**Challenge**
Implement FPGA based radio signal processing in a small team mainly consisting of people with signal processing and programming background

**Solution**
Use HDL Coder to generate VHDL for signal processing

**Results**
- Successful implementation running on FPGA
- Generated code easy to integrate into main design
- Very short lead time for changes in design
RADIO TEST BED DESIGN USING HDL CODER
OUTLINE

› Introduction
› Challenges in test bed design
› Solutions enabled by using HDL Coder
› Experiences made along the way
› Conclusions
BIOGRAPHY

Tomas Andersson

› Licentiate of Engineering in Communications and Information Theory, KTH, 2006
› Working at Ericsson since 2008
   - Test bed development
   - Base band signal processing
   - Implementations in C and Assembly
   - Limited prior experience with FPGA:s and VHDL
› Runs a small horse farm in the spare time
SYSTEMS & TECHNOLOGY

› Department at Ericsson
› Responsible for technology leadership
› Involved in
  – Standardization
  – Concept development
  – Pre-studies
  – Radio technology strategic work
› Test bed activities
  – Demonstrate new technology
  – Proof of concept
  – Pre commercial testing
TEST BED DEVELOPMENT

› Done in small teams (<10 persons)
  - Highly skilled in programming and signal processing
› Short lead times
  - Typical project duration ~1 year
› Important to be first
  - Constant race against competitors
› Cost optimization not a priority
CHANGING SCOPE

› Previous test beds
  – Focus on system level or base band features
  – Commercial front end radio, re-banded to test frequency
  – Drawback: Difficult to add features to radio front end

› New test bed
  – Develop front end radio within test bed organization
  – Possible to implement features in front end FPGA
  – New development required
In a short time frame
Develop signal processing for an FPGA
In a team with little experience in writing VHDL
Use HDL Coder for VHDL generation
FRAMEWORK

Code generation for signal processing

Manual implementation of a few interfaces

FPGA vendor supplied IP cores

FPGA vendor high-level integration tool
DESIGN GUIDELINES

› Use HDL supported blocks in simulation
› Use fixed point from start of design
› Simulink primitives for low level algorithm design
› Matlab function blocks for control logic
› Subsystems to raise abstraction level
INTRODUCING HDL CODER INTO WORKFLOW

› Training period, ~2 months
  – Learning Simulink
  – Testing HDL Coder
  – Finding limitations in synthesis
  – Finding workarounds for limitations

› Implementation, 1 month
  – Design of custom components
  – High level design of signal processing chain

› Integration, 1 week
  – Include generated code in FPGA framework
  – Resolve timing issues
BENEFITS

› A single model for simulation and code generation
  – No hand offs between systemization and implementation
  – Ready for FPGA as soon as simulation works
› Short iterations for changes in design
› Simulink block diagram resembles “manager level” power points
  – Drawback: makes it look a bit too easy
› FPGA implementation done by person with limited VHDL competence
CHANGING SCOPE AGAIN

› New ideas emerge
  – Other carrier configurations
  – Wider bandwidths

› Need for major changes in design
  – New systemization
  – Re-tuned algorithms
  – Changed implementation

› Using HDL Coder we were able to demonstrate that the change was possible

› Working demonstration in less than one week!
CONCLUSIONS

› Code generation is well suited for test bed purposes
› A single model for simulation and code generation significantly shortens design lead time
› Generated code is well structured, readable and accurate

› HDL Coder has been a key factor in this project