WInnF Transceiver Facility

David Hagood (Cynosure)

Presentation at WInnComm 2021

2 Dec 2021 – online event

Agenda

Introduction

Design philosophy – PIM vs PSMs

The PSM (Platform Specific Model) Specifications

Conclusions

Introduction

Why WInnF Transceiver Facility

Portability, re-usability, and interoperability are the goals

- SCA focuses on the software level – reuse the waveform on multiple platforms
- Hardware level is left to the platform vendor – platform should be able to run multiple waveforms.

Hardware changes...

- Different numbers of IF stages.
- Banks of filters vs. tunable filters.
- Different distribution of gain controls.
- etc.

Software shouldn't have to.

Isolate the knowledge of the hardware issues to one software element – the Transceiver Facility

Background APIs

JTNC MHAL RF Chain Coordinator

- Initially developed by the JTRS program
- First published API for Transceiver functionality
- Release: 2007 (latest update: v3.0, oct. 2013)

WInnF Transceiver Facility v1.0

- Openly developed as a WInnF standard
- Leveraging the European project End-to-End Efficiency (E3) and FRA MoD R&T project PEA AL
- Release: Jan 2009

ESSOR Architecture Transceiver API

- Developed by the ESSOR program
- Turns WInnF Facility v1.0 into a JTRS-like API specification
- Finalized: 2010 – Released to WInnF: 2016

WInnF Transceiver Facility v2.0

WInnF XCVR v2.0 is a WInnF Facility (SDR Standard)

- One PIM Spec: fully implementation agnostic API and Properties
- Several PSM Specs: PIM Spec mapping to programming paradigms: Native C++, FPGA and SCA

A successful and unprecedented international harmonization

- 12 contributors **from 6 countries**: DGA, ENSTA, FKIE, Harris, Hitachi Kokusai Electric, JTNC, Leonardo, NordiaSoft, Rockwell-Collins, Rohde & Schwarz, Thales, Viavi Solutions (formerly Cobham), NordiaSoft (now part of Viavi Solutions)
- Reflecting SDR background from **worldwide manufacturers**
- Reflecting lessons learned from **EUR military programs**: ESSOR (ESP, FIN, FRA, ITA, POL, SWE) and SVFuA (GER)

Not just for communications...

WInnF XCVR v2.0 isn't just for communications

- It just focuses on getting data in and out
- Could be radar pulses as much as communications
- Supports MIMO
- Wide bandwidth can support ELINT/SIGINT
- Multiple apps can use the same data streams – piggyback ELINT on comms
- Omit the tuning sections (which are optional) and it becomes a high speed ADC/DAC interface for any use, e.g., control electro-optical modulators, servos

Design Philosophy

XCVR Facility PIM – Overview

The Transceiver Facility standardizes

- The service-oriented *Transceiver Application Programming Interface* (*Transceiver API*)
- Standard associated *Transceiver Properties*

It therefore supports portability of *applications* and openness of *transceiver* implementations (a.k.a. hospitality)

Transceiver is the processing stage between the antenna and the radio physical layer baseband processing

Transceiver I/Os are the baseband signal and the radio signal

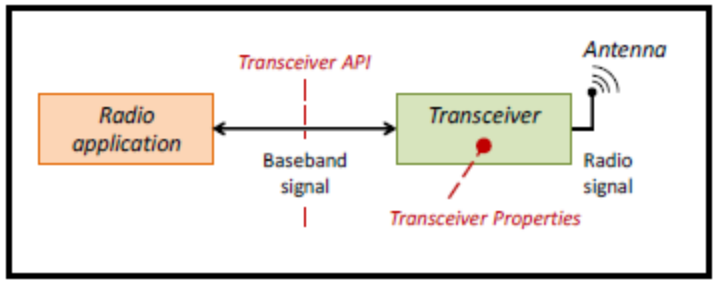

Figure 1 Overview of Transceiver Facility

XCVR Facility PIM – Concepts

Presets: isolating hardware differences

- Specifies what the signal processing wants the hardware to provide.
- Sample rate, channel bandwidth, allowable ripple, power ramps.
- No specifics to hardware implementation.

Bursts: what to send and when

- Immediate – good for PTT
- Absolute – send at specific time, good for frequency hopping
- Relative – send some time after last burst, good for TDMA
- Strobe-based waveforms – hardware triggered, good for Test and Measurement

Metadata

- Optional feature. Data source, gain settings, or anything else.
- Can map to VITA-49

XCVR Facility PIM – Concepts

Tuning

- Specifies frequency, gain settings for a burst
- Can be done in advance of the burst, can be queued

Data movement:

- TX can be queued before the burst starts
- Bursts can be queued

Timing

- Times are specified in either “System Time” (a.k.a. GPS) or “Terminal time” (local clock)
- Abilities to relate one to the other (what system time is this terminal time? What terminal time is 03:21 9/9/2022?)
- Ties to the Timing Service, previously discussed.

Distribution

All the above functions can be split to different processing nodes. This allows data generation on a fast processor, tuning/hopping on a secure processor, timing on an FPGA.

PIM services groups

- **Essential SGs**
  - `Tuning` Control of the tuning parameters
  - `BasebandSignal` Packet-based exchange of samples blocks
  - `BurstControl` Creation and termination of bursts

Additional SGs

- `Management` General control
- `Transceiver Time` Access to transceiver time
- `Strobing` Strobes triggering for creation of bursts
- `GainControl` Automated gain control
- `Notifications` Notification of events and errors

The PSM Specifications

PSM Specifications Overview

PSM Specs under finalization

- Native C
- Native C++
- FPGA
- SCA

PSM Specs under consideration

- Other IDL-based component frameworks, e.g. Redhawk

Reported usages

- SCA PSM: consistent content (developed alongside the spec) used in current developments of a radio manufacturer and one Viavi Solutions product

User-specific PSMs are possible, although discouraged from portability standpoint

Conclusions

End of the presentation
Thank you for your attention

Any questions?

Contacts:

- David.Hagood@cynoinc.com