Spectrum Sniffer Software Implementation for ISM Bands

SDR`10

30 November – 3 December, 2010 • Washington, DC

E. Picco

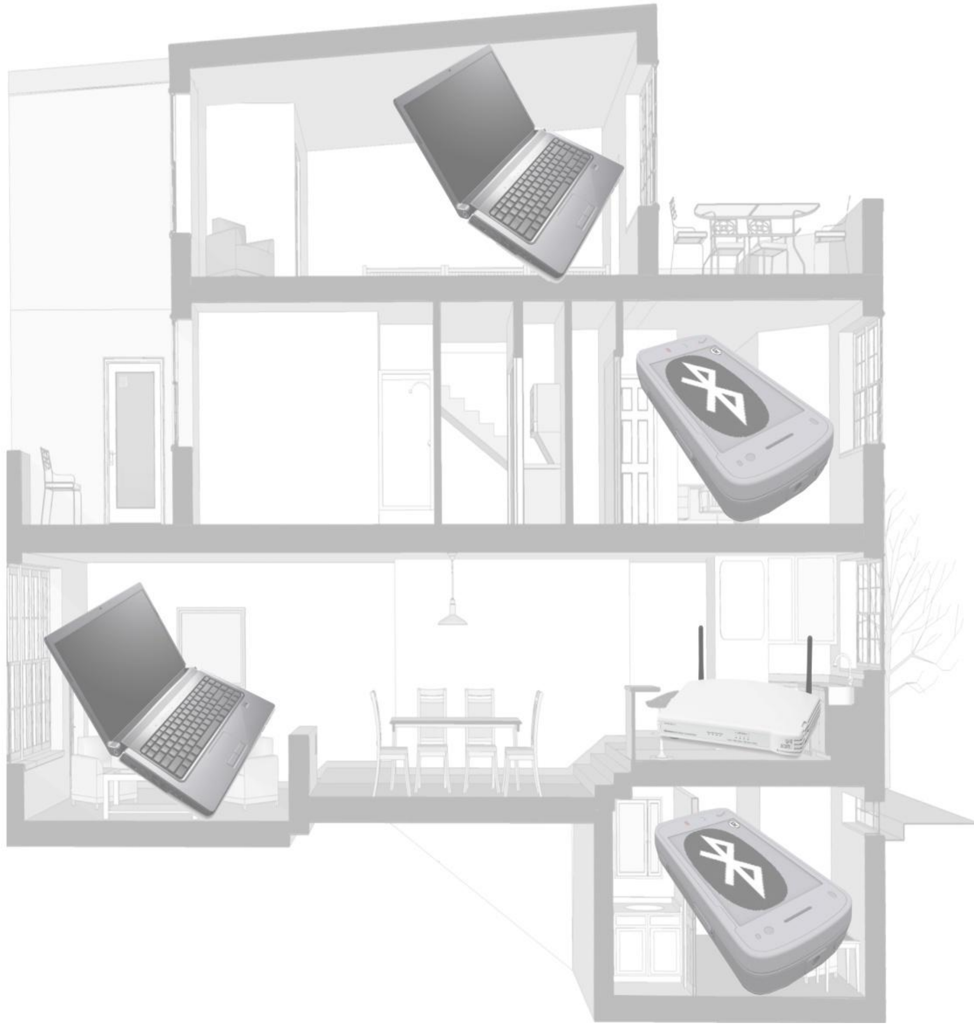
Advanced System Technologies

STMicroelectronics, Agrate Brianza, Italy

Outline

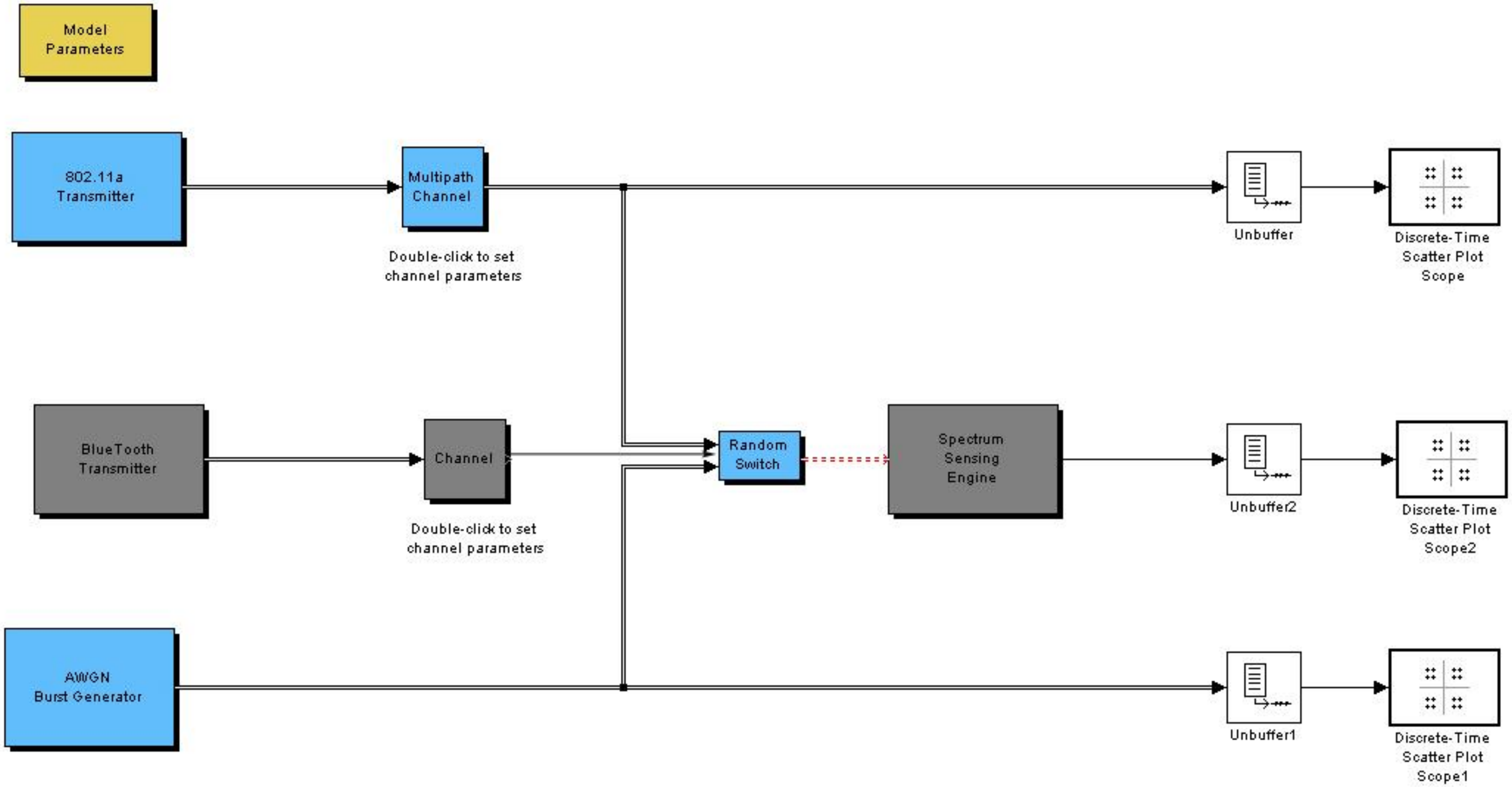
- Scenario: ISM Band devices indoor
- Background: cyclostationary signals
- Cyclostationary based Spectrum Sensing
- The Spectrum Sniffer Architecture
- Results

Scenario



- Home scenario
  - IEEE 802.11a/b/g/n devices operating in the 2.4GHz – 5GHz, typically PCs, modem-routers, mobiles
  - Bluetooth devices operating in the 2.4GHz
  - DECT devices
  - Microwave devices

Scenario model



Stochastic model switching received signals between typical IEEE 802.11a and Bluetooth transmission or Additive White Gaussian Noise

Cyclostationary Signals

- For the considered scenario
  - Received signal is $y(t)=\int_{-\infty}^{+\infty} h(\tau;t)x(t-\tau)d\tau$
    where x(t) is a periodic signal
  - Channel autocorrelation is $A_h(\tau;\Delta t)=E\left[h^*(\tau;t)h(\tau;t+\Delta t)\right]$
  - Transmitted signal autocorrelation function is

$$R_x^{\alpha}(\Delta t)=\lim_{T\to\infty}\frac{1}{T}\int_{-T/2}^{T/2}E\left[x(\tau)x^*(\tau+\Delta t)\right]e^{-j2\pi\alpha\tau}d\tau$$

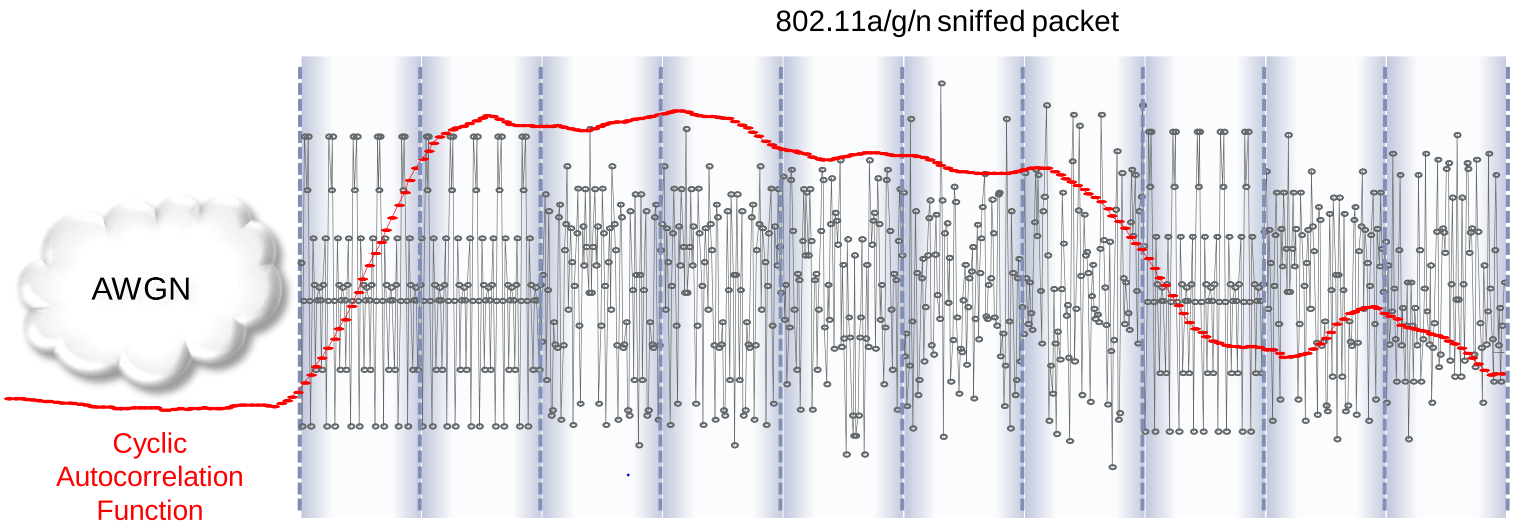
  - So received signal Autocorrelation function is

$$R_y^{\alpha}(\Delta t)=\lim_{T\to\infty}\frac{1}{T}\int_{-T/2}^{T/2}E\left[y(t)y^*(t+\Delta t)\right]e^{-j2\pi\alpha t}dt=\int_{-\infty}^{+\infty}A_h^*(\tau;\Delta\tau)R_x^{\alpha}(\Delta t)e^{-j2\pi\alpha\tau}d\tau$$

Cyclic Spectral Density

$$S(f,\alpha)=\sum_{\tau=-\infty}^{+\infty}R_y^{\alpha}(\tau)e^{-j2\pi f\tau}$$

Cyclostationary based Spectrum Sensing

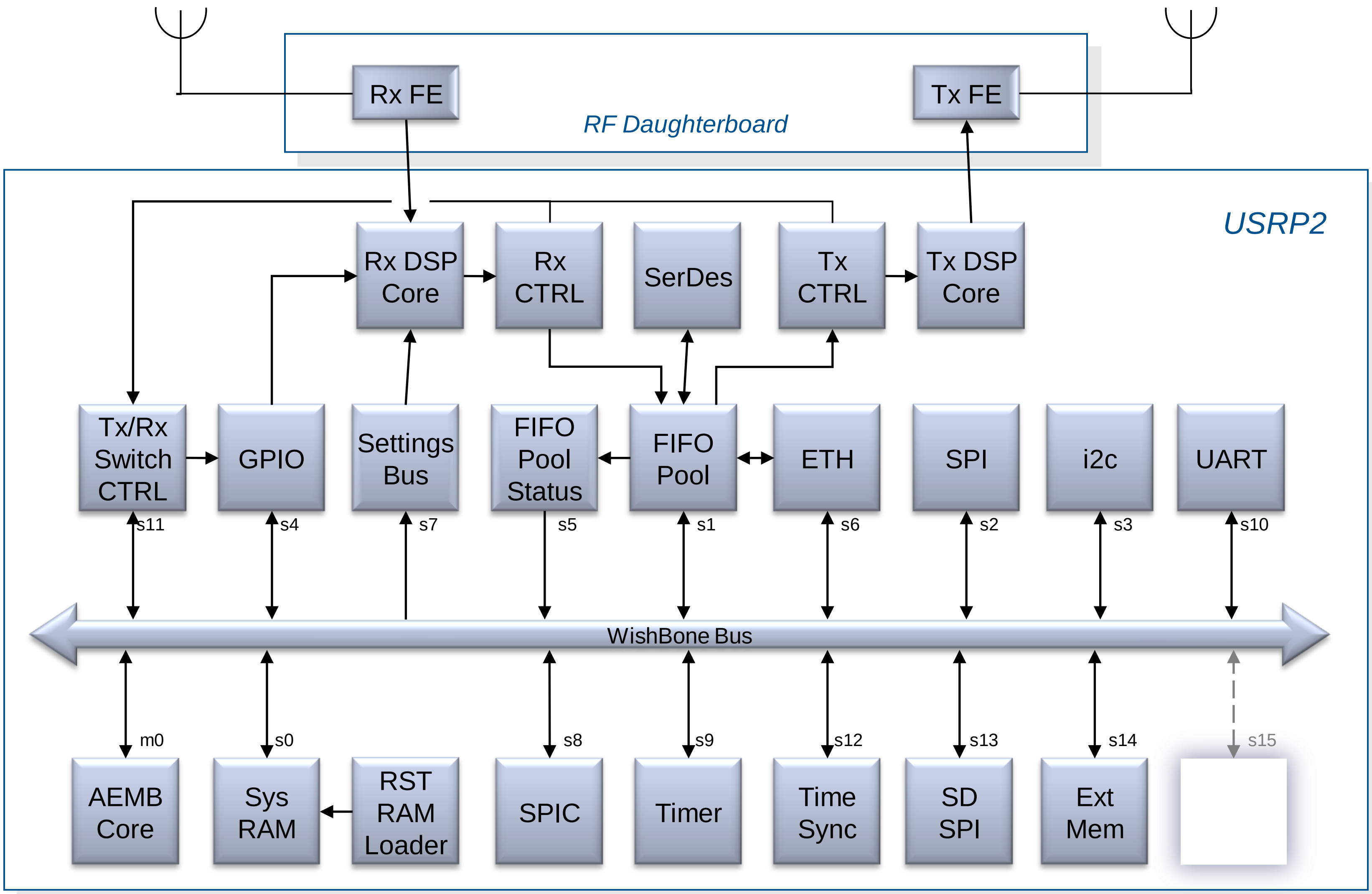


- Cyclostationary algorithm applies to signals with periodic symbol structure that include all the typical signals present in the ISM Band (802.11n, Bluetooth,…)
- Based on the Cyclic Autocorrelation Function computed to obtain the Cyclic Spectral Density as a reliable and consistent estimate of the Power Spectral Density
- Cyclic Autocorrelation Function tends to zero if AWGN only is present
- Setting the bandwidth limits and the sweeping step, CSD information for the whole band can be obtained to be passed to the CRM

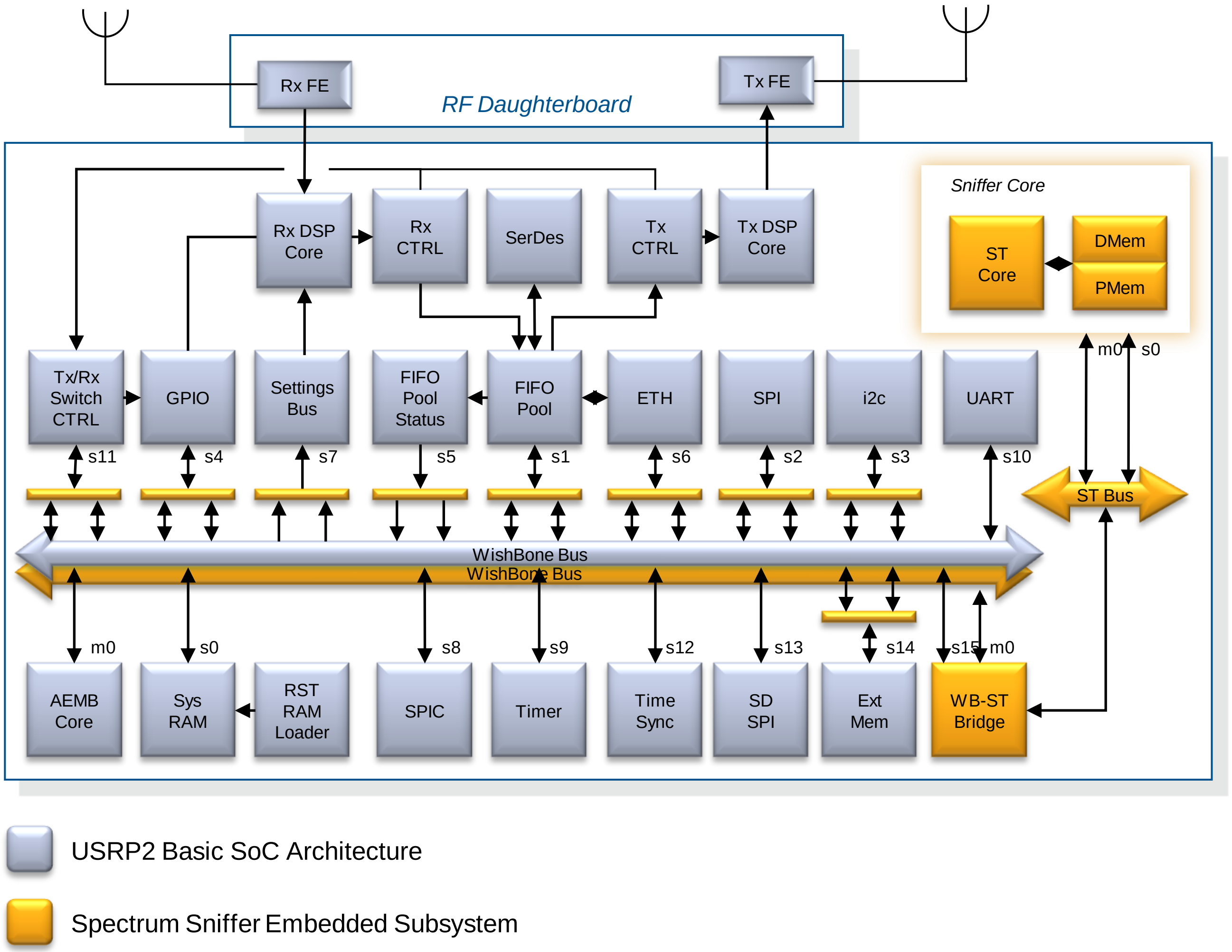
The Spectrum Sniffer Architecture

- Based on USRP2 (Universal Software Radio Peripheral 2) SoC
- Embedding a programmable proprietary ST Core subsystem dedicated to Spectrum Sniffing routines
- As for the SDR philosophy, a totally programmable general purpose architecture has been adopted, no need for dedicated HW

Universal Software Radio Peripheral 2



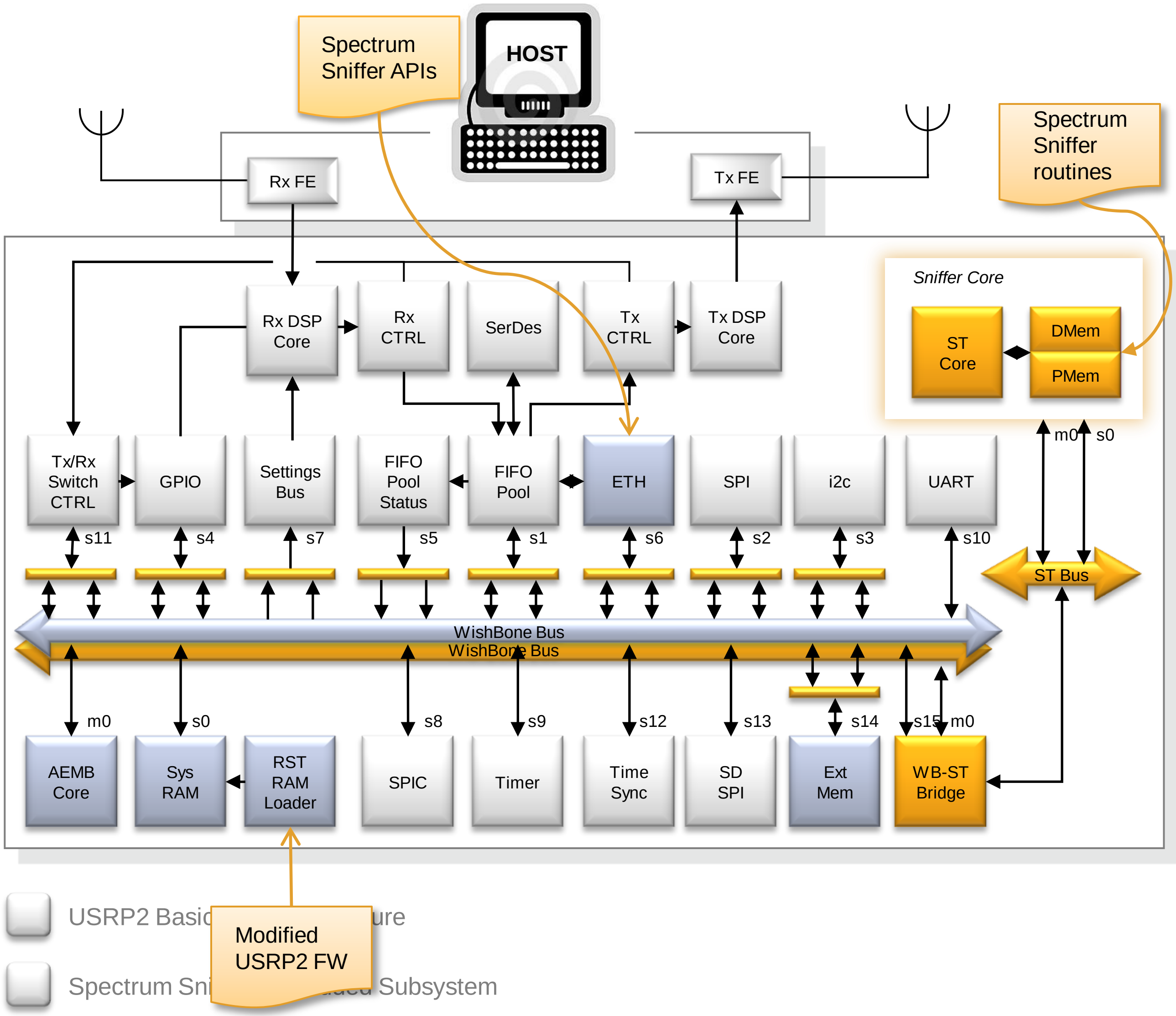
Embedded Spectrum Sniffer Architecture



Spectrum Sniffer Subsystem

- Based on a STMicroelectronics proprietary core
- Local Tightly Coupled Program and Data memory to speed up computation reducing instuction and data fetching time
- Configurable via aeMB core and via host
- Embedded in the USRP2 SoC via Wishbone Bus Lane
- Peripherals sharing via bus arbitration

Spectrum Sniffer Software Architecture



Spectrum Sniffer Software

- Modified USRP2 Firmware is loaded on boot-strap and initializes peripherals and Sniffer subsystem
- Spectrum Sniffer APIs allow to configure the Spectrum Sniffer subsystem, run the sniffing routines and getting the results from the externam memory
- Spectrum Sniffer Firmware performs the sniffing routines according to the configuration parameters passed by the host or the aeMB core and stores the results in the external memory.

Software Partitioning

Spectrum Sniffer Firmware

- Low level interface
- Takes the spectrum sniffing sweep parameters from the SoC main core or from the host processor
- Performs the spectrum sensing cyclostationary algorithm accessing data from the peripherals
- Stores the results into the external memory location according to the configuration parameters

USRP2 Firmware

- Low lever interface
- Performs the board's boot procedure
- Initializes the peripherals configuration registers
- Initializes the Spectrum Sniffer subsystem

Spectrum Sniffer APIs

- Higher level interface
- Configure the Spectrum Sniffer subsystem
- Starts a single frequency sniffing as well as a complete sweep over a given frequency band
- Get the results from the external memory to the host memory

Spectrum Sniffer Mapping on FPGA

Xilinx Spartan3 XC3S2000	Modified USRP2 SoC	
Resource	**Usage**	**%**
Registers	12808	31%
Single Port RAM RAM16X1S	16	
Double Port RAM RAM16X1D	4336	
Single Port RAM RAM64X1S	108	
ROM ROM128X1	10	
Block RAM	40	100%
Block MULT	8	20%
Clock Buffers	6	75%
Kgates equivalent	1740	87%

Critical issues

- Block RAMs are the main bottleneck for the mapped design since aeMB core and ST proprietary core have their own tightly coupled memories. Also a memory buffer pool is shared among the peripherals.
- Minimize the firmware size for the two cores
- Buffer pool resize
- Clock frequency is another critical point to be matched in the Place&Route

Results

- Demonstrator implementing a full Software Spectrum Sniffer for ISM Bands as a main part of a Cognitive Radio Physical Layer
- Fully reconfigurable, programmable multicore SoC embedding a Spectrum Sniffer subsystem

Thank you!