

Satisfaciendo la demanda del diseño complejo de comunicaciones

Por Jon Friedman



www.mathworks.com

Jon Friedman es director de marketing del sector aeroespacial y de defensa, The MathWorks

Hoy en día, los ingenieros de comunicaciones deben crear diseños capaces de ajustarse a misiones cambiantes, ciclos de vida de productos más cortos y la creciente potencia de los ordenadores. Como resultado, los ingenieros crean sistemas híbridos que incluyen radiofrecuencia, procesamiento de señales de alta velocidad, procesamiento de señales de velocidad reducida y sistemas de control y de lógica de control. En las etapas iniciales del proceso de diseño, a menudo no queda claro si se deben usar componentes analógicos o digitales y qué porción de un diseño debería implementarse en software o en hardware. Los encargados de diseñar e implementar los sistemas deben hacer conjeturas sobre cómo dividir el diseño, lo que podría dar lugar a diseños por debajo del nivel óptimo y un rendimiento inferior de los sistemas. Normalmente, solo casi al final del proceso de diseño es cuando los encargados de diseñar e implementar los sistemas pueden saber si sus conjeturas iniciales cumplen los requisitos de rendimiento de los sistemas. Si no los cumplen, tienen que rehacer una cantidad significativa de trabajo, lo que provoca costes y retrasos en el proyecto. Para solucionar estos problemas en etapas anteriores del proceso de diseño, los ingenieros de comunicaciones han adoptado el diseño basado en modelos.

Con el diseño basado en modelos, los ingenieros desarrollan un modelo ejecutable, al que se suelen referir como especificación ejecutable, que es independiente de la implementación. En este modelo, los ingenieros de diseños y sistemas pueden desarrollar, probar y dividir el diseño antes de su implementación e integración. Este enfoque les permite localizar los errores en etapas anteriores del proceso, cuando resulta más fácil y menos costoso solucionar los problemas.

A menudo el algoritmo inicial se desarrolla en punto flotante utilizando lenguajes basados en texto como

MATLAB o C/C++. Como ejemplos de tareas se podrían incluir el diseño de filtros en cascada para la conversión digital de frecuencia o el desarrollo de bucles de seguimiento de portadoras. El ingeniero de algoritmos se centra en verificar que el algoritmo de procesamiento de señales cumple los objetivos del diseño, como ajustar la respuesta del sistema en una máscara especificada de frecuencias o realizar el seguimiento de perfiles Doppler esperados.

Tras probar el algoritmo, es posible elaborar aún más el modelo con los detalles de la implementación y verificar el rendimiento del sistema conforme a los objetivos del diseño con los detalles adicionales de la implementación. Por ejemplo, si el destino del algoritmo es una FPGA, entonces es posible añadir al modelo los detalles de punto fijo y evaluar el rendimiento del sistema para confirmar que se siguen cumpliendo los objetivos. También en esta etapa, es posible evaluar el efecto que produce la introducción de componentes no ideales en el modelo del sistema. Por ejemplo, es posible añadir al modelo el comportamiento del amplificador RF utilizando datos de parámetros S medidos. Normalmente, el comportamiento del amplificador RF se modela y analiza en el dominio de la frecuencia, mientras que el algoritmo de comunicaciones se desarrolla en el dominio del tiempo. Al combinar estos modelos en un entorno de modelado ejecutable común, es posible determinar el comportamiento a nivel de sistema y obtener métricas de rendimiento tales como la tasa de errores de bits (BER).

Cuando concluye la elaboración del algoritmo y se verifica el rendimiento a nivel de sistema en el modelo, es posible llevar a cabo las pruebas con distintos lenguajes de implementación a fin de detectar los errores introducidos en la implementación. Por ejemplo, si se divide parte del algoritmo para el procesamiento de señales a baja velocidad en un DSP, la implementación del algorit-

mo será en C/C++. En este caso, los ingenieros pueden usar la generación automática de código a fin de crear rápidamente una implementación de prototipo que se pueda probar en el entorno de modelado con los mismos vectores de prueba utilizados para verificar el rendimiento del modelo. De forma similar, si el destino es una implementación de alta velocidad en un ASIC o FPGA, entonces se puede usar la generación automática de código a fin de crear una implementación en HDL que se pueda probar en Simulink y un simulador EDA, como ModelSim de Mentor Graphics, Incisive de Cadence o Discovery de Synopsys.

Cuando concluyen estas pruebas, la versión final del código generado se puede usar para la producción. Si para el despliegue es necesario un código altamente optimizado o especializado, este último paso de la implementación se puede realizar a mano. Tanto si se genera automáticamente como si se codifica a mano, la implementación final se puede probar en el hardware de destino (microprocesador, DSP o FPGA).

Ahora que hemos descrito este marco de trabajo, veamos un ejemplo de cómo el diseño basado en modelos se podría utilizar para desarrollar un receptor GPS. El sistema GPS lleva totalmente operativo con 24 satélites en su constelación desde 1994, y su uso es omnipresente. Este ejemplo implica la utilización de CDMA (acceso múltiple de división de código) para la medición del retardo de tiempo en el rango de rendimiento. Todos los satélites comparten la misma frecuencia de portadora de 1.57542 GHz para el acceso civil, que no ha cambiado en los últimos 30 años.

El proceso de diseño comienza con la creación de un modelo a nivel de sistema de un receptor GPS a partir de la especificación escrita para una capa física GPS. El modelo (figura 1) contiene los subsistemas de transmisor, canal, receptor y visualización de medición, y presen-

ta numerosos niveles de jerarquía. El modelo de transmisor introduce errores de sincronización para probar el bucle de control de recuperación de sincronización del receptor, mientras que el modelo de canal añade el desplazamiento Doppler para probar el bucle de seguimiento de portadora del receptor. Una vez que la simulación del diseño cumple los objetivos de rendimiento requeridos, el modelo a nivel de sistema se convierte en una especificación ejecutable.

A continuación, el modelo de algoritmo se puede dividir en una porción que residirá en la FPGA y otra porción que residirá en el DSP de punto flotante. Por ejemplo, unos datos de I/Q entrantes que lleguen a una tasa de 8 millones de muestras por sección y pasen a través de un filtro FIR de coseno elevado a la raíz son más adecuados para la FPGA. Por otro lado, una vez que se cancela la propagación de la señal, la tasa de datos es de 1.000 Hz, y un DSP puede gestionarla fácilmente.

Después de que el modelo de receptor trabaje con aritmética de punto flotante, el siguiente paso consiste en elaborar el modelo con atributos de punto fijo que serán necesarios para la partición de la FPGA.

La implementación en la FPGA se puede automatizar en este punto utilizando el modelo de punto fijo de la partición de la FPGA con herramientas como Simulink HDL Coder o Xilinx System Generator. Entonces, utilizando simuladores HDL de empresas como Mentor Graphics, Cadence o Synopsys, es posible verificar que la implementación es funcionalmente equivalente al modelo de diseño. De forma similar, empleando el modelo de la partición del DSP, es posible generar automáticamente el código C con herramientas como Simulink Coder y Embedded Coder para el despliegue. Antes de integrar el código C en el DSP, es posible extraerlo al entorno de simulación y compararlo con el modelo para confirmar su equivalencia funcional, del mismo modo que se ha hecho con el código HDL. El último paso consiste en desplegar el código HDL en la FPGA y el código C en el DSP y confirmar que no se han introducido errores en el hardware de destino.

Usando el diseño basado en modelos, los equipos de ingeniería

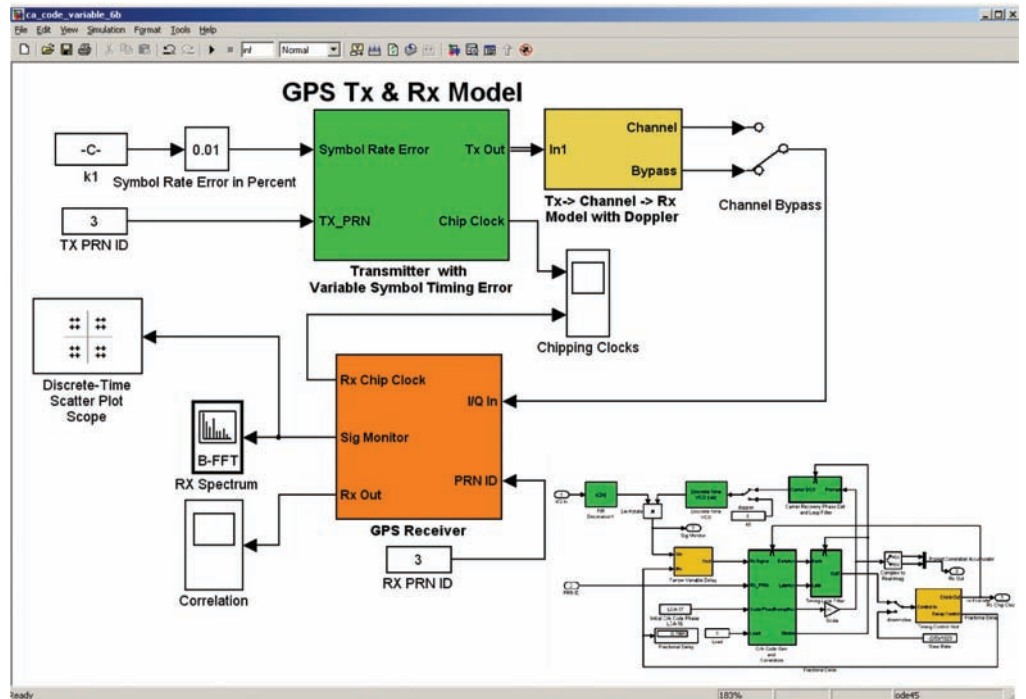


Figura 1. Vista general de un sistema GPS, incluidos el modelo de transmisor con error de sincronización, el modelo de canal con desplazamiento de frecuencia Doppler y el modelo de receptor con bucles de sincronización y recuperación Doppler. El modelo contiene numerosos niveles de jerarquía.

que desarrollan sistemas avanzados de comunicaciones pueden colaborar en un entorno común que les permite capturar tanto el diseño de algoritmos como los efectos a nivel de sistema de hardware no ideal. Como resultado, los equipos pueden

optimizar sus diseños a través de la exploración de diseños, identificar los errores de diseño antes de la implementación y utilizar modernas herramientas disponibles en el mercado para automatizar gran parte del trabajo.

Figura 2. Porción de receptor del modelo particionado con la porción de punto fijo de alta velocidad (en amarillo) y la porción de punto flotante de precisión única de velocidad reducida (en verde). Las radiofrecuencias de los satélites GPS reales se capturan con un analizador de espectro. Esta es ahora la fuente de datos para verificar el modelo con datos del mundo real.

