Description of the Ψ-chart models for the physical layer of a Zigbee (IEEE 802.15.4) transmitter

Andrea Enrici, Ludovic Apvrille, and Renaud Pacalet

Institut Mines-Telecom, Telecom ParisTech, CNRS/LTCI, Biot, France
{andrea.enrici,ludovic.apvrille,renaud.pacalet}@telecom-paristech.fr

This project is the model of our implementation for the 2.4 GHz physical layer of the Zigbee transmitter as standardized by the IEEE 802.15.4 group.

Application model

The tab "Zigbee_TX" shows the model of the data-processing algorithm that we derived from the Zigbee specifications. The block X_Source produces the data to be transmitted in the form of a flow of bits. Each incoming chunk of 4 bits is converted to 32-bits symbols by the X_Bits_to_Symbol block. The X_Chip_to_Octet block, then transforms each incoming bit into an un-signed 8-bits integer as expressed in equation 1:

$$\{0; 1\} \rightarrow \{0x00; 0x01\} \tag{1}$$

X_Chip_to_2Octet also models the separation between the even-indexed chips that are used to modulate the in-phase (I branch) carrier component and the the odd-indexed chips that are used to modulate the quadrature (Q branch) carrier component. The output is then transformed by means of a Component Wise Lookup (X_CWL block) that maps unsigned 8-bits integers to signed 16 bits integers as expressed by equation 2:

$$\{0x00; 0x01\} \rightarrow \{0xffff; 0x0001\} \tag{2}$$

At this point, the I and Q branches are processed independently by two distinct Component Wise Product (CWP) blocks: X_CWP_I and X_CWP_Q. These blocks multiply the input samples with a half-sine wave to realize the O-QPSK modulation. The quadrature shift between the I and Q branches is implemented by means of an offset between the memory addresses of the output samples. The resulting dataflow is collected by block X_Sink and is ready to be transmitted over the air. Signal processing operations are modeled as a composition of two SysML blocks: a F_block (i.e., Firing) for the operation configuration and triggering of the X_block (i.e., eXecution block).

Architecture model

The tab "Embb" is a model of a specific architecture instance of Embb, a generic baseband architecture dedicated to signal processing. The instance we used for our implementation of the Zigbee transmitter deploys three types of Processing Sub-Systems: the Front-End Processor (FEP), the Mapper, the Interleaver (INTL) and the Analog-to-Digital Interface (ADAIF).

Communication models

Tabs "CP_Memory_Copy", "CP_Single_DMA_Transfer" and "CP_Double_DMA_Transfer" are our models (Communication Patterns, CP) dedicated to capture the communication protocols and patterns that are available in the target platform (Embb). More in detail, "CP_Memory_Copy" models a non-cachable memory transfer between two memory units by means of the CPU. Tab "CP_Single_DMA_Transfer" models a DMA transfer where the transfer completion is signaled to the CPU via interrupt mechanism (no polling). Tab "CP_Double_DMA_Transfer" models a series of two DMA transfers.

Mapping model

The tab "Embb_with_Mapping" is the mapping model for our implementation. Firing tasks are all mapped on Embbs main CPU, whereas the execution tasks are mapped to Processing Sub-Systems according to their processing capabilities. Communication Patterns are mapped via a dedicated artifact where the instances of Sequence Diagrams are assigned to specific units in the architecture, and messages parameters are assigned a value for code generation.