de comunicaciones modelizado en MATLAB mediante los objetos de sistema de comunicaciones y procesamiento de señales

Diseño de algoritmos para sistemas de streaming

Muchos ingenieros inician el desarrollo de algoritmos de comunicaciones y procesamiento de señales en MATLAB mediante la aritmética en punto flotante. Estos desarrolladores de algoritmos pueden sacar partido de las sólidas capacidades de adquisición y análisis de señales de MATLAB, así como de las bibliotecas de algoritmos integradas de diversas toolboxes. Sin embargo, en algunas

organizaciones, estos algoritmos se reescriben después en código C a fin de ajustarlos para la implementación, la conversión a la aritmética de enteros o en punto fijo, o bien la integración con otros elementos de diseño. Este paso de reescritura es el ejemplo perfecto de una discontinuidad potencialmente costosa y perjudicial en el flujo de diseño.

Los objetos de sistema, una importante incorporación reciente en MATLAB, permiten a los ingenieros diseñar modelos de sistemas de procesamiento de señales para aplicaciones de streaming directamente en MATLAB, además de utilizar varios cientos de nuevos componentes de biblioteca para el procesamiento de señales, el procesamiento de imagen y vídeo, y las comunicaciones.

A modo de ejemplo, la Figura 1 muestra un diagrama de bloques que representa un sistema de comunicaciones básico con un transmisor, un canal y un receptor como componentes. Con el fin de modelizar y simular un sistema como este, algunos ingenieros escriben muchos miles de líneas de código C y, después, buscan

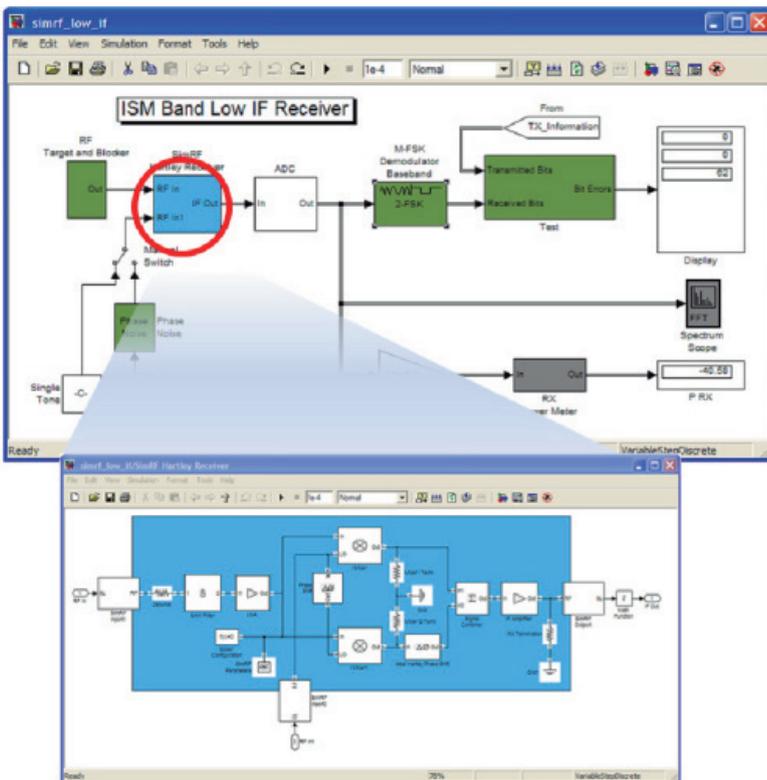


Figura 3. Receptor de FI baja de la banda ISM con subsistemas digital y RF en un modelo unificado (arriba) y detalle del subsistema RF del receptor Hartley modelizado en SimRF (abajo)

formas de integrar el diseño con el equipo de pruebas o de analizar los resultados de la simulación.

Por el contrario, en relación con los varios miles de líneas de código C que se suelen escribir, el código de MATLAB mostrado en la Figura 2 emplea diversos objetos de sistema disponibles en DSP System Toolbox and Communications System Toolbox. Por ejemplo, para modelizar el transmisor, un ingeniero puede crear una instancia de los objetos de sistema Codificador Reed-Solomon, Codificador convolucional, Entrelazador de bloques, Modulador QAM rectangular y Codificador de bloques espacio-tiempo ortogonales de Communications Blockset y llamarlos uno tras otro, tal y como se muestra en la Figura 2.

Asimismo, al contrario que en los estilos tradicionales de programación funcional, los objetos de sistema de MATLAB son implementaciones de algoritmos orientadas a objetos; gestionan de forma implícita la indización, el almacenamiento en buffer y la administración de estado, lo cual hace que el código resulte mucho más fácil

de escribir, depurar y mantener. La estructura del código permite a los ingenieros compararlo con facilidad con la especificación original o el diagrama de bloques. Los diseñadores de algoritmos pueden combinar este código de forma rápida con su código de MATLAB existente y probar los algoritmos con datos dinámicos adquiridos mediante instrumentos de medición.

Los algoritmos codificados mediante objetos de sistema facilitan la reutilización de código a lo largo del proceso de diseño del sistema. El código de MATLAB en punto flotante o en punto fijo se puede incluir directamente en un modelo de Simulink como parte del proceso de arquitectura, modelización y diseño del sistema.

Los ingenieros también pueden generar código C automáticamente a partir del código de MATLAB que emplea objetos de sistema usando MATLAB Coder y, después, utilizar ese código C para la simulación o la integración con otros elementos de diseño C/C++ tras su adecuada verificación.

Arquitectura de sistemas digitales y de RF

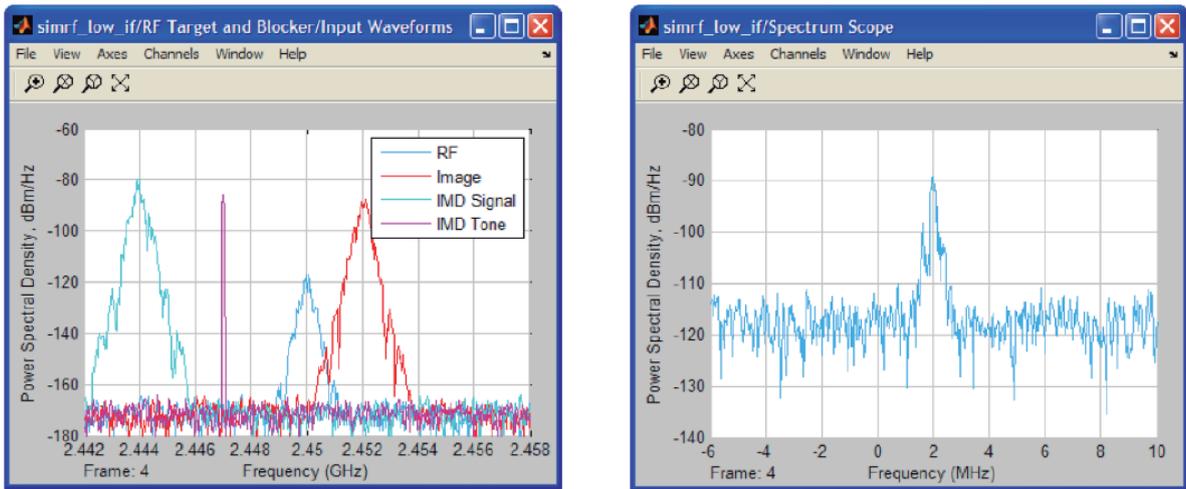
El cálculo del presupuesto para vínculos estáticos es un primer paso habitual en los diseños de RF basados en especificaciones para LTE, Bluetooth, ZigBee, Wi-Fi u otras tecnologías. Este cálculo es un buen punto de partida, pero no incluye la modulación de señales de entrada, los efectos de imagen y otros fenómenos del mundo real. A fin de modelizar y simular de forma efectiva el impacto de los inconvenientes de RF en los sistemas de comunicaciones, actualmente los arquitectos de sistemas hacen malabarismos con varias herramientas desconectadas que admiten diseños de RF digitales o analógicos, pero no ambos.

SimRF es un entorno versátil que proporciona cubiertas de circuitos y capacidad de modelado de banda base equivalente, permitiendo la simulación de la dinámica multifrecuencia en receptores de RF y componentes de tres puertos, tales como los mezcladores. SimRF y Simulink juntos proporcionan un entorno común para modelizar y simular sistemas de RF y banda base en un diseño unificado. El uso conjunto de estas herramientas permite a los arquitectos de sistemas llevar a cabo simulaciones realistas en una fase temprana del proceso de desarrollo, así como tomar decisiones de tradeoff informadas en los diseños que incluyen componentes de RF digitales y analógicos.

La Figura 3 muestra el modelo completo del sistema de un receptor de FI baja de la banda ISM que incluye tanto los componentes de procesamiento de señales digitales como el subsistema del receptor de RF. También se muestran los detalles del subsistema de RF que implementa un receptor Hartley de FI. Al contrario que los métodos tradicionales de modelización de receptores de FI que emplean cascadas de elementos de dos puertos y aproximaciones de frecuencia única, el uso de elementos de tres puertos simplifica el modelo del receptor. Este modelo también emplea tecnología de simulación de cubiertas de circuitos y admite la modelización multifrecuencia.

Los arquitectos de sistemas también pueden explorar la viabilidad y las ventajas relativas de enfoques

Figura 4. Gráficos de densidad espectral de potencia de entrada (izquierda) y salida (derecha) que muestran los resultados del rechazo de imagen de FI baja



alternativos para el rechazo de imagen, tales como una arquitectura de heterodino o de conversión directa en el entorno unificado. Además de simular los efectos de los problemas de RF, los arquitectos de sistemas pueden utilizar los mismos modelos de sistema empleados para el diseño con el fin de efectuar las tareas de verificación de la simulación que normalmente se realizarían en un entorno de laboratorio.

FPGA están las lentas iteraciones de diseño necesarias para encontrar el equilibrio adecuado entre consumo de energía, rendimiento y área.

La Figura 5 muestra un filtro FIR simétrico implementado en aritmética en punto fijo. Para plasmar un filtro así en hardware, los ingenieros deben equilibrar cuidadosamente el rendimiento y la latencia, así como supervisar la cantidad de recursos de hardware utilizados. El resultado de

rutas críticas es una de las funciones clave que proporciona información procesable sobre posibles cuellos de botella en el sistema. Mediante el uso de la información postsíntesis generada por la herramienta de síntesis Xilinx ISE, Simulink HDL Coder anota el tiempo de las rutas críticas en el modelo de Simulink. Los ingenieros pueden usar esta información junto con técnicas de canalización para particionar los diseños y las latencias

Diseño de hardware

Una vez finalizados el diseño de algoritmos y la arquitectura del sistema, el siguiente paso en muchos ciclos de desarrollo es la implementación en FPGA y la verificación de las partes digitales. Entre las causas principales de ineficiencia en la creación de prototipos y la implementación en

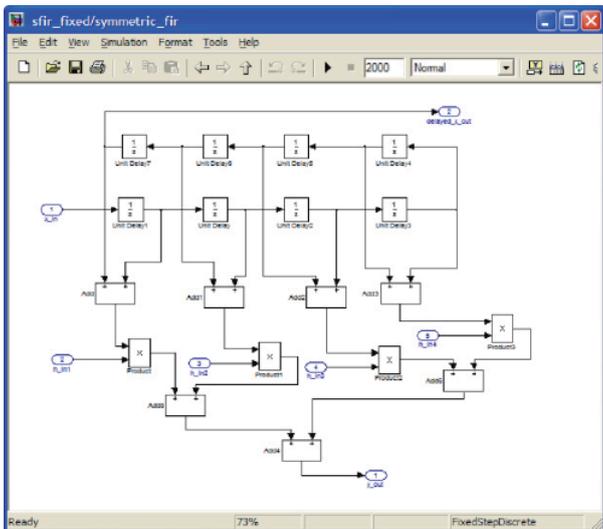


Figura 5. Un filtro FIR simétrico en punto fijo modelado en Simulink

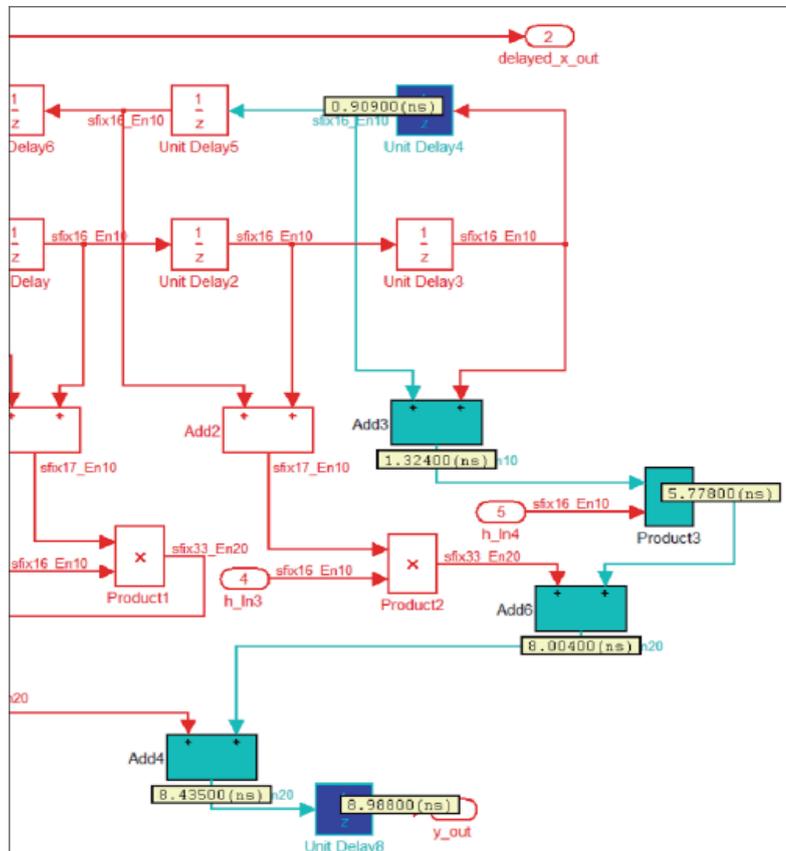


Figura 6. Filtro FIR simétrico en punto fijo con las rutas críticas y las latencias estimadas

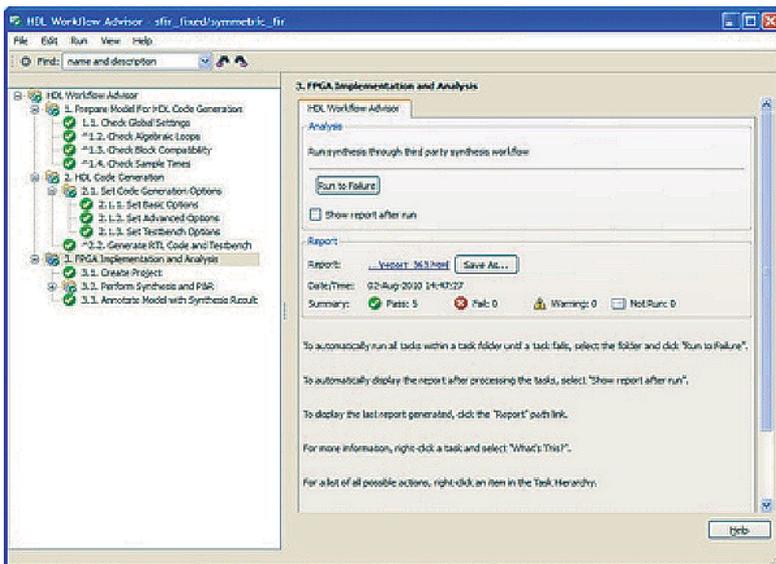


Figura 7. Consola del asistente de flujo de trabajo para realizar iteraciones de diseño rápidas en Simulink HDL Coder

de las rutas críticas. La Figura 6 muestra el mismo diseño de filtro, con las rutas críticas resaltadas automáticamente, junto con la latencia estimada de cada segmento de ruta.

Tal y como se mencionaba anteriormente, la canalización es una de las técnicas clave que emplean los ingenieros para hacer frente a las latencias de las rutas críticas. Uno de los retos conocidos de la canalización es que las rutas paralelas pueden tener latencias dispares, lo que podría conllevar un comportamiento inesperado o indeseado del sistema. La canalización distribuida, que es una técnica empleada con frecuencia para solucionar este problema, ahora se puede automatizar. Al elegir esta opción, los ingenieros pueden reprogramar automáticamente el modelo y equilibrar las latencias introducidas por los registros de canalización de las rutas paralelas relevantes.

En el pasado, estos tipos de iteraciones de diseño y evaluaciones tradeoff requerían una cantidad significativa de tiempo y esfuerzo. La Figura 7 muestra una nueva consola del asistente de flujo de trabajo que permite a los ingenieros realizar las iteraciones de diseño con mucha más rapidez y de forma intuitiva.

Esto resulta especialmente útil para los no expertos en programación HDL pero que necesitan sacar partido del procesamiento en FPGA. Además de utilizar el resaltado de rutas críticas y la canalización distri-

buida, los ingenieros también pueden examinar un informe de utilización de recursos generado automáticamente (como el que se muestra en la Figura 8) para supervisar el tipo y el número de componentes de hardware críticos empleados, así como determinar la mejor opción de arquitectura para una situación concreta mediante la iteración rápida de varias opciones de diseño viables.

Aceleración del diseño en equipos

Los directores de ingeniería de hoy en día se enfrentan al reto de coordinar equipos dispersos geográficamente que trabajan en distintas partes de un sistema global mediante diversas herramientas desconectadas.

En muchos casos, los diseños de nivel de sistema se realizan mejor en entornos gráficos mientras que, algunos detalles de nivel más bajo, se expresan mejor como texto en MATLAB o C. En este artículo se presentan algunos avances nuevos que mejoran las ineficiencias en distintas etapas del flujo de diseño.

En el caso del diseño de algoritmos, los objetos de sistema son un nuevo avance clave para la modelización y la simulación de sistemas de procesamiento de señales y comunicaciones. Ya hay disponibles en MATLAB varios cientos de objetos de sistema para el procesamiento de señales y las comunicaciones listos para

usar. Además, los objetos de sistema admiten la aritmética en punto fijo, y se pueden integrar con Simulink y utilizarse para la generación automática de código C.

SimRF, que ofrece cubiertas de circuitos y banda base equivalente, es una importante herramienta nueva para lograr flujos de arquitectura de sistemas más eficientes. Permite a los ingenieros modelizar componentes de sistemas de RF y banda base en un entorno unificado, así como llevar a cabo simulaciones multifrecuencia reales.^{8e}

A fin de mejorar los tiempos de los ciclos de iteración de diseño en el diseño de hardware y prototipado FPGA/ASIC, existen varios avances recientes, entre los que se cuentan el asistente de flujo de trabajo, el resaltado de rutas críticas, la canalización distribuida, la anotación de corrección y los informes de utilización de recursos, que proporcionan un marco de trabajo e información crítica procesable sobre el rendimiento del sistema y los recursos empleados.

Independientemente de que los equipos sean grandes o pequeños, estén distribuidos geográficamente o situados en la misma oficina, las organizaciones de ingeniería que se enfrentan a las discontinuidades en su flujo de trabajo de diseño pueden aplicar estas tecnologías para simplificar y acelerar el desarrollo de sistemas complejos de comunicaciones y procesamiento de señales. 

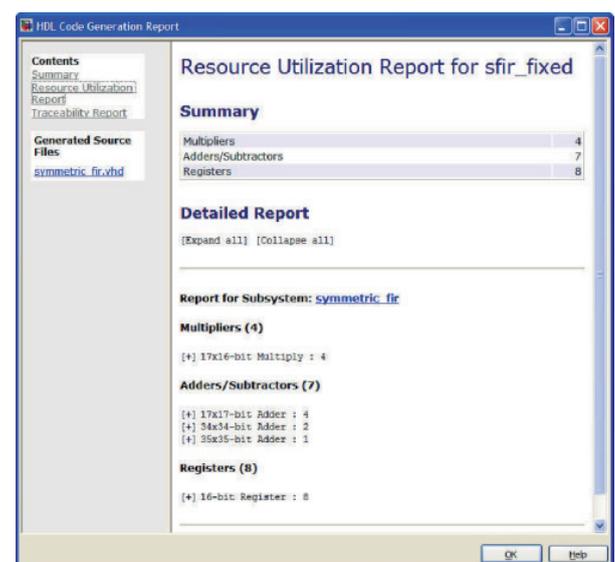


Figura 8. Informe de utilización de recursos que indica el tipo, el número y los detalles de los recursos empleados