

# Northrop Grumman automatisiert die Implementierung von Software Defined Radios

Von Gus DiPierro, Doug Jaeger, Patrick Ring und Matt Vondal, Northrop Grumman

FÜR EINEN INGENIEUR IN DER KOMMUNIKATIONSINDUSTRIE IST ZEIT DIE mit Abstand knappste Ressource. Neue Wireless-Systeme oder Datenlinks mit mehreren Megabit/s müssen oft in nur einem Monat implementiert und einsatzbereit sein. Auch Operational Radios müssen häufig möglichst schnell rekonfiguriert werden, um weiterentwickelten Standards, neuen Anforderungen oder wechselnden Einsatzbedingungen genügen zu können. Software Defined Radios (SDR) sind Geräte mit der hierfür erforderlichen Flexibilität, aber diese Anpassungsfähigkeit hat ihren Preis.

Die Entwicklung flexibler oder für die verschiedensten Einsatzzwecke tauglicher SDRs erfordert den Einsatz von viel Zeit, Ressourcen und praktischer Erfahrung, weil viele verschiedene Disziplinen wie Signalverarbeitung, HF Entwurf, die Integration von Mixed-Signal-Komponenten und die FPGA- und Softwareentwicklung daran beteiligt sind. Kommunikations-Ingenieure mit solch umfassender Erfahrung sind außerdem schwer zu finden.

Northrop Grumman beschloss daher, ein System namens Hotrod aufzubauen, das die Implementierung von SDRs automatisiert (Abb.1). Den Kern dieses integrierten Frameworks bilden die Modellierungs- und Simulationsfähigkeiten aus MATLAB®, Simulink® und dem Xilinx® System Generator. Mit dem System sollten vor allem Anwender ohne tiefgehende theoretische Kenntnisse – etwa junge Ingenieure oder Techniker – ein funktionsfähiges

SDR anhand von Parametern wie Datenrate, Modulationsschema und verfügbarer Bandbreite auf einer hohen Abstraktionsebene spezifizieren und konstruieren können.

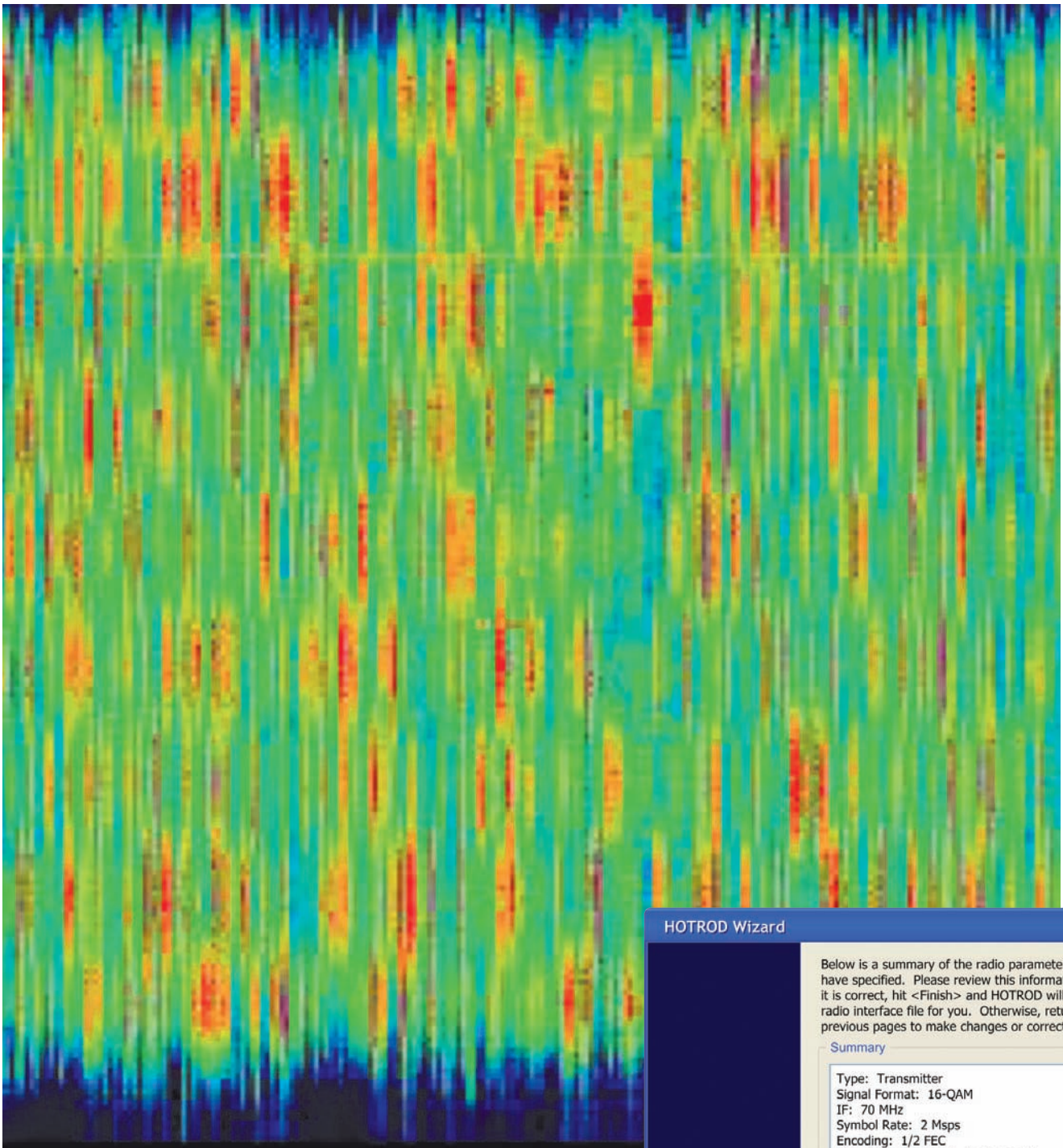
Darüber hinaus konnten mit Hotrod beachtliche Zeiteinsparungen realisiert werden. Hotrod erzeugt eine Roharchitektur, die später von erfahrenen Ingenieuren verfeinert werden kann. Durch den Einsatz von MATLAB und Simulink im Hotrod-Framework konnten die Ingenieure den Prozess des SDR-Entwurfs automatisieren und solche Systeme nun praktisch ohne Eingriff von außen synthetisieren, wodurch sich die Entwicklungszeit für rekonfigurierbare Plattformen um eine Größenordnung reduzieren ließ.

Den Ingenieuren war klar, dass Hotrod erheblich mehr leisten musste, als lediglich vordefinierte Entwürfe anhand von Spezifikationen auszuwählen. Es musste sich intelligent

anpassen können, etwa indem es mehrere Trägerfrequenzen verwendet, um die Vorteile von nicht direkt benachbarten Spektralbändern nutzen zu können.

## Entwicklungsziele

Die Zielsetzung war nicht, dass Hotrod die gleiche, vollständig optimierte Lösung erzeugt, zu der auch ein erfahrenes Team bei gleichen Eingabeparametern oder Spezifikationen kommt; vielmehr sollten das Wissen und die Denkprozesse von Ingenieuren mit jahrelanger Erfahrung im SDR-Entwurf in ein automatisiertes, regelbasiertes Expertensystem umgesetzt werden. Das Team steuerte daher zunächst eine konservative Implementierung an, die Basisanforderungen des Anwenders erfüllt. Hierzu wurde eine Bibliothek erprobter Architekturen mit einem Gerüst verknüpft, in dem sich Funktionsmodule nach Bedarf neu konfigurieren lassen.



*Dieses Wasserfall-Diagramm zeigt, wie die Nutzungsdichte eines einzelnen Funkkanals über die Zeit variiert. Selbst am gleichen Standort trifft man daher immer auf eine neue, einmalige Funkumgebung. Im Gegensatz zu konventionellen Radios lässt sich ein Software Defined Radio (SDR) dynamisch rekonfigurieren und damit veränderlichen Anforderungen oder Umgebungsbedingungen anpassen.*

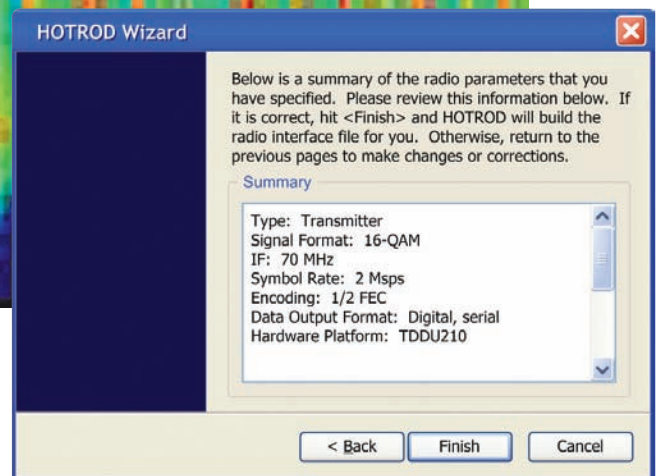


ABB. 1. In der Übersichtsseite des Hotrod Wizard sind die wichtigsten Charakteristiken für den Entwurf des SDRs aufgeführt.

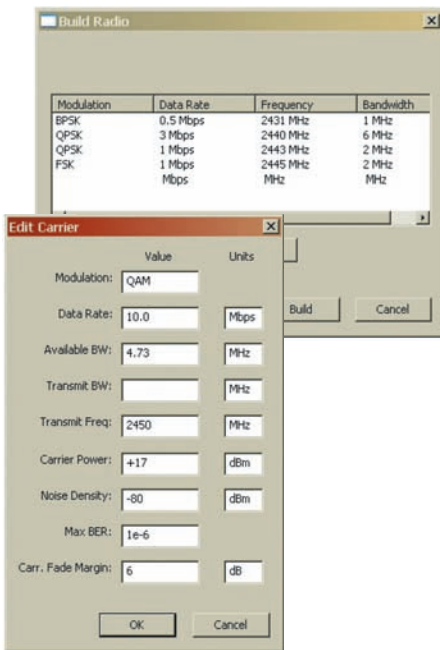


Abb. 2. Hotrod-Dialogboxen.

## Hotrods Bibliothek mit funktionellen Blöcken

Die erste Version von Hotrod konnte ausschließlich FPGA-spezifische Lösungen erzeugen. Zu diesem Zweck wurde eine Bibliothek aus Simulink-Blöcken entwickelt, die Hardwaremerkmale von Funktionen wie Verschlüsselung, differentielle Enkodierung, Fehlerkorrektur, Modulation, Synchronisation und Entzerrung exakt auf FPGA-Strukturen zugeschnitten modellieren. Diese vor allem mit dem Xilinx Blockset modellierten, funktionellen Blöcke waren für die Implementierung auf Xilinx-FPGAs optimiert und mit standardisierten I/O-Schnittstellen sowie Eingabeparametern für verschiedenste Konfigurationen ausgestattet. Dies war relativ leicht möglich, weil das Team dazu aus einem Fundus vorhandener Entwürfe schöpfen konnte. Wie auch viele Softwareingenieure pflegen viele der Digital-Designer eine Art „Kochbuch“, dem sie gängige Funktionen bei Bedarf entnehmen können. Zur Qualitätssicherung erwies es sich jedoch als notwendig, die korrekte Synthetisierbarkeit jedes Blockes vor seiner Aufnahme in die Bibliothek zu verifizieren. Hierzu dienen sowohl HDL-Simulationen als auch der Auf-

bau von Hardwareprototypen, die dann über repräsentative Parameterbereiche hinweg getestet wurden.

## Wie Hotrod automatisch ein Simulink-Modell aufbaut

Ein Experte im Hardwareentwurf könnte allein mit Simulink- und System Generator-Blöcken ein für Xilinx-FPGAs optimiertes Radio Block für Block oder Subsystem für Subsystem aufbauen. Für weniger erfahrene Ingenieure oder Techniker ist die Entwicklung eines SDR aber auch dann noch eine Herausforderung, wenn sie lediglich funktionelle Blöcke aus einer Bibliothek auswählen müssen. Hotrods regelbasierte Engine wurde so angelegt, dass sie Blöcke aus der Bibliothek auswählt, richtig im System platziert und dann alle Parameter den Spezifikationen gemäß einstellt.

Hotrod automatisiert diesen komplexen Prozess vor allem mit Hilfe von MATLAB und Simulink. Viele Funktionen aus der Bibliothek verfügen über mehrere alternative Strukturen, die als Blockparameter definiert sind. Hotrod setzt diese Parameter gemäß den Spezifikationen des jeweiligen SDR. Es konfiguriert beispielsweise Modulator-Blöcke mit einer BPSK-, QPSK- oder 16-QAM-Modulation und ruft dann Signalverarbeitungs-Methoden aus MATLAB auf, um die passenden Filterkoeffizienten zu berechnen.

Nach Zusammenstellung aller Spezifikationen erzeugt ein Ingenieur in der Regel ein neues Simulink-Modell, baut die erforderlichen Blöcke ein und verdrahtet diese dann miteinander. Hotrod geht analog vor, beschleunigt den Prozess aber mithilfe des MATLAB Application Programming Interface (API) und konstruiert automatisch ein funktionsfähiges Simulink-Modell. Aufgrund der Flexibilität und der Skripting-Fähigkeiten der API lässt sich auf einfache Weise eine Schnittstelle zwischen Simulink und CLIPS, einer kostenlosen Experten-System Entwicklungsumgebung, mit dem die regelbasierte Engine erzeugt wird, erzeugen. Hotrod kann ohne Eingriff des Anwenders ein MATLAB-Skript

mit allen zum Aufbau des Simulink-Modells nötigen Befehlen generieren und ausführen.

Dieser Ausschnitt eines automatisch erzeugten Skripts legt zentrale Blockparameter anhand von Vorgaben mithilfe der `set_param()` Funktion fest. Das Skript nutzt außerdem die Autorouting-Fähigkeiten von Simulink über den Aufruf der `add_line()` Funktion, die systematisch Signalleitungen und -busse zwischen den Komponenten erzeugt.

```

...
src = ['STC_Library/Modulator IQ'];
src = hruGetBlkPath(src, LibList);
dst = [syspath '/Mod_A'];
add_block(src, dst);
set_param(dst, 'Position',...
[120 60 150 90]);
set_param(dst, 'MaskValueString',...
'BPSK 1.0 Mbps');
src = ['xbsIndex_r4/Gateway In'];
src = hruGetBlkPath(src, LibList);
dst = [syspath '/Freq_A'];
add_block(src, dst);
set_param(dst, 'Position',...
[60 60 90 90]);
set_param(dst, 'arith_type',...
,Unsigned');
set_param(dst, 'bin_pt', ,0);
set_param(dst, 'n_bits', ,8);
add_line(syspath, 'Freq_A/1',...
'Mod_A/1');
...

```

## Arbeiten mit Hotrod

In der Praxis können Anwender mit Hotrod ein Radio auf einer hohen Abstraktionsebene beschreiben und dann seine Systemparameter wie Kapazität, Typ und Hardware-Plattform in der Sprache des Endnutzers definieren. Dabei muss noch nicht einmal jeder einzelne Parameterwert explizit vorgegeben werden. Ein Entwickler kann beispielsweise die gewünschte Datenrate, die maximal tolerierbare Bitfehlerrate (BER) und die Rauschdichte für einen Datenlink vorgeben und Hotrod die Berechnung der dazu am besten passenden restlichen Parameter überlassen.

Abbildung 2 zeigt zwei Dialogboxen für die Spezifikation von Parametern wie Modulation, Datenrate, verfügbarer Bandbreite und nutzbaren Frequenzen. Kann Hotrod auf einer vorgegebenen Plattform keinen den Anforderungen genügenden Entwurf erzeugen, dann gibt es eine beste Schätzung der erzielbaren Leistung aus - etwa dass angesichts der Bedingungen statt der vom Anwender geforderten 100 Mb/s bestenfalls 73 Mb/s erreichbar sind.

Hotrod bietet weniger erfahrenen Anwendern einerseits Hilfestellungen an und verbirgt vor ihnen einige Details, gibt andererseits Experten dennoch umfangreiche Steuermöglichkeiten. Mit dem von Hotrod erzeugten Simulink-Modell kann ein System wie üblich vor dessen Hardwareumsetzung simuliert werden. Ingenieure können Stimulus-Signale mit Simulink erzeugen und die Reaktionen des Systems mit Scopes überwachen oder die Simulationsergebnisse mit MATLAB visualisieren. In diesem Stadium können sie den Entwurf in Simulink aber auch modifizieren, etwa durch Hinzufügen von Patch-Logiken in HDL. Auf diese Weise werden gegebenenfalls mehrere Design-Iterationen durchgeführt, wobei die Änderungen in Simulink vorgenommen, der neue Entwurf simuliert und schließlich die Ergebnisse grafisch dargestellt werden.

### FPGA-Implementierung des Entwurfs

Wenn der Anwender mit dem Entwurf zufrieden ist, ruft Hotrod den Xilinx System

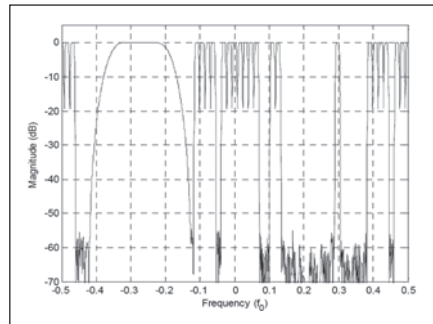


ABB. 4. Dieses Diagramm zeigt, dass synthetisierte Radios sich selbst stark ausgelasteten Spektren anpassen können.

Generator auf und erstellt eine FPGA-Netzliste aus dem Modell. Der Anwender muss dazu eine Wrapper-Vorlage für die gewählte Hardware-Plattform vorgeben, anhand derer Hotrod beispielsweise mit Xilinx ISE ein passendes Device Configuration File erzeugen kann. Für die neuen rekonfigurierbaren Plattformen, die auf Xilinx Virtex-4- und Virtex-5 Bausteinen basieren, wurde dies mit großem Erfolg durchgeführt.

Dieser automatisierte Prozess öffnet Möglichkeiten für einige interessante Anwendungen. So konnte beispielsweise eine automatische „Spectral Scavenging“-Fähigkeit demonstriert werden. Ein Scan-Empfänger ermittelt hier an einem Standort im Feld die Frequenzen und Periodizitäten aller in einem bestimmten Frequenzband vorhandenen Signale und gibt diese an einen Entwickler weiter, der Hotrod auf einem Laptop ausführt. Nach Vorgabe einer gewünschten Datenrate entwirft, synthetisiert

und lädt Hotrod das neue Radio-Design völlig selbstständig auf eine Plattform. Diese Fähigkeit, freie Zeit- und Frequenzressourcen einer beliebigen Umgebung zu nutzen, ohne dazu permanent ein kompliziertes Radio-System betreiben zu müssen, könnte einen enormen Vorteil in Situationen bedeuten, die schnelle Reaktionen erfordern.

### Beschleunigung der SDR-Entwicklung

Durch die Automatisierung des Entwicklungsprozesses mithilfe von Simulink und MATLAB verringert Hotrod erheblich die Entwicklungszeiten von SDRs. Je nach Größe des Entwurfs kann ein Ingenieur mit Hotrod einen FPGA in weniger als einer Stunde implementieren. Die manuelle Entwicklung eines Radios mit vergleichbaren Spezifikationen würde mindestens den zehnfachen Aufwand benötigen. Selbst wenn ein Ingenieur sämtliche erforderlichen Schritte kennt, benötigt er geraume Zeit, um Blöcke auszuwählen, Parameter einzustellen, Blöcke zu verbinden, anfallende Datenfiles durch den Syntheseprozess zu schleusen und nicht zuletzt Zeit für das unvermeidliche Debugging.

Bei der Entwicklung mit Hotrod kann man Millionen möglicher Single- und Multi-Carrier-Konfigurationen erzeugen. Durch Sammlung erfolgreicher Entwürfe in einer Datenbank kann Hotrod diese außerdem später wiederverwenden und so den Prozess weiter beschleunigen.

Der künftige Erfolg kommunikationstechnischer Projekte hängt empfindlich davon ab, wie schnell Ingenieure auf neue Anforderungen reagieren können. Hotrod wird hierbei sicherlich eine wichtige Rolle spielen. ■

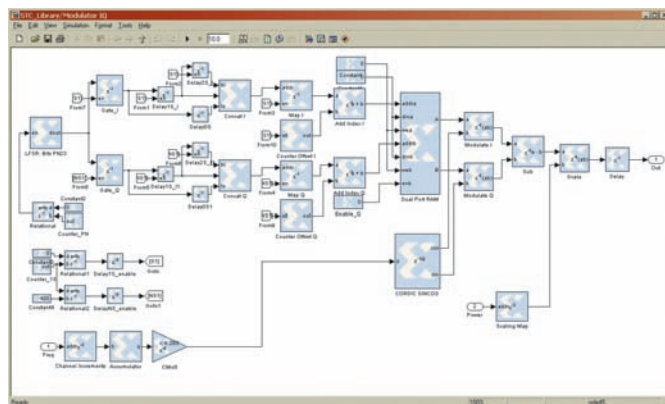


ABB. 3. Dieser aus Xilinx-Blöcken in Simulink aufgebaute Modulator ist eine der vielen Komponenten in der Hotrod-Bibliothek.

➤ **Quellen**

**NORTHROP GRUMMAN**  
[www.northropgrumman.com](http://www.northropgrumman.com)  
 Kontakt: [matthew.vondal@ngc.com](mailto:matthew.vondal@ngc.com)

**WEBINAR: Model-Based Design and FPGA Implementation with Simulink**  
[www.mathworks.de/nn8/wbnr30807](http://www.mathworks.de/nn8/wbnr30807)