RAPID PORTING OF AN SCA-COMPLIANT FM3TR WAVEFORM

Per Johansson, Zhongren Cao, William Hodgkiss

California Institute for Telecommunications and Information Technology (Calit2)
University of California, San Diego
La Jolla, CA

Outline

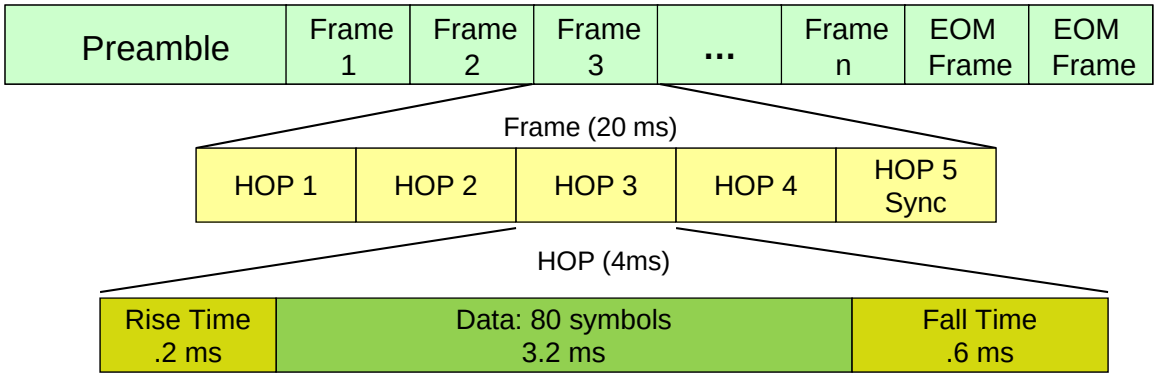
- Introduction
- FM3TR Waveform
- Waveform Porting
- Demonstration Setup
- Conclusion

Introduction

- Software Communication Architecture (SCA)
  - Adopted by the JTRS program
  - Defines common waveform component interface
  - Enables software reuse
- Waveform porting
  - Move waveform component implementation among different host environment
  - A common core framework to install, initiate and control waveforms and radio functionalities
  - JTRS Information Repository
- FM3TR – an ideal waveform for education and experimentation
  - Open standard
  - Good for promoting SCA and waveform porting to a broad audience
    - Universities
    - Research institutes
    - Private companies

FM3TR
Future Multi-band Multi-waveform Modular Tactical Radio

- FM3TR is an international test waveform
  - Initially developed by the Air Force Research Labs (AFRL)
  - International cooperative effort among US, UK, France and Germany
- Spectrum features
  - 25 kHz bandwidth
  - Frequency hopping (250 hops per second)
  - VHF and UHF military bands (30 MHz-400 MHz)
- CPFSK modulation
- Voice: CVSD 16 kHz PCM
- Data: capable of 9.6 kbps



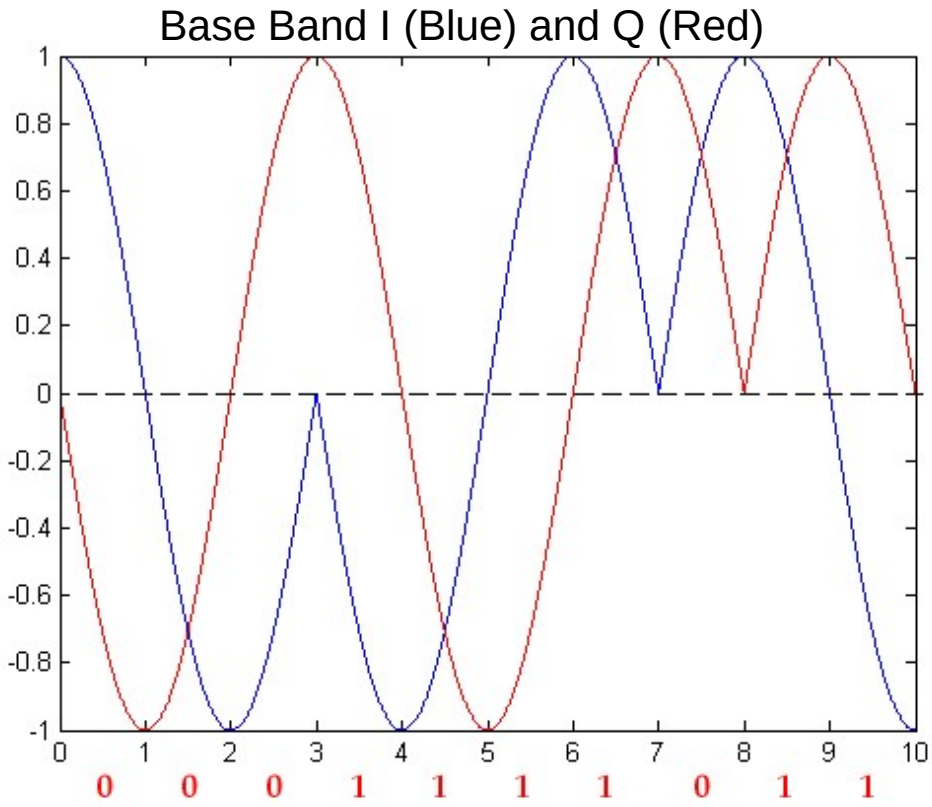
FM3TR CPFSK -- Basics

- Waveform expression

$$s(t) = A\cos\left(2\pi f_c t + 2\pi h \int_0^t \sum_{i=0}^{[t/T]} b_i g(\tau - iT) d\tau\right)$$

$$b_i = \begin{cases} +1, \text{bit } 0 \\ -1, \text{bit } 1 \end{cases}$$

$$g(t) = \begin{cases} \frac{1}{T}, 0 \le t < T \\ 0, \text{otherwise} \end{cases}$$



Waveform Porting Targets

- Port an SCA compatible FM3TR software to an SDR platform
  - Unknown original platform
  - New platform: Spectrum Signal's SDR-4000
- Implement the data mode for FM3TR
  - Only voice mode available in the basic code
- Implement To Applications
  - Push-To-Talk (PTT)
    - On the voice channel
  - Instant Text Messaging (ITM)
    - On the data Channel

Push-To-Talk Application

- The PTT application operation
  - Between SDR units over the voice mode FM3TR
  - Broadcast with a single shared channel
  - No provision for media access control
- The PTT client
  - Implemented by Java
  - Hosted on a PC connected with Ethernet to the SDR platform
  - The audio device uses 16kHz PCM

Instant Text Messaging Application

- The ITM application operation
  - Between SDR units over the data mode FM3TR
  - Tow operation modes
    - Conference mode
      - Text messages are broadcasted to all participating users
    - User-user mode
      - Text messages are unicasted to the targeted user
- The ITM client
  - (Java) is hosted on a PC connected to the SDR unit over Ethernet.

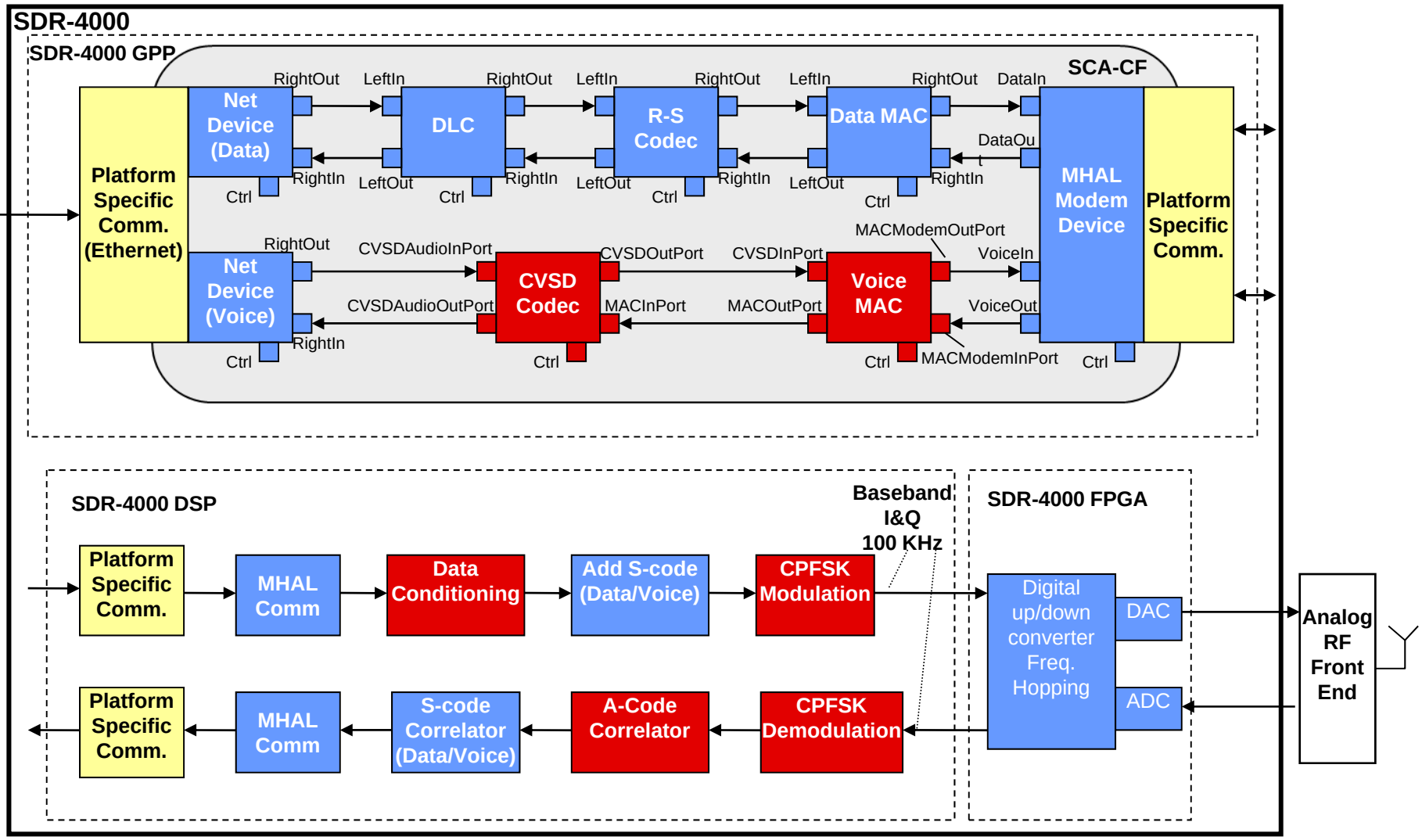
Instant Text Messaging (ITM) Application

- The ITM application operates between 2-3 SDR units
- Conference mode where text messages are broadcasted to all participating users
- User-user mode where text messages are unicasted
- The ITM client (Java) is hosted on a PC connected to the SDR unit over Ethernet.

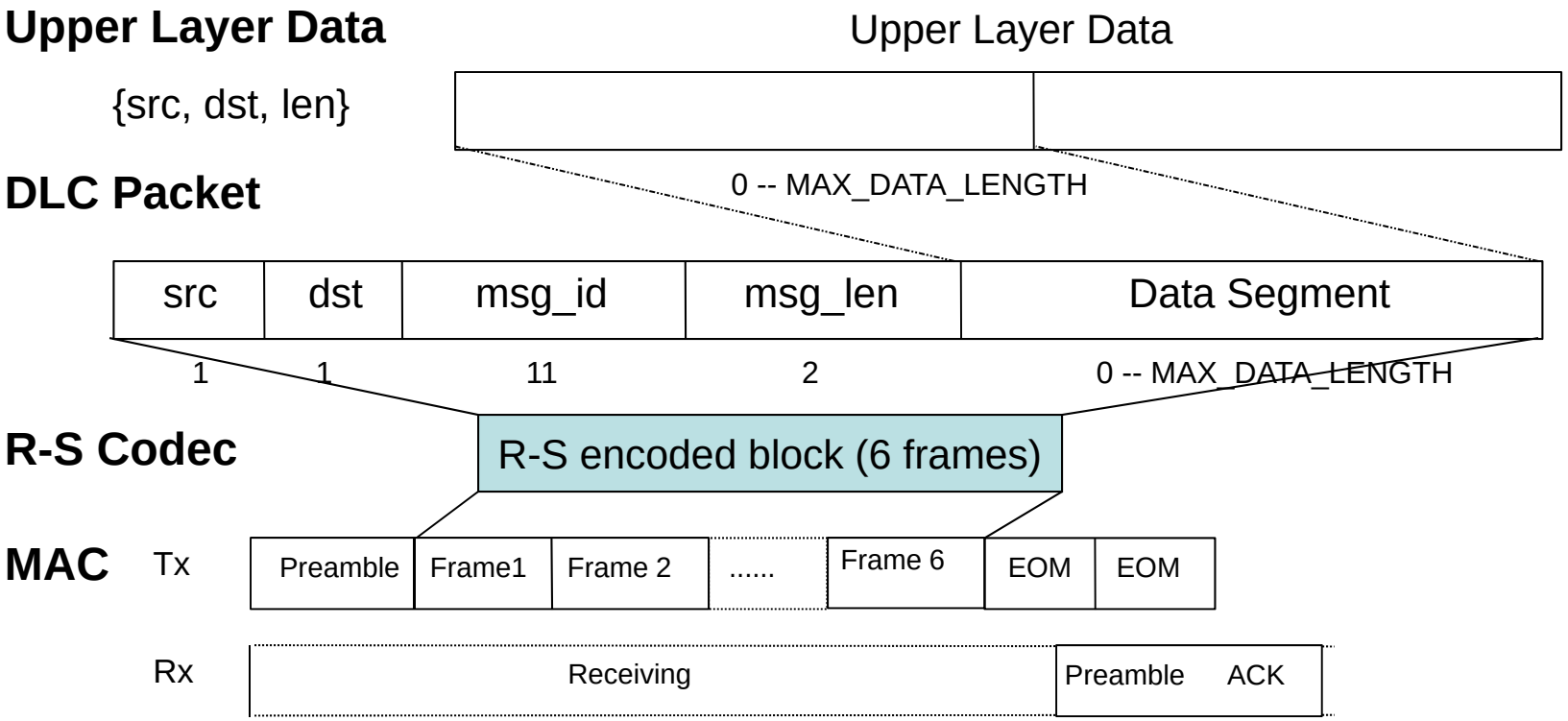
FM3TR Data Mode

- Reed-Solomon coding added to support the FM3TR robust mode.
- A data link layer control (DLC) component is introduced to handle ARQ for unidirectional data traffic.
  - For broadcast data flows, ARQ is not provided
- No full-duplex mode ARQ implemented
- No media access control implemented
  - No check for busy media before sending (or resending) data packets

Functional Blocks – SDR-4000

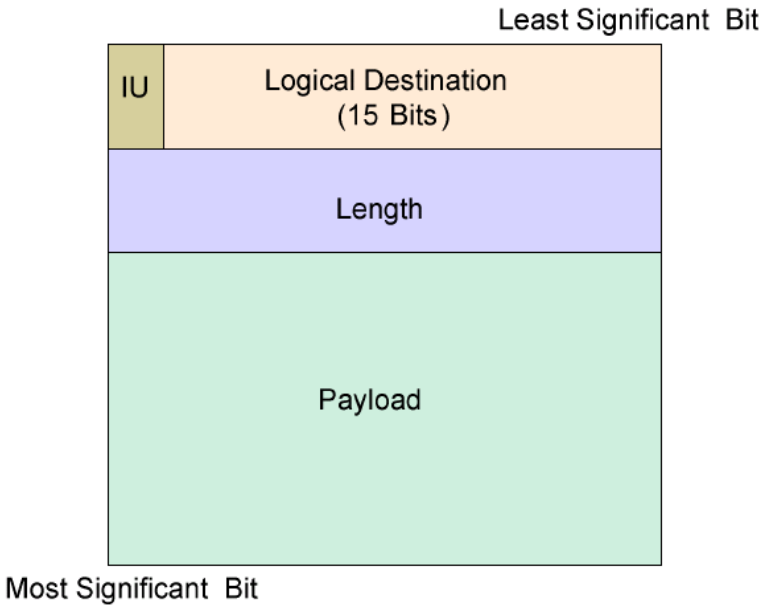


Data: DLC, R-S Coding and MAC



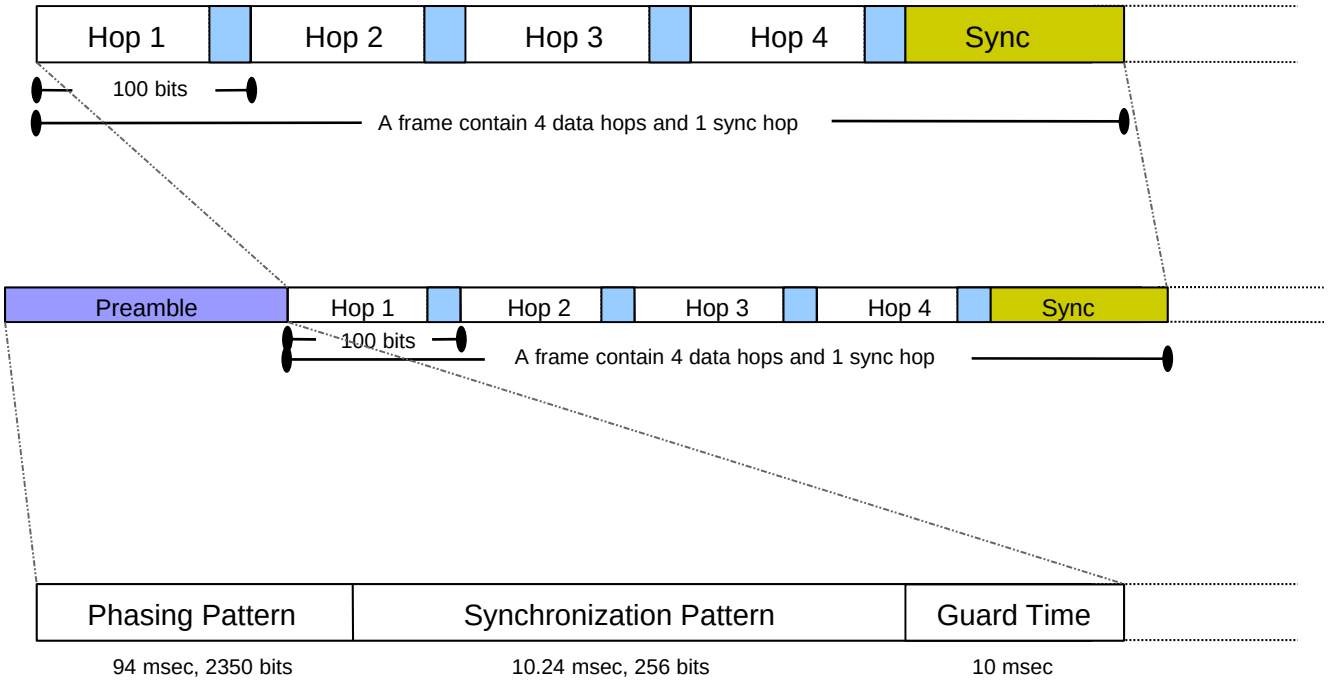
- DLC segments messages into blocks with padding
- Blocks are Reed-Solomon coded
- MAC formats frames and hops
  - Adds EOM frames and sync hops
- ARQ scheme on a per block basis (stop-and-go)

MHAL Modem Device



- An SCA compliant MHAL Modem Device resource
- Encapsulating the voice and data packets into the MHAL frame structure
- Corresponding MHAL function in the DSP

FM3TR Modem (DSP)

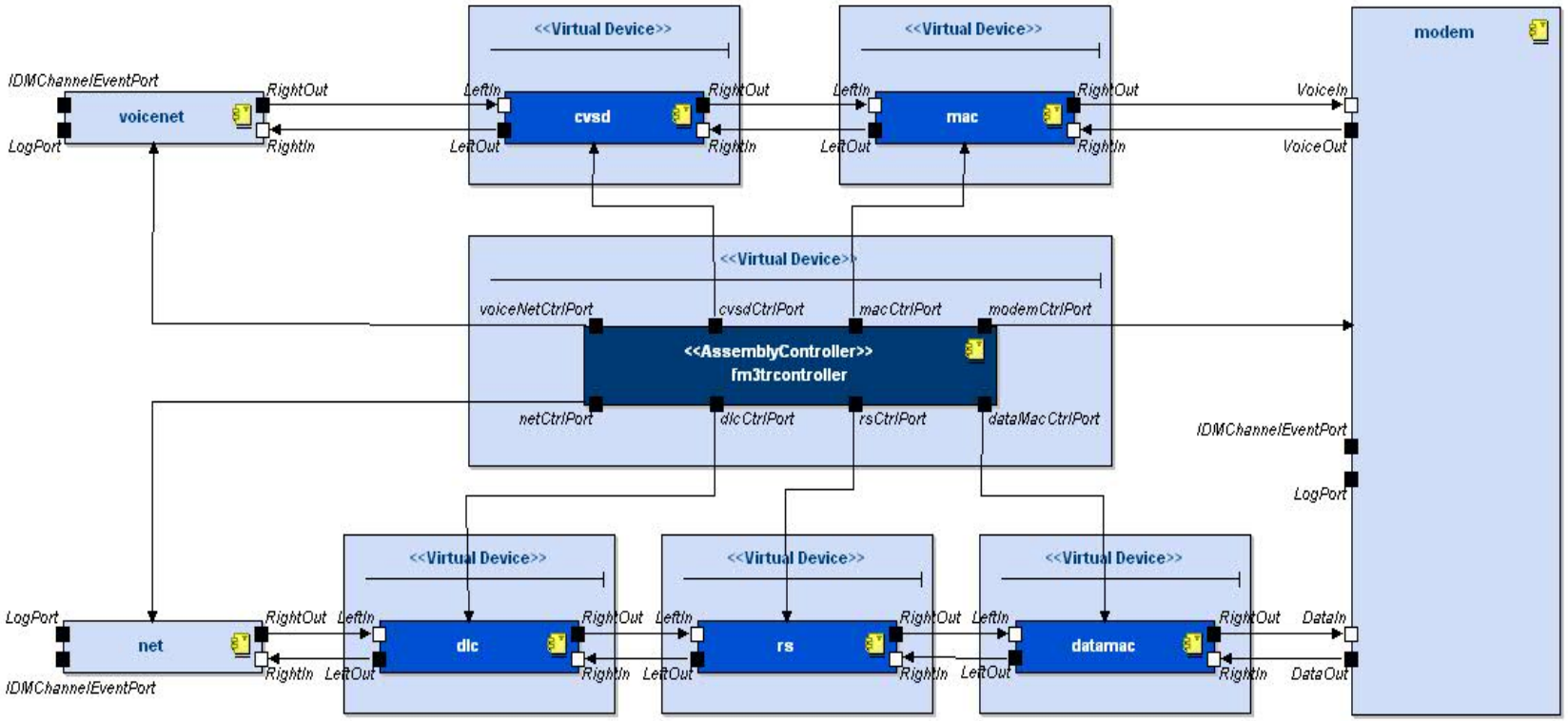


- Adds a preamble to in the beginning of each message (voice or data)
- Adds padding between each hop for ramp-up and ramp-down

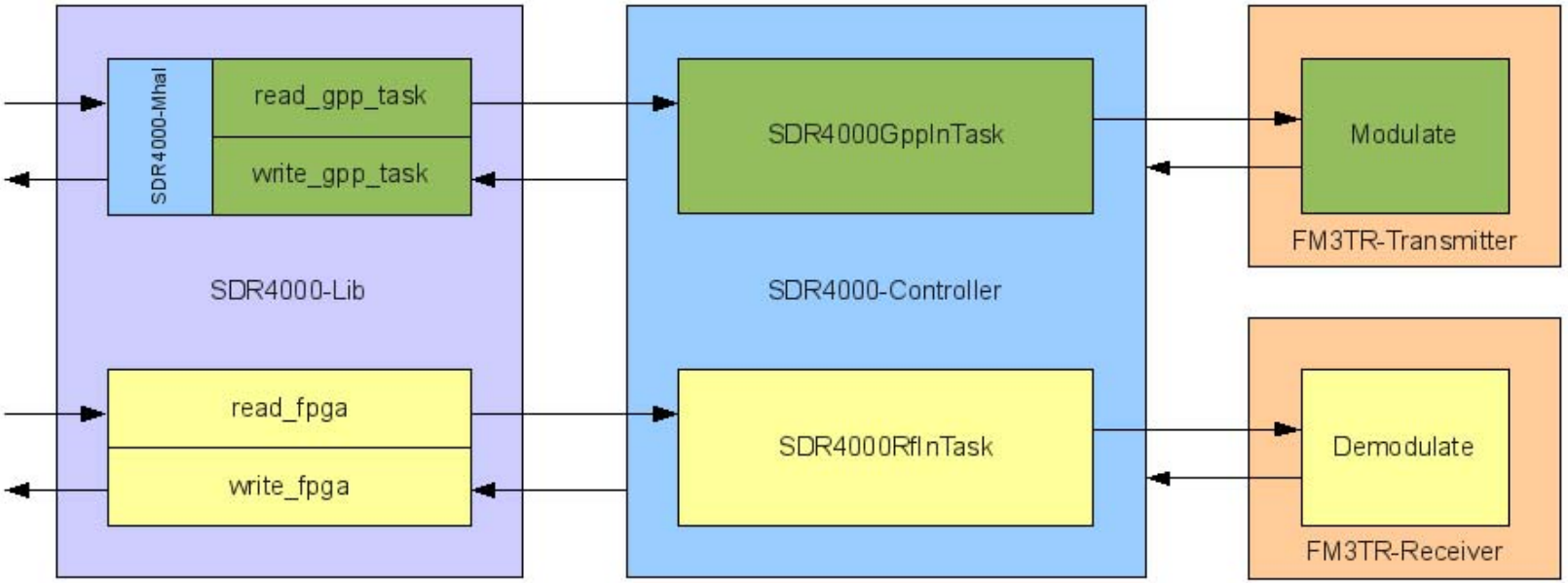
SDR-4000 porting

- Learning curve quite steep: new platform, new OS, new tools (SCARI++)
  - And getting GPP-DSP-FPGA to actually intercommunicate!
- SCARI++ tools made SCA code porting quite easy
- DSP porting straight forward
  - Most time spent on using/understanding quicComm
  - FH implementation added quickly
- RF channel added via a one NI PXI1045 unit
  - Shared analog RF channel
  - Same unit – synch is avoided
- Debugging a bit cumbersome
  - Spectrum and CRC very helpful
  - Still more to learn!

FM3TR SCA components



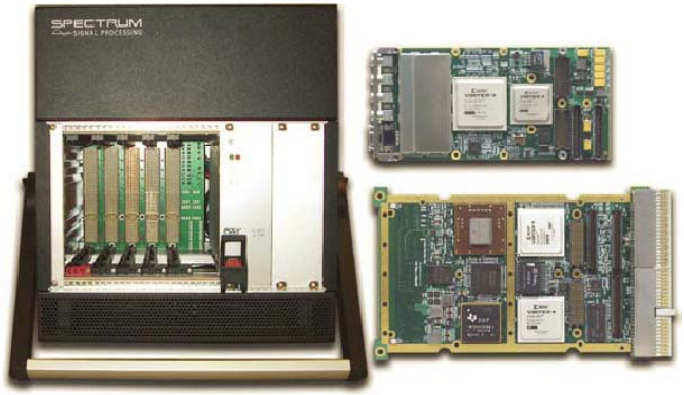
Modem (DSP) code structure



Structure of FM3TR port to SDR4000

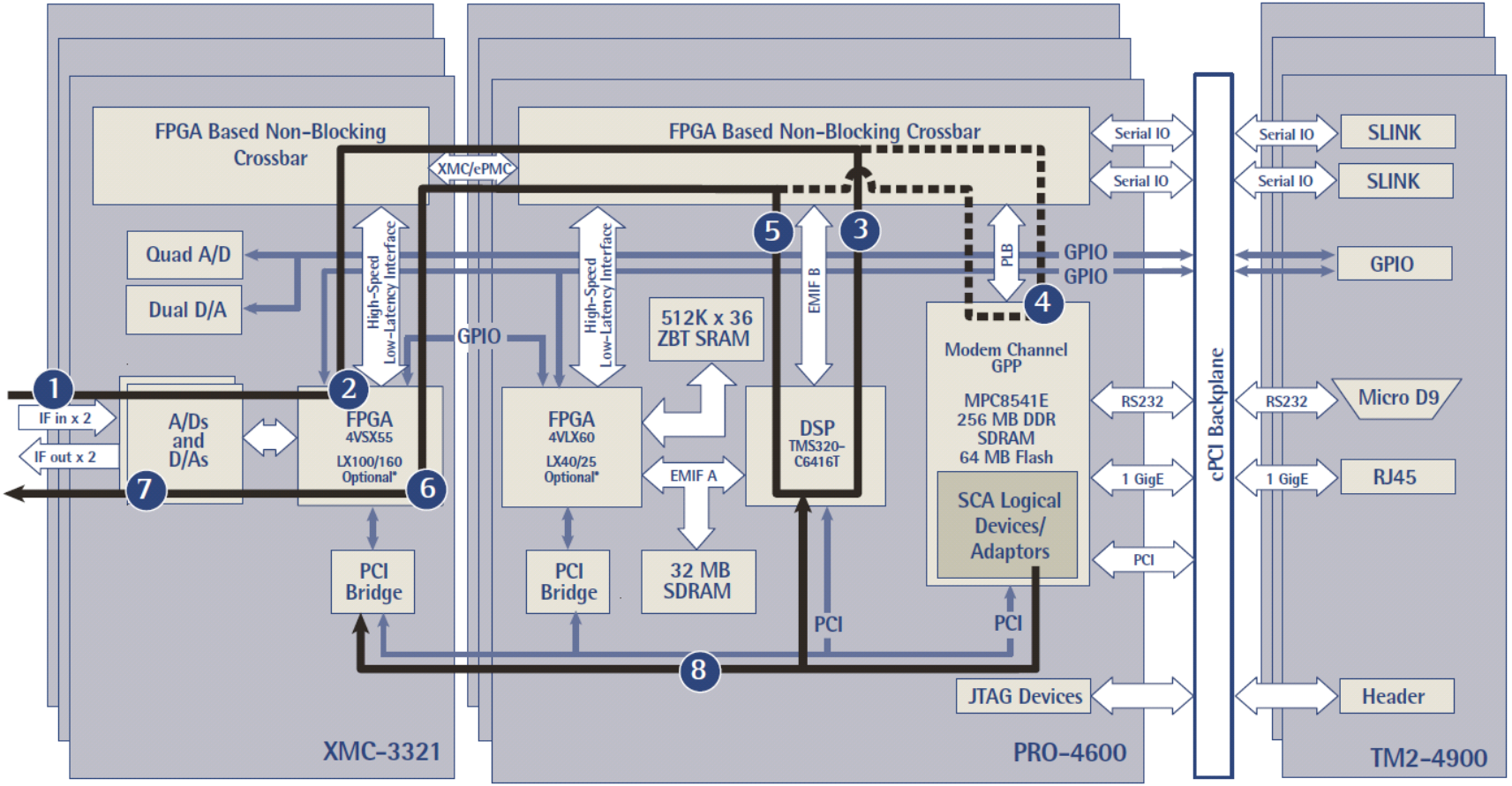
SDR-4000 Development Environment
(by Spectrum Signal Processing)

- Xilinx Virtex-4 FPGA
- TI-C6416 DSP
- MPC8541E GPP
- Green Hills INTEGRITY
- Xilinx ISE FPGA Tools

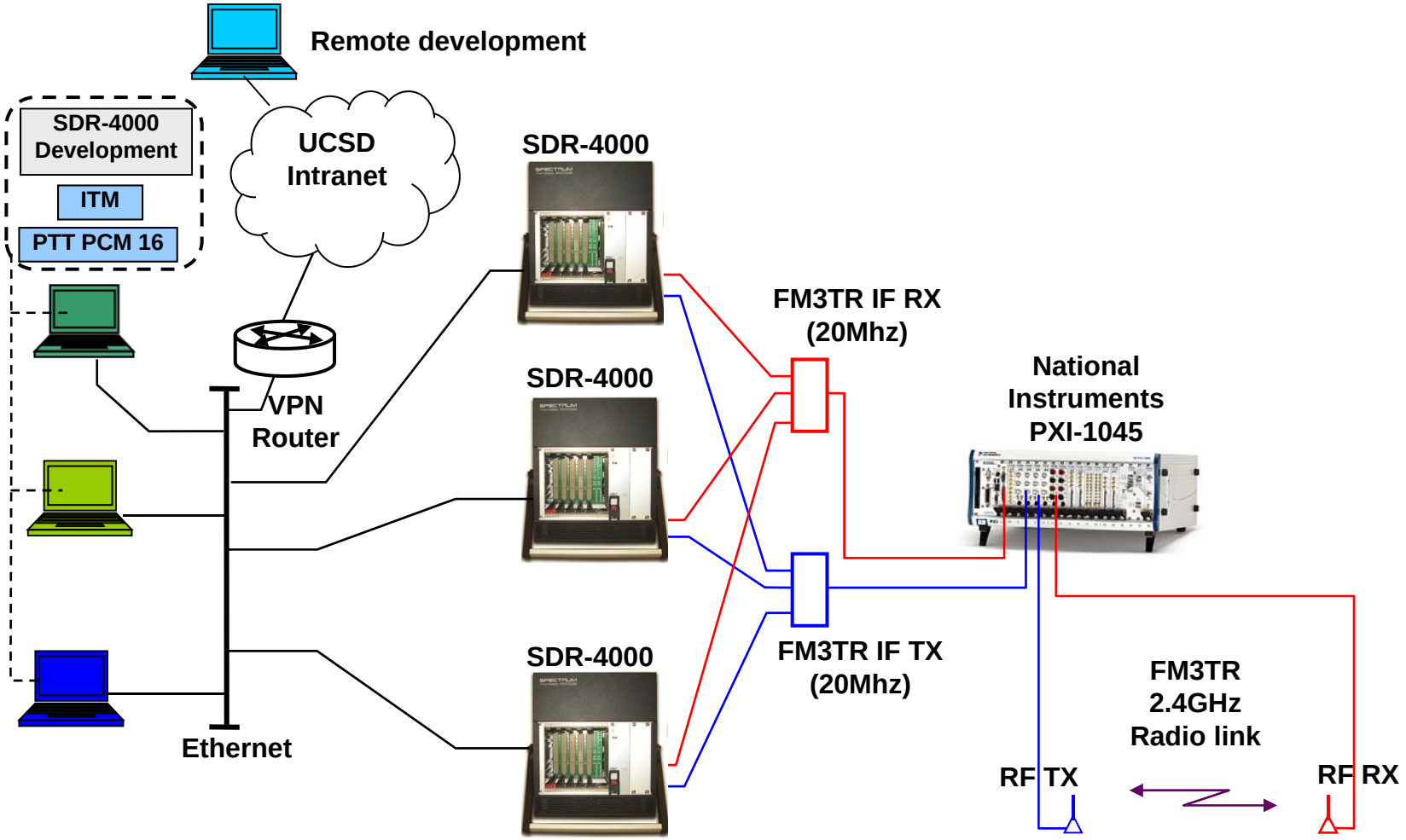


- Texas Instruments Code Composer Studio
- SCARI++ SCA Core Framework v2.2 from CRC
- ORB Express from Objective Interface Systems (OIS)
- NOTE: *Typically **NOT** found in a university lab!!*

SDR-4000 Architecture



FM3TR SDR-4000 Lab and Demonstrator Configuration
Fall 2008



Summary

- FM3TR voice (ported) and data (developed) for
  - Linux
  - SDR-4000 (including RF)
- A lot of the code actually outside SCA (GPP) in this solution
  - Modem (DSP)
  - Up/down conversion (FPGA)
- A few changes made to base code were necessary
  - Modified structure and bug fixes both for SCA (GPP) code and modem (DSP) code)
- *A lot more to learn!*