NEONSEVEN

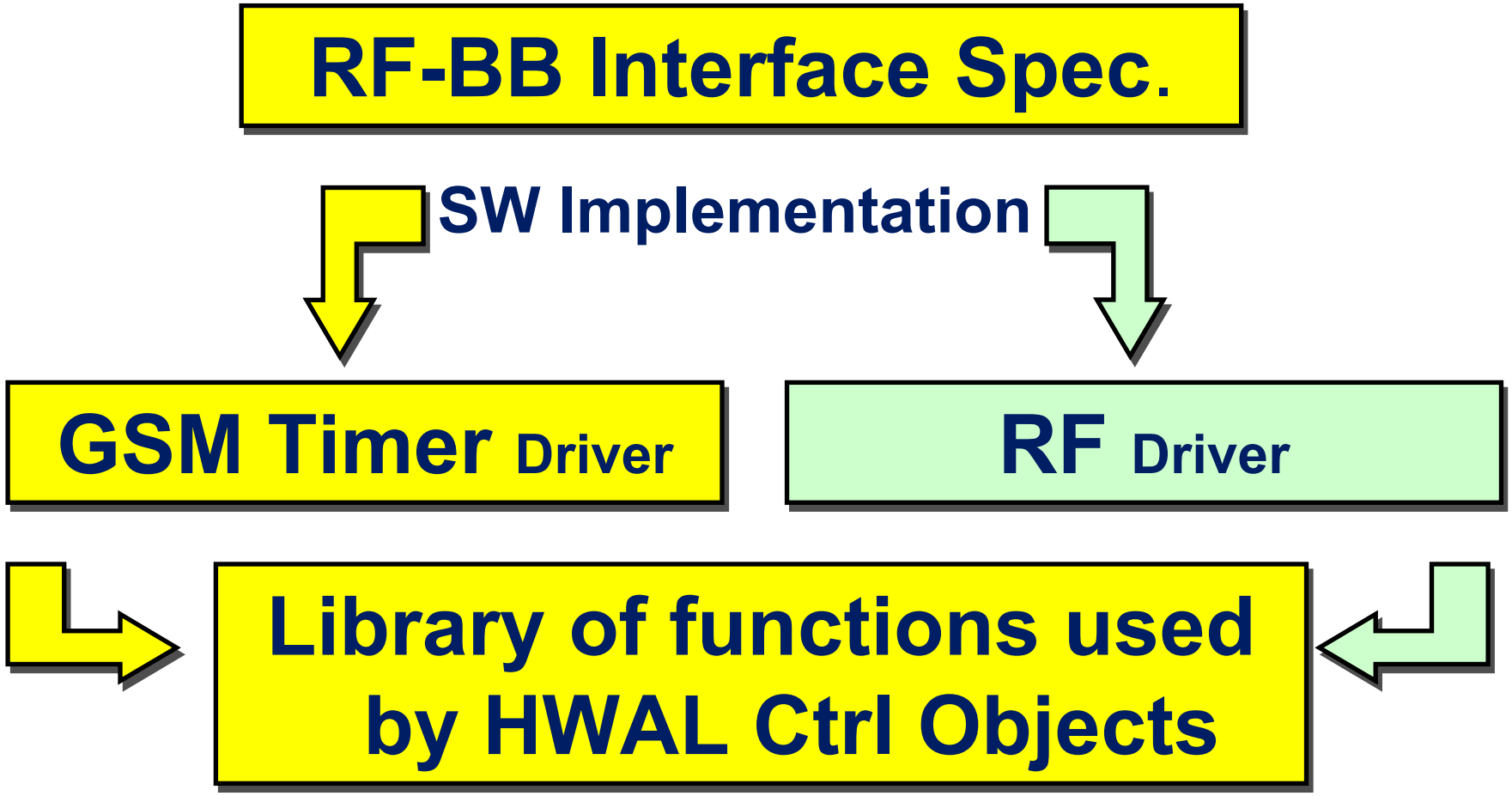
RF and GSM TIMER DRIVERS

L1 Group

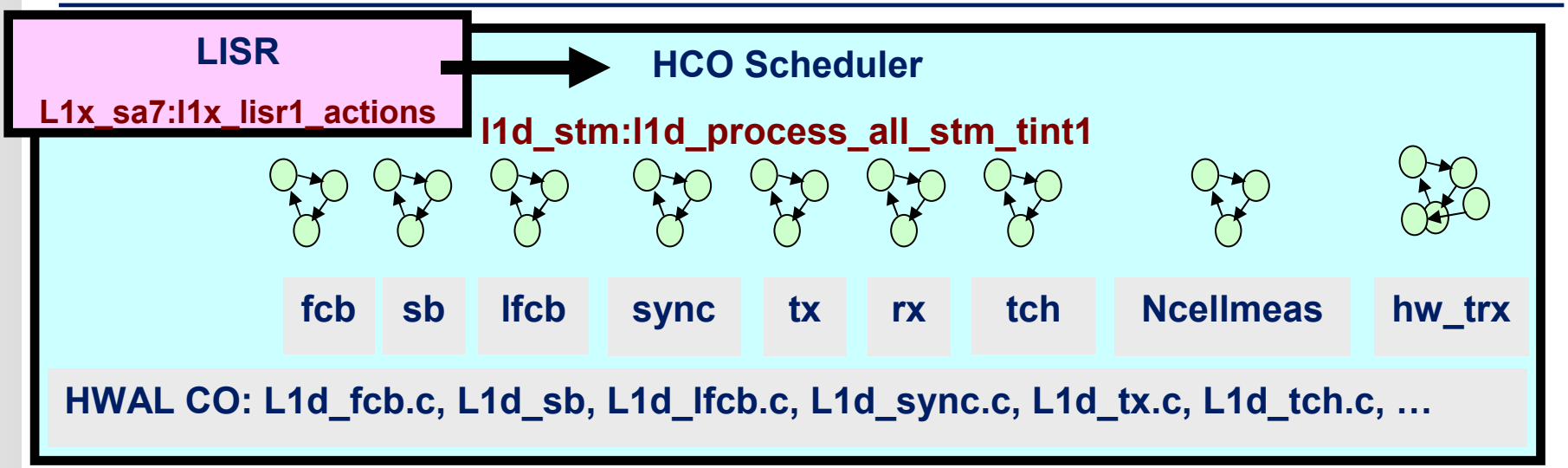
Paolo Buoncompagno

NEONSEVEN

OVERVIEW



HardWare Abstraction Layer Control Objects



Library of functions used by HWAL Ctrl Objects:

GSM Timer Driver:

gsmtu.c/h, L1d_tu.c/h, L1d_tu_init.c/h

DSP Driver:

dsp.c/h

RF Driver:

rf_ctrl.c/h, Rf.c/h, Rf_freq_table.h

NEONSEVEN

CONTROLLED UNITs and RF MODULE

RF-BB Interface

RF Driver controls:

RF Control Unit

Automatic Frequency Control Unit

GSM Timer Driver

GSM Timer Unit

RF module:

Transceiver SMARTi SD2

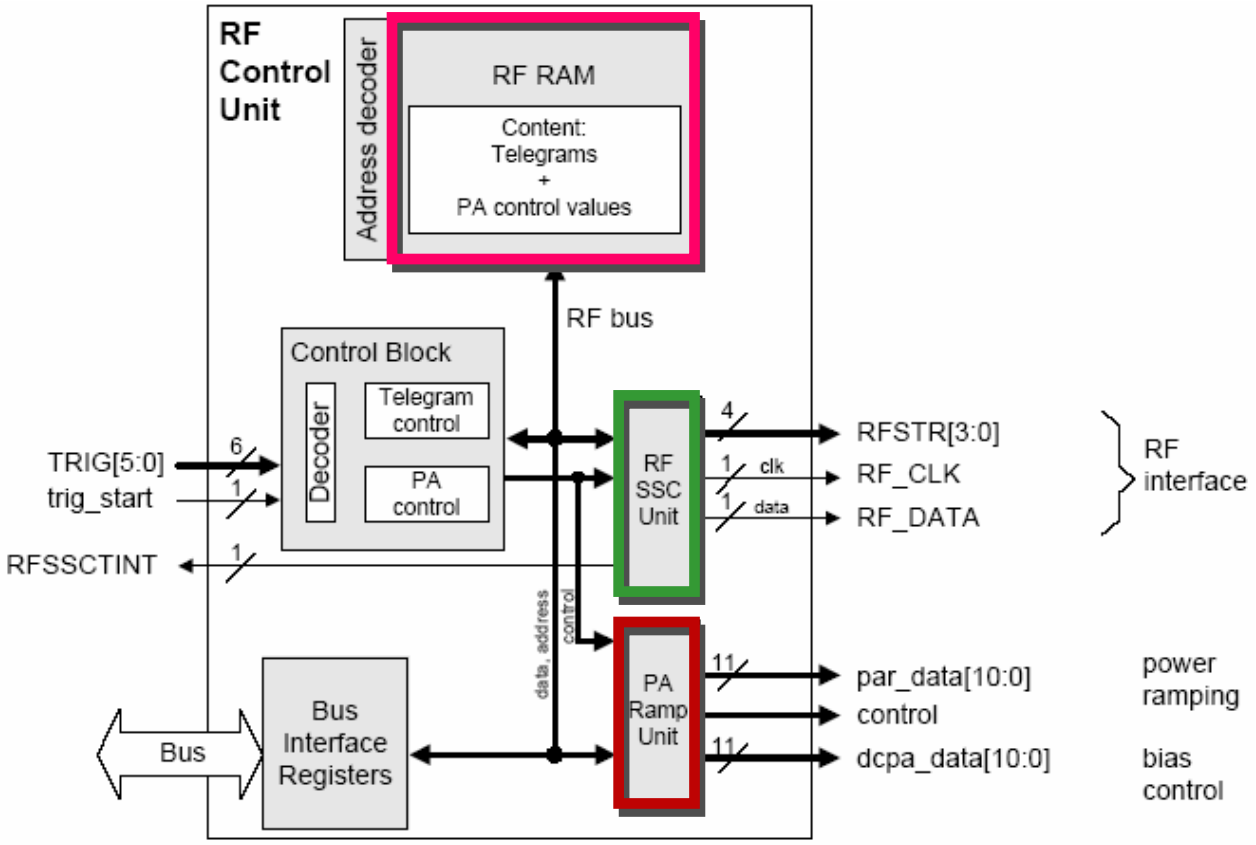
Antenna Switch

Power Amplifier

RF CONTROL UNIT

Main Sub blocks: Control Block, RF RAM, RF SSC Unit,

PA Ramp Unit, Bus Interface Register.



NEONSEVEN

RF DRIVER

- Initialization of RF
- Calculating of RF Telegrams
- Handling of Power Ramps
- AFC Programming

NEONSEVEN

INITIALIZATION OF RF

- Handled in rf_ctrl.c
- Actions:
  - **INITIALIZATION of AFC** step size and default parameters from EEPROM (AFC middle position and AFC slope values);
  - **CONFIGURATION of T_OUTx** used to control external RF devices (antenna switch and power amplifier);
  - **INITIALIZATION of RF Telegrams** (control header + init data) used to control the transceiver;
  - **INITIALIZATION of PA control values** used to perform Power Ramp up and down;
  - **ENABLE AFC**;
  - **SMARTi SD2 INITIALIZATION** transmitting in asynchronous mode the sequence of **Stealth Telegrams**.

NEONSEVEN

CALCULATING RF TELEGRAMS

- Handled in rf_ctrl.c/h and rf_freq_table.h
- Function for calculating RF telegrams are normally called from HISR level.
- Calculated RF Telegrams are stored locally and copied to RF RAM on LISR interrupt (Early prep. Int. or Frame Int.).
- Telegram types:
  - 2 Frequency Setting Telegrams used to program internal REG0 and REG 1 registers of SMARTi SD2 (Programming of the Fractional-N SIGM-ADELTA Synthesizer); used by RX/TX/MON
  - RX/TX Register to program REG2; Gain telegram for RX and MON win are programmed or TX Power Attenuation for TX win

NEONSEVEN

ADDITIONAL INIT TELEGRAMS

- THEIR NUMBER AND VALUES COULD BE DIFFERENT FOR EACH VERSION OF CHIP(CHECK YOUR VERSION).
- THE INIT TELEGRAMS SHOULD BE TRANSMITTED AFTER DCXO INIT TELEGRAMS.

NEONSEVEN

DCXO INITIALIZATION TELEGRAMS

Coefficient_Calculations.exe

α, β, γ

- 26MHZ Xtal

+ PCB LAYOUT

- XCAL = 4

default value

REGISTER PROGRAMMING

- XO_INIT1
- XO_INIT2
- XO_INIT3

- AFC = 4096

default value

- XO_TUNE

AFC and XCAL default values will be replaced by CALIBRATED VALUES

NEONSEVEN

REGISTER PROGRAMMING

The following registers are normally used for standard operation:

- **CHANNEL1** (Addr. 0x0): PLL frequency control.
- **CHANNEL2** (Addr. 0x2): PLL frequency control, RX/TX selection and Band switch and Power Mode.
- **RXTX** (Addr. 0x4): RX gain, RX common mode input voltage level, DC offset compensation (OFC).

NEONSEVEN

CALCULATION OF SYNTHESIZER PROGRAMMING

- The frequency of the internal VCO (f_{VCO}) is controlled by 31 bit of fractional channel word **CW** that is formed by a fractional part **NF** and integer part **NI**:

$$CW = NI \bullet NF \qquad f_{VCO} = CW \bullet f_{PD}$$

$$NI = floor\left(\frac{D \bullet f_{CH}}{f_{PD}}\right) \qquad \frac{NF}{MOD} = \frac{f_{CH} \bullet D - NI \bullet f_{PD}}{f_{PD}}$$

f_{CH}, f_{PD} = 26 MHz, MOD = 8388608;

D = 4 for GSM850 and GSM900 Band;

= 2 for DCS and PCS Band;

NEONSEVEN

CALCULATION OF SYNTHESIZER PROGRAMMING

f_{CH}, D

→ **NI** (8 bit), **NF** (23 bit)

→ Register Programming:

- **Channel 1**
- **Channel 2**

NEONSEVEN

HANDLING OF POWER RAMPS

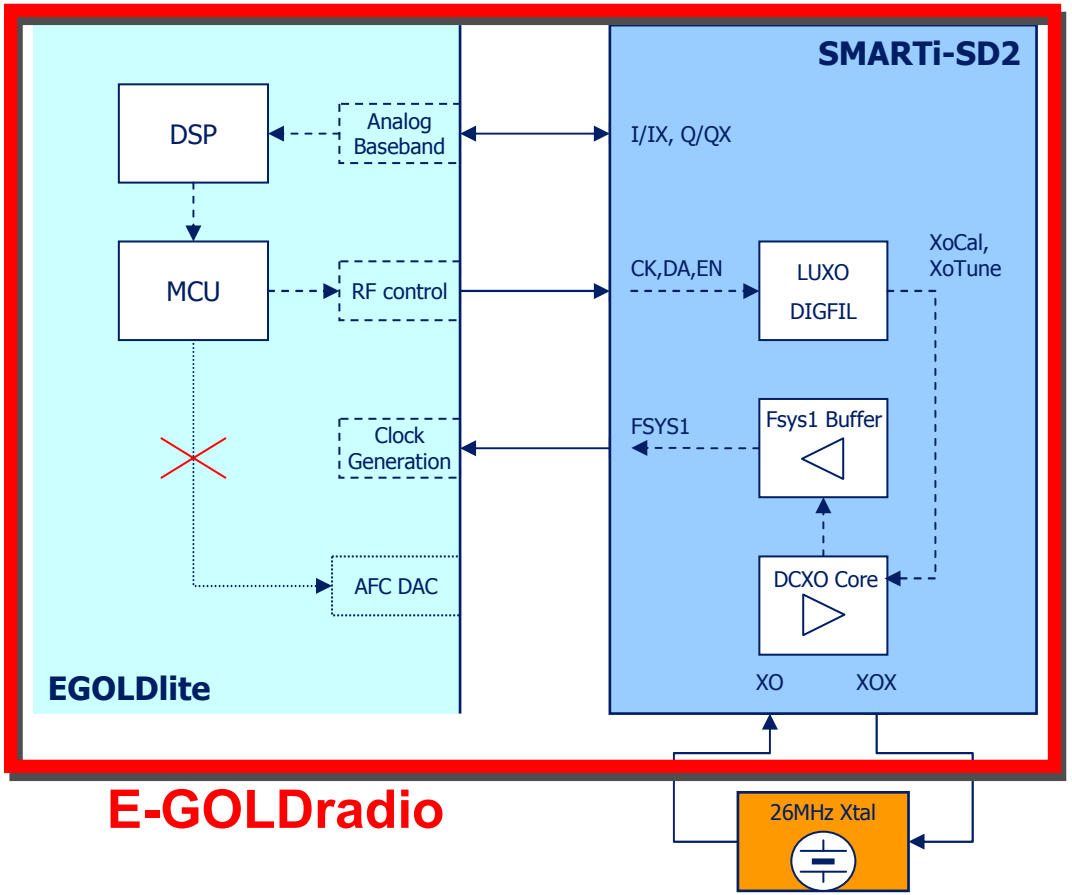
- Default PA control values are defined in rf.h:
  - One set of default ramps for each BAND
- Dynamic update of PA control values is handled in rf_ctrl.c/h:
  - Functions called from HISR stores locally new power ramp profile
  - PA control values updated in RF RAM on LISR frame interrupt

AFC CONTROL

- Handled in rf_ctrl.c
- Based on a predetermined default parameters and step size:
  - Default parameter is used during reset as the initial AFC DAC value
  - Step size (Hz/DAC step) used in calculating new AFC value
- RF driver holds the current value of AFC locally
- AFC value adjustment:
  - NEW AFC value = OLD value + frequency offset / step size
- AFC value is updated on LISR frame interrupt; XOTUNE telegram is transmitted.

AFC LOOP

- Automatic Frequency Control loop is no longer using the analog AFC line from EGOLDlite
- Instead the updated AFC value is sent through the 3-wire bus (XoTune)
- The AFC routine remains unchanged
- AFC value is scaled from 11-bit to 13-bit (0..8191)



NEONSEVEN

DETERMINE BAND

- Handled in rf_ctrl.c

- ARFCN input → BAND output

  of function l1d_rf_determine_band:

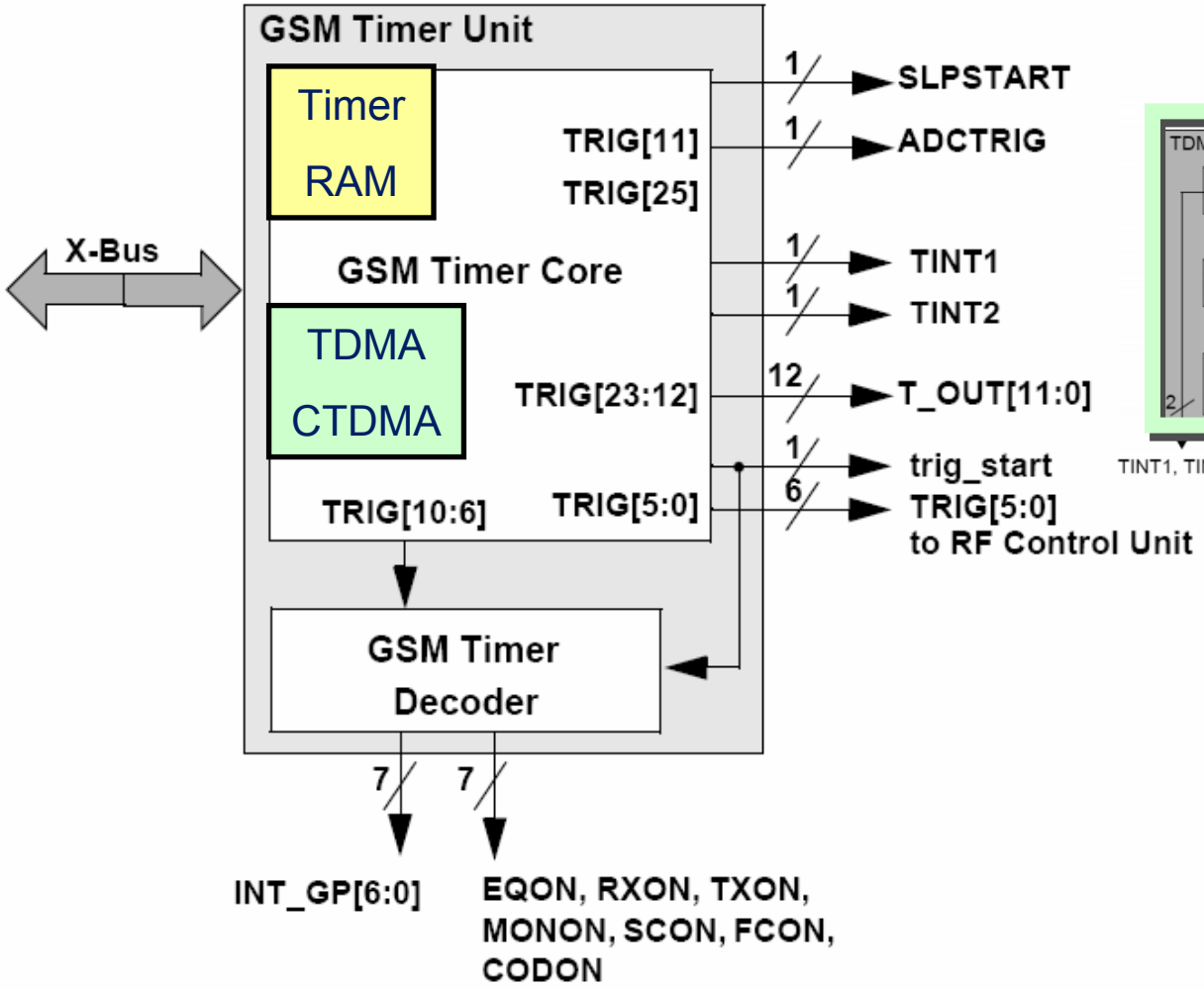
  – GSM850 : [128 - 251]

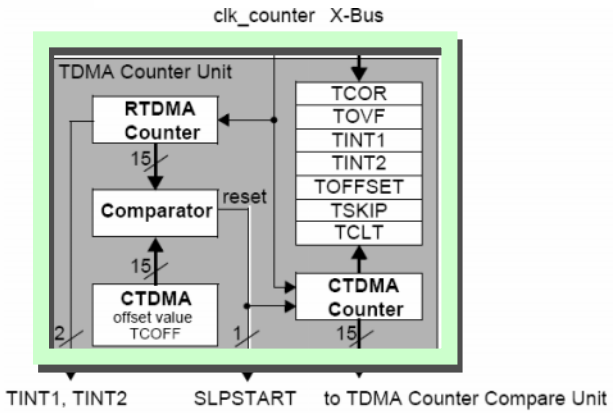
  – EGSM900 : [975 – 0 .. 1 - 124]

  – DCS1800 : [512 - 885]

  – PCS1900 : [512 - 810] – MSB is set in ARFCN to indicate 1900

GSM TIMER UNIT (GSM TU)





NEONSEVEN

Timer RAM (TRAM)

- Timer RAM could be programmed using a table of Timing Events and Timing Advance Events in order to perform all useful actions during a TDMA frame: RX-MON-TX and measurement WIN.
- Timing Event (TE) contains:
  - Timing Compare Value (TCV)
  - all programmed actions (T_OUTx set/reset, TGx transmission, starting PRx, EQON, MONON,… ,RXON and TXON signals)
- Timing Advance Event TA:
  - Timing Offset Value (TOV)
- TE and TA HAS TO BE ORDERED in the array with increasing TCV to avoid dead locks.

NEONSEVEN

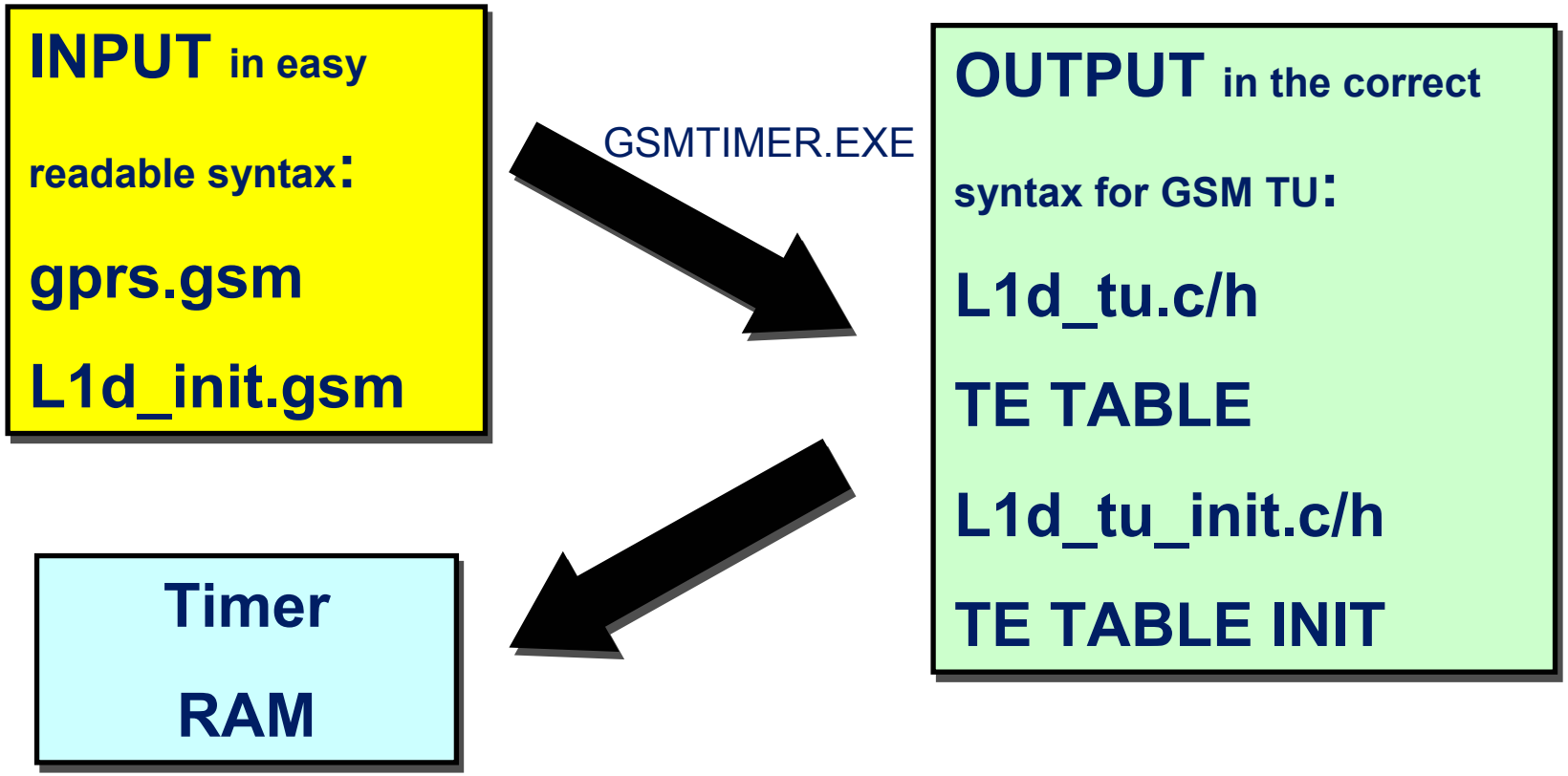
TE and TA Scheduler

- TE and TA are organized in several groups.
- Frame by frame only groups that are active will be scheduled. Activation is controlled by **TGER** register.
- An active TE is loaded from Timer RAM (TRAM): GSM TU waits until **CTDMA** = TCV + Current Timing Offset (CTO); all programmed actions will be performed and the next entry is loaded from TRAM.
- A TA is loaded: the contained TOV will be immediately applied as CTO for all follow TEs.
- The two step before is repeated starting from the first element **TEAPB** to the last one **TEAPT**. After reaching the last one GSM TU restarts from the beginning.

NEONSEVEN

TE TABLE GENERATION

- The Tool used to generate the TE table is GSMTIMER.EXE made by Application Group at IFX



NEONSEVEN

GSMTIMER.EXE TOOL

- The tool can handle different feature like:
  - OFFSET: static timing offset value applied to all TCV value of events that follow it inside TE table until next OFFSET.
  - LABEL: used to provide the index of the associated TE or TA element in the TE TABLE array; this index is used to modify TE programmed values, TCV or TA TOV during runtime.

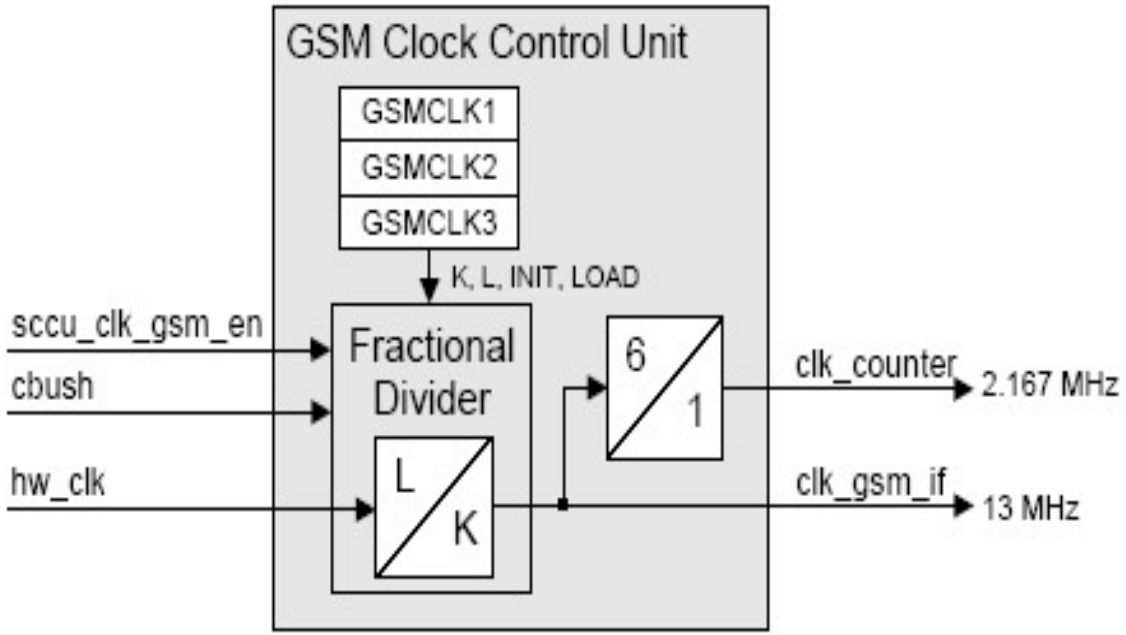
NEONSEVEN

GSM Timer Driver – gsmtu.c

- In gsmtu.c there are all functions used by HWAL CO to control GSM Timer Unit and modify the Timer RAM dynamically:
  - Enabling/Disabling Groups (TGER programming);
  - Configuring window starting time and duration (changing CTO using LABEL pointed to the correspondent TA);
  - Updating the BAND dependent T_OUTx signals (ANTENNA switch control)
- TRAM is initialized with TE TABLE INIT at the beginning to perform monitoring measurement after MPH_RXLEV_REQ
- Before first FCB-SB search TRAM is overwritten by TE TABLE
- TRAM is updated only on LISR (Early prep. Int. for RX Win or Frame Int. for TX/MON)

GSM Clock Control Unit

- Activation of new K and L values is done by setting appropriate bits in register **GSMCLK3** or by toggling the signal cbush with the bit CBUSH in the MST_CLK_CTRL register.
- This signal changes synchronously the frequency of clk_kernel or clk_bus and the ratio of the fractional divider.



NEONSEVEN

GSM Clock Control Unit

- Clock of GSM System Interface: the clock clk_bus is used to generate the clk_gsm_if with a frequency of 13 MHz for the GSM Timer Unit, the RF Control Unit and the PA Control Hardware. This clock clk_gsm_if can be programmed by setting the correspondent control reg. GSMCLK1/2T, GSMCLK1/2B and GSMCLK3. All programmed by reset values, but only GSMCLK2B needs to be programmed:

  GSMCLK2B = L_BOT_MCU_CLK_52MHZ = 0x0004;

  OR if MCU clock master frequency = 26MHz

  GSMCLK2B = L_BOT_MCU_CLK_52MHZ = 0x0002;

- Set INIT bit to load K and L value and initialize the fractional divider:

  GSMCLK3 = (1U << INIT);

Entering Power Saving

- ATTENTION: 52 to 26 MHz frequency change implies immediate reprogramming of fractional clock dividers, especially for the GSM Timer Interface to avoid to lose synchronization with BS:

  GSMCLK2B = L_BOT_MCU_CLK_26MHZ;

- Switch from 52 MHz PLL out to 26 MHz clk_in of master clock. Reset **CPUH** bit in **MST_CLK_CTRL** and set **LOAD** bit in **GSMCLK3** at the same time (use atomic instruction). This guarantees minimum skew on GSM Timer Unit.

Wake up after Power Saving

- Switch from 26 MHz clk_in to 52 MHz PLL out of master clock. Set **CPUH** bit in **MST_CLK_CTRL** and set **LOAD** bit in **GSMCLK3** at the same time, theoretically. This guarantees minimum skew on GSM Timer Unit.

NEONSEVEN

INLINE MODE

- INLINE is only used for RF module test and implemented as an annex to GSM/GPRS L1 used in normal mode.

- The RF driver used in INLINE Mode are different by those used in normal mode; the correspondent files should be aligned:
  - Rf_ctrl.c/h → inl_rf.c/h

NEONSEVEN

Thank you!