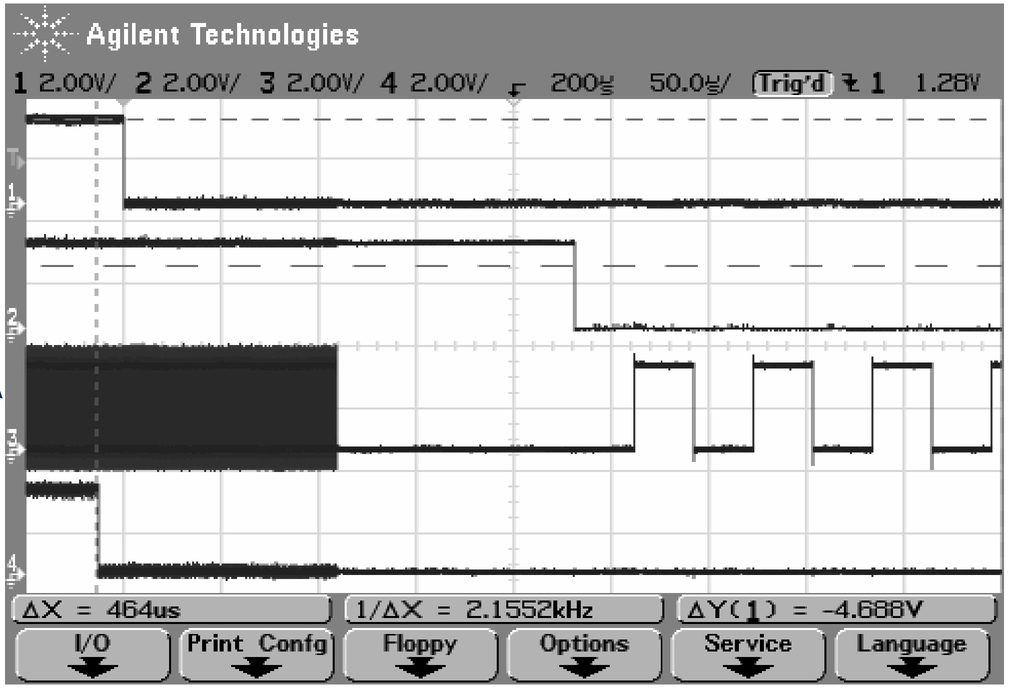
Power Saving - Overview

Software Driver 10001001 00100101 11100011		**Full running**	**Low power**
Low-Level (HW & L1)	Clock	26 MHz (nominal) VCXO (high stability)	32768 Hz (nominal) RTC clock (low stability)
	Power Supply +V	All voltages Are available	LRF0, LRF1 and LRF2 shut down
High Level (APOXI)	Peripherals	Initialized and useable	Programmed for low consumption

Power Saving – Clock

- Higher the clock, higher the power required.
- Most of the time mobile is idle in Normal Paging, CCCH has to be monitored 1% to 4% of total time, typically.
- Solution: reduce the master clock frequency while no activity is running, using 32768 Hz (A.K.A. 32 kHz or “RTC clock” or “slow clock”) and stopping 26 MHz clock (A.K.A. VCXO or “fast clock”) and derived frequencies.
- SCCU (Standby Clock Control Unit) block **automatically overviews** the clock switching operation from 26 MHz **(Care: not 52 MHz !)** to 32 kHz and back.

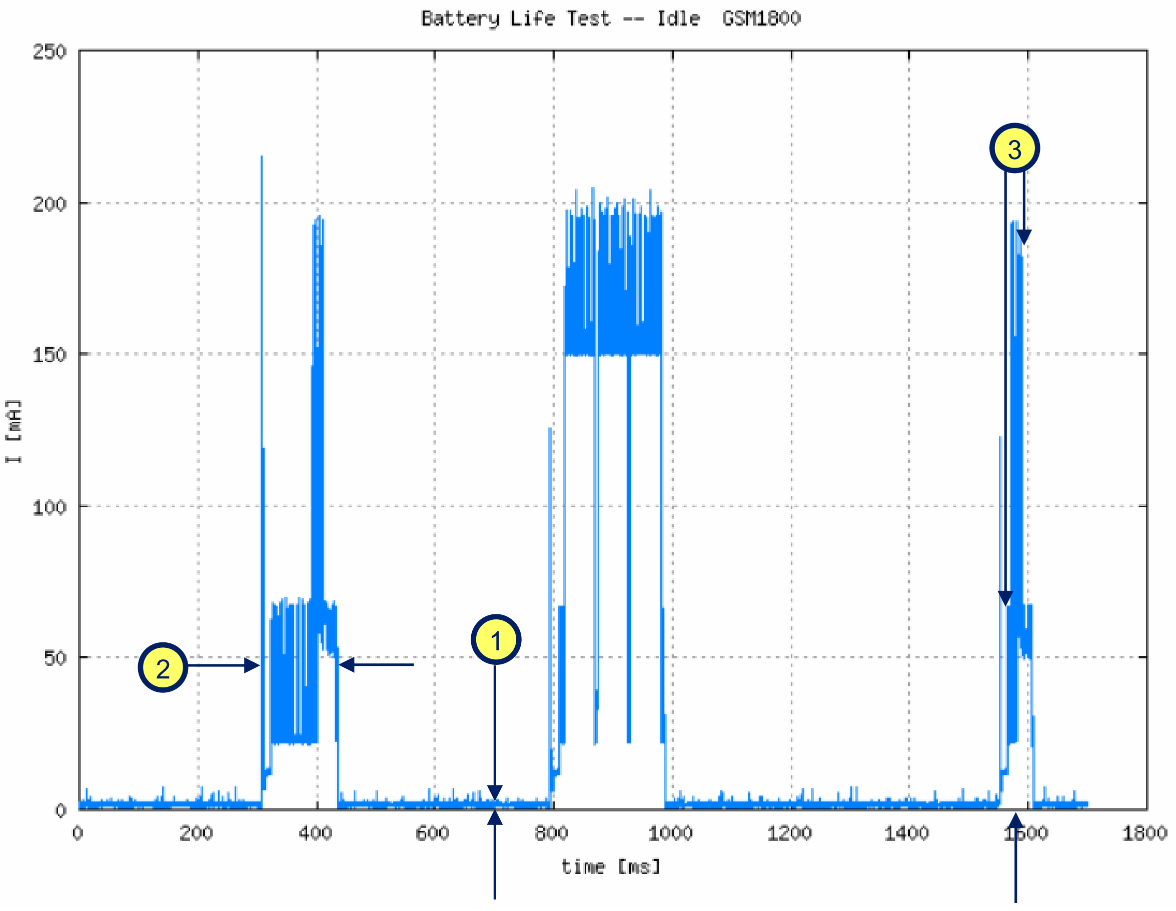
cgu_clk_master



Power Saving – Clock

Power reduction priorities

1. Reduce base (min) current consumption.
2. Minimize full-speed running periods, minimize power saving on/off switching.
3. Reduce max current consumption.



Power Saving – Clock

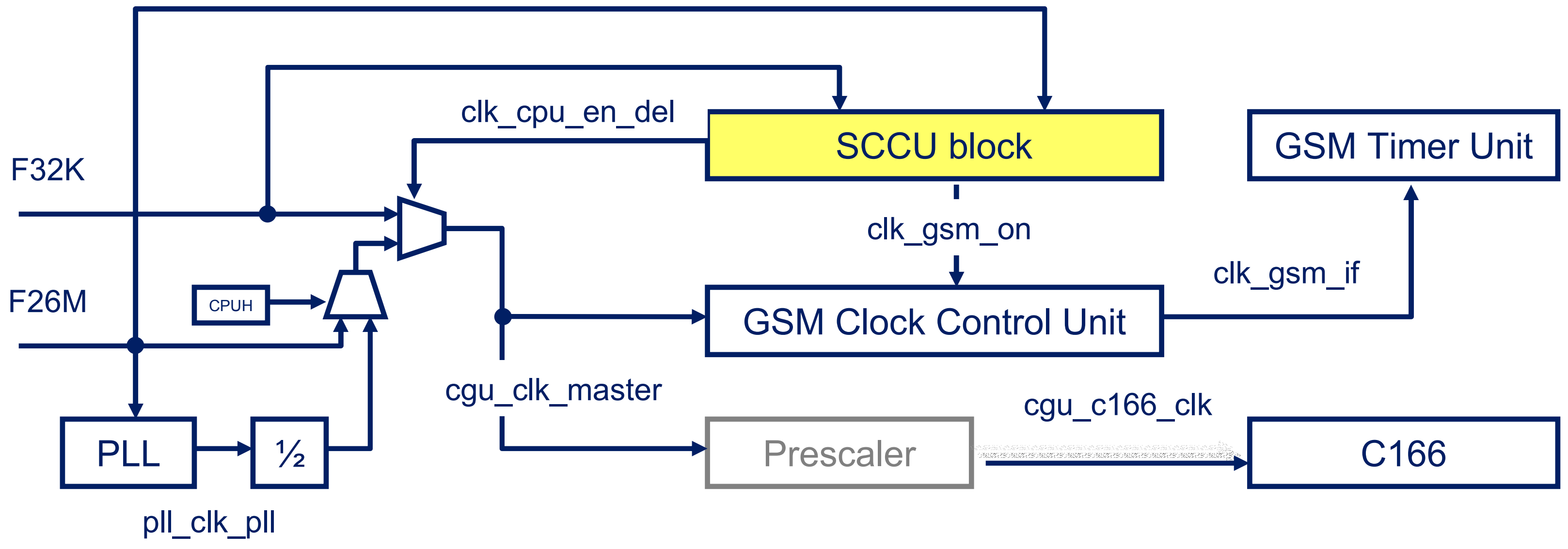
Problems arise !

- 26 Mhz VCXO power-up and run-in requires **time** (typ. 2 ms – max. 7.69 ms).
- For periodic activities (e.g. normal paging) system shall **foreseen in advance** how much time it can run slowly (A.K.A. “sleep”) since clock speed up switching requires “prewakeup”.
- The time to sleep is evaluated in **frames**. Frame Number counter is updated by software when sleep status exits (A.K.A. “awakes”). Maximum sleep period can be up to 456 frames (456 * 0.12 / 26 s = 2.105 s), minimum 1 frame only !
- System awakes immediately on **unpredictable events** (interrupts). Prewakeup delay is normally acceptable.

Power Saving – Clock

GSM Timer Unit – Run/Stop

- GSM Timer Unit is the **GSM heartbeat**. It is clocked by clk_gsm_if that may be derived from F26M (fast clock) or F32K (slow clock).
- GSM Timer Unit clock shall be **disconnected** before switching to slow clock and **reconnected** after switching to fast clock by SCCU, else it will run at slow speed, causing **MS vs BS synchronization loss**.



Power Saving – Clock

RTC Calibration – 1 of 2

- GSM Timer Unit shall be enabled/disabled with an accuracy of few octal bit (one octal bit = 1/8 of symbol bit = 12/26 µs = 0,4615 µs) to avoid loss of MS vs BS synchronization.
- To guarantee about +/-1 usec error over up to 2.105 s sleeping period (measured by RTC clock) a temporary accuracy of 0.5 ppm on RTC clock is required.
- Since RTC clock long term accuracy is roughly 100 ppm, a periodic measurement (every 1000 frames = 4.615 s) of RTC frequency referred to high accuracy VCXO clock has to be performed.
- This operation is called "RTC calibration". Care: **RTC frequency is not adjusted, really**, but just the **time difference** between 16 * NQTZ * [RTC period] and 16 frames **is simply taken into account while sleeping** .
- NQTZ is namely 151 for RTC nominal frequency of 32768 Hz. NQTZ is the integer value of RTC cycles per frame period.

Power Saving – Clock

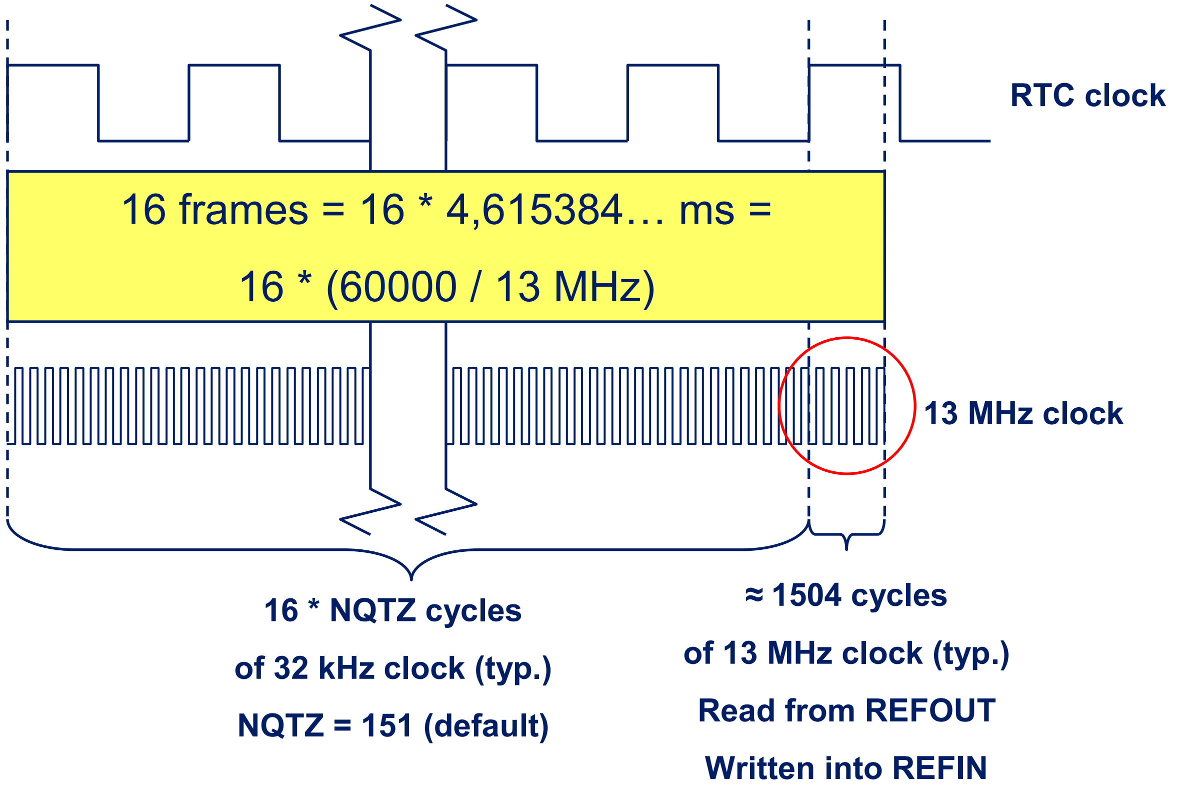
RTC Calibration – 2 of 2

- The differential time between 16 frames and 151 * 16 = 2416 RTC clock cycles is measured in 13 MHz clock periods.
- In the hypothesis of nominal RTC frequency we will found:

$$\left[\left(16\cdot\frac{0.12}{26}\right)-\left(16\cdot\frac{151}{32768}\right)\right]\cdot 13000000 = 1503{,}9 \cong 1504$$

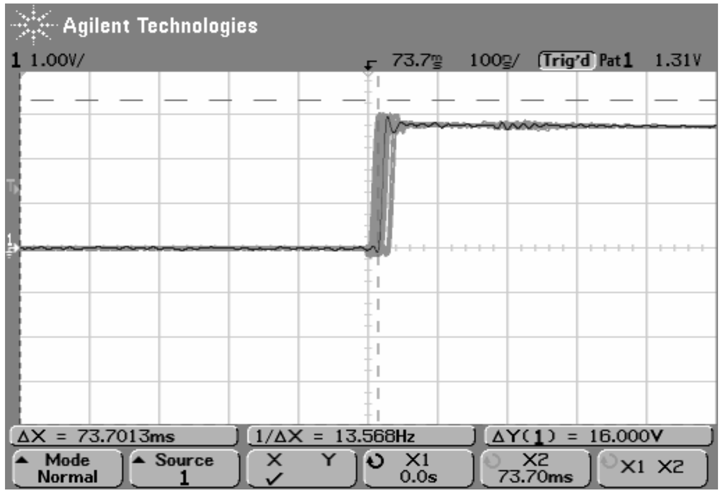
- This "calibration" value can be read from SCCUREFL and will be used automatically by SCCU to sleep µs-accurate period. It is also used to adjust hour/minute/second MMI user time.
- If 32 kHz clock is out of range (calibration failure), the system tries to count for different than 151 RTC clock strokes per frame. If also this operation fails power saving entering is aborted.

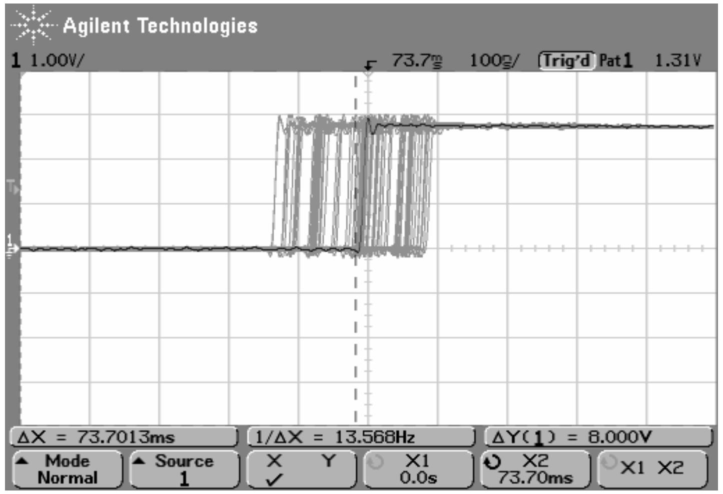
Power Saving – Clock



RTC Calibration Concept

Power Saving – Clock





The RTC jittering problem:

- RTC oscillator has very high output impedance, so it is very sensitive to external crosstalk coupled noise when EBU is active.

Solution:

- HW: Good design of RTC xtal PCB layout.
- SW: RTC calibration shall be performed only with stopped EBU activity (CPU idle waiting in ROM or IRAM).

Power Saving – Clock

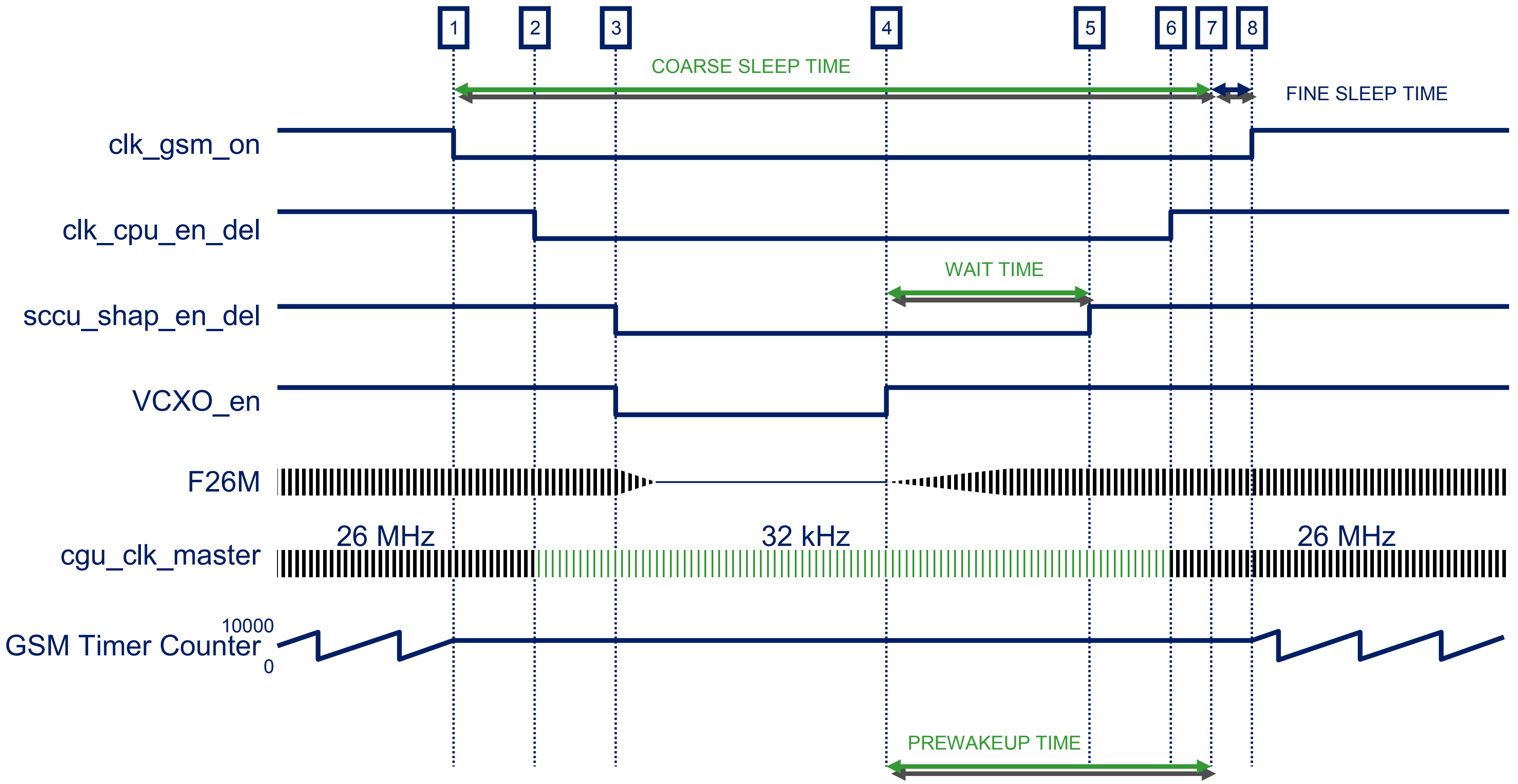
SCCU control 26 MHz ⇔ 32 kHz: Normal Sleeping Period

Integral number of frames **sleep time** is programmed in TDMAIN register. But, since a RTC clock period **granularity** is about 30 times wider than acceptable accuracy, a 2-phase (coarse/fine) time freezing is **completely managed by SCCU.** Coarse timer is based on 32 kHz, Fine timer is based on 13 MHz:

1. SCCU disconnects GSM Timer Unit and starts **coarse timer** to wait for 151 * ((TDMAIN+1) - (PREWUP+1)) RTC clock cycles.
2. 1 or 2 RTC cycles after (1), cgu_clk_master is switched **26 MHz ⇨ 32 kHz**;
3. 3 RTC cycles after (1), VCXO and input shaper are turned off;
4. When sleep timer started in (1) elapses, SCCU turns on VCXO
5. (WAIT * 4) + 1 RTC clock cycles after (4) SCCU switches on the F26M shaper;
6. 1 RTC clock cycle after (5), SCCU switches cgu_clk_master **32 kHz ⇨ 26 MHz** from stable VCXO. C166 is now running on 26 MHz (not yet 52 MHz !) **stable clock**.
7. 151 * (PREWUP+1) RTC clock cycles after (4), SCCU starts **fine timer** to wait for residual (((TDMAIN+1) * SCCUREFL) / 16) periods of 13 MHz clock
8. SCCU reconnects GSM Timer Unit perfectly in phase.

Power Saving – Clock

SCCU control: 26 MHz ⇔ 32 kHz - Sleeping period



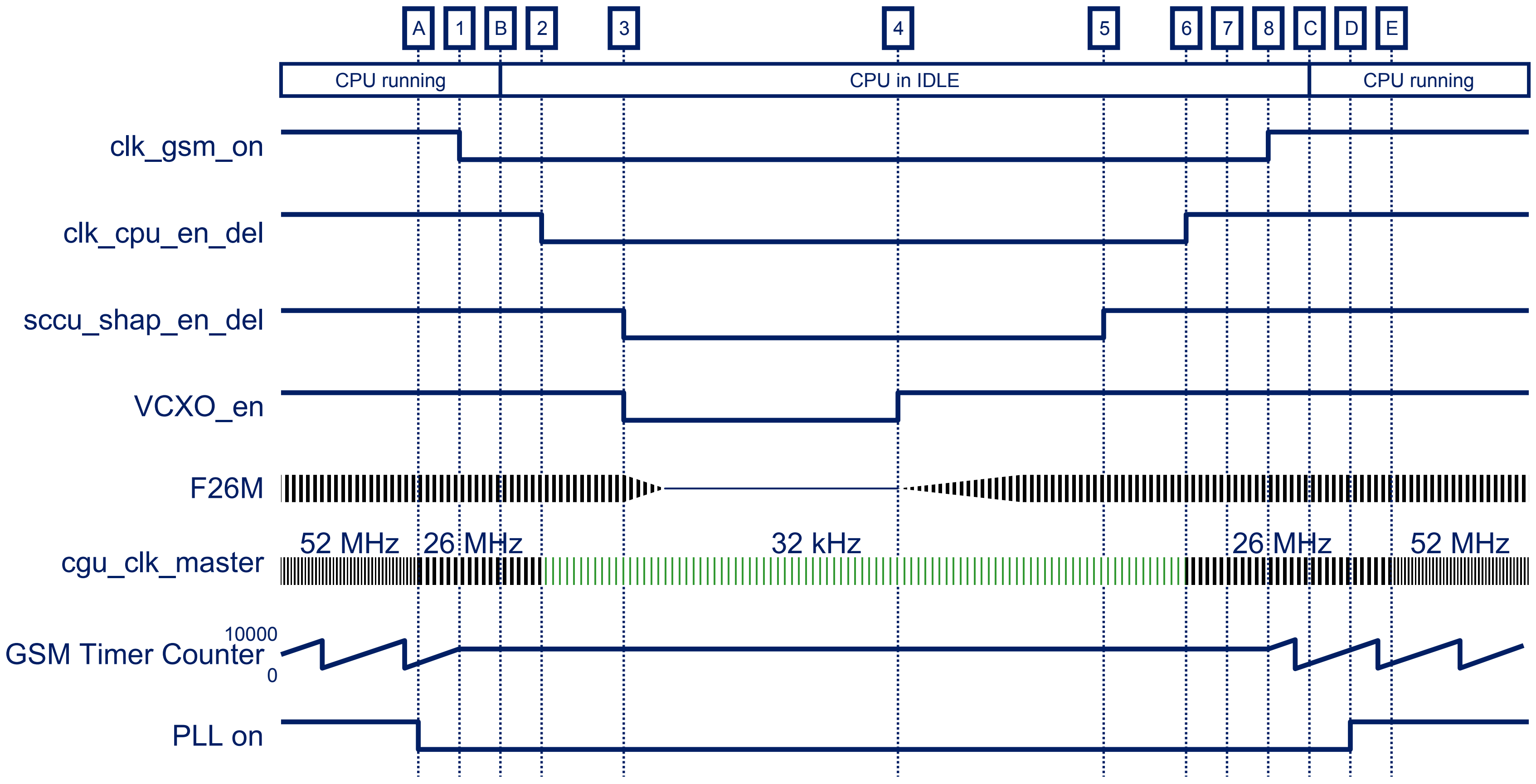
Power Saving – Clock

Software clock switching: 52 MHz ⬄ 26 MHz

- The internal PLL used to generate 104 MHz clock (used by DSP) and 52 MHz (dividing by 2, used by CPU) shall be turned off (A) **under software control**.
- After sleep mode is commanded the CPU continue its execution. Since there is nothing to do, the CPU executes IDLE instruction (B).
- If synchronous sleep was performed under SCCU control, CPU exits IDLE on the first **Frame Interrupt** (C) generated by the unfreezed GSM Timer Unit.
- The internal PLL is turned on (D) and CPU waits in idle loop for PLL be in lock.
- PLL locks within 100 µs after it was turned on (or VCXO running stable, whichever the last). Software switches cgu_clk_master from F26M to 52 MHz clock (E).

Power Saving – Clock

CPU control: 52 MHz ⇔ 26 MHz (⇔ 32 kHz) - Sleeping period



Power Saving – Clock

Clock switching: Notes

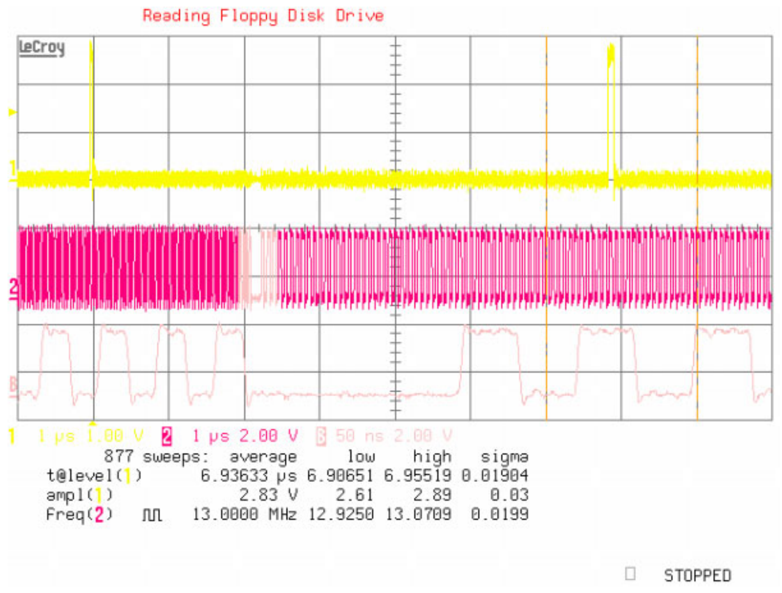
- When IDLE instruction is executed the CPU remove clock from itself, but leave it to Interrupt Controller to answer to any incoming interrupt. If interrupt arrives the CPU continue execution after IDLE instruction.
- IDLE instruction is executed from internal SRAM and surrounded by few instructions that put, then restore, Address & Data bus pulled to ground. This reduces FLASH & RAM combo power consumption to a minimum.
- Since only the 26 MHz clock, not 52 MHz clock, is automatically available when system awakes, the GSM Clock Control Unit shall be programmed to divide cgu_clk_master by 2, rather than by 4 (and cgu_clk_master shall be switched from 52 MHz to 26 MHz then PLL can be turned off) before enter sleep phase.
- When software switches cgu_clk_master from 26 MHz to 52 MHz clock it shall **at the same time** reprogram GSM Clock Control Unit for a division by 4, rather 2. This makes GSM Timer Unit running **always** at 13 MHz.

Power Saving – Clock

Software clock switching: 52 MHz ⇔ 26 MHz

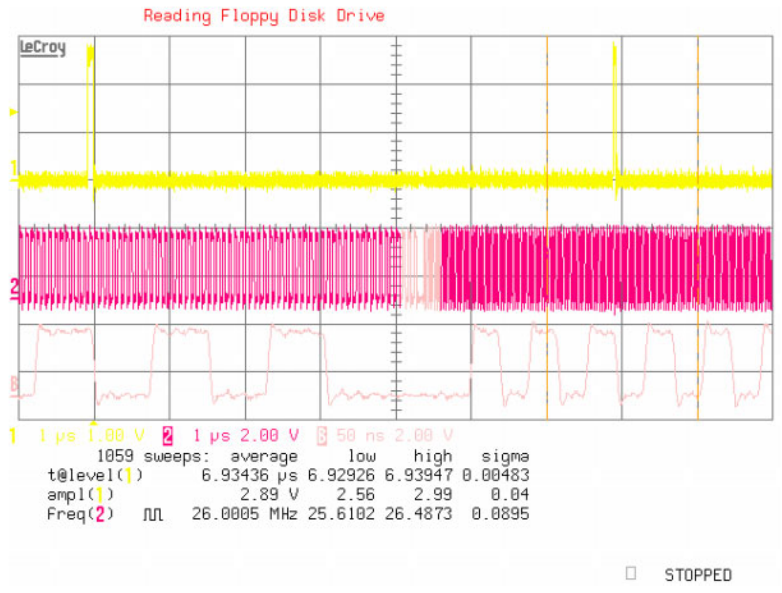
Scope of (A)

Scope of (E)



```
1   1 µs 1.00 V   2   1 µs 2.00 V   B  50 ns 2.00 V
        877 sweeps:   average      low     high    sigma
 t@level(1)        6.93633 µs 6.90651 6.95519 0.01904
 ampl(1)                2.83 V    2.61    2.89    0.03
 Freq(2)    ⎍⎍    13.0000 MHz 12.9250 13.0709   0.0199
                                                STOPPED
```

Figure 1 - 52 to 26 MHz clock switching (sleep enter phase)



```
1   1 µs 1.00 V   2   1 µs 2.00 V   B  50 ns 2.00 V
       1059 sweeps:   average      low     high    sigma
 t@level(1)        6.93436 µs 6.92926 6.93947 0.00483
 ampl(1)                2.89 V    2.56    2.99    0.04
 Freq(2)    ⎍⎍    26.0005 MHz 25.6102 26.4873   0.0895
                                                STOPPED
```

Figure 2 - 26 to 52 MHz clock switching (sleep exiting phase)

Power Saving – Clock

SCCU Control: Early Termination

- Sleeping phase can be also terminated **before** the normal SCCU frame timing sequence elapses. This can happen due to 2 possible reasons:

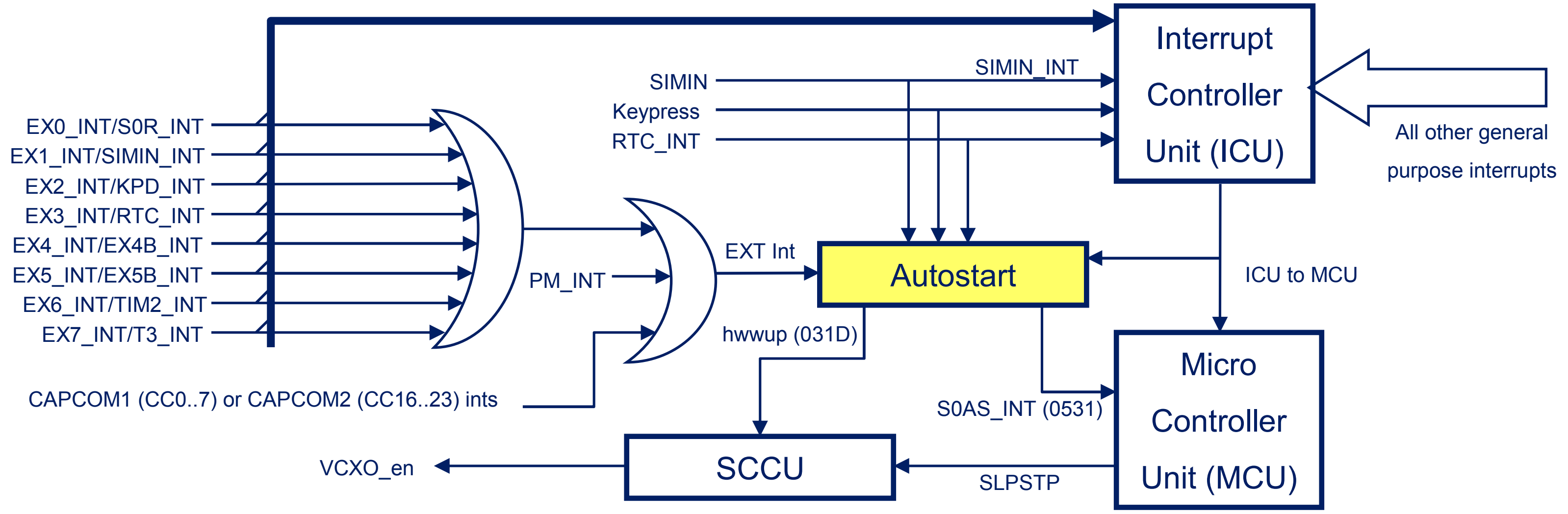
1. `hwwup` hardware signal from **autostart block**. This block handles various possible hardware interrupt sources (RTC, Keypad, any Fast External Interrupt or any general Interrupt Controller priority-solved interrupt). Selective masking is possible using SCCUHWWAKEUPH and SCCUHWWAKEUPL registers.
2. Software (running at 32 kHz) writes to SLPSTP bit in SCCUSLPCTRL control register. Even if this operation is normally not required, it represents an additional security wakeup in LISR codes.

- Each FEXTn_INT source is the logical selection / and-or-combination of 2 possible sources: main or alternate (EXISEL register).
- FEXTn_INT interrupt sources are edge sensitive: n=0,1,2,3,6,7 are rising-edge only, N=4,5 are both rising and falling edge sensitive (EXICON register).
- On SCCU early termination the VCXO_en is risen up immediately. After WAIT time elapsed, and cgu_clk_master is running at 26 MHz, GSM Timer Unit is connected perfectly in phase on the next useful frame.

Power Saving – Clock

SCCU Control: Early Termination

Edge	Main source – FEX_INT_A	Alternate source – FEX_INT_B
↑	EX0_INT (0544) – Alt 2 of pad E12 (KP0) – Used as KEYOUT0	S0R_INT (051E) –Interrupt on RXD of ASC0
↑	EX1_INT (0545) – Alt 1 of pad C11 (RXD) – Used as RXD_0	SIMIN_INT (0543) of Sim Card Interface
↑	EX2_INT (0546) – Alt 2 of pad M7 (RSTOUT_n) – Used as KEYOUT1	KPD_INT (052F) of Keypad Interface
↑	EX3_INT (0547) – Alt 2 of pad P2 (SSC1_MRST) – Used as CAM_EOF	RTC_INT (051B) of RTC block
↑↓	EX4_INT (0548) – Alt 1 of pad C5 (CS1_n) – Used as CS1_N	EX4B_INT (0551) – Alt 1 of pad P7 (I2S2_TX) – Used as HS_DET
↑↓	EX5_INT (0549) – Alt 1 of pad E13 (KP6) – Used as KEYIN0	EX5B_INT (0552) – Alt 1 of pad N1 (CC00IO) – Used as MA3_IRQ_VINT
↑	EX6_INT (054A) – Alt 1 of pad P5 (I2S2_CLK0) – FLIP_SENSE (to gnd)	TIM2_INT (0536) - INT_GP(2) from GSM Timer Unit
↑	EX7_INT (054B) – Alt 1 of pad P8 (T_OUT5) – Used as ACC_PWR_EN	T3_INT (0514) – GPT1 timer 3

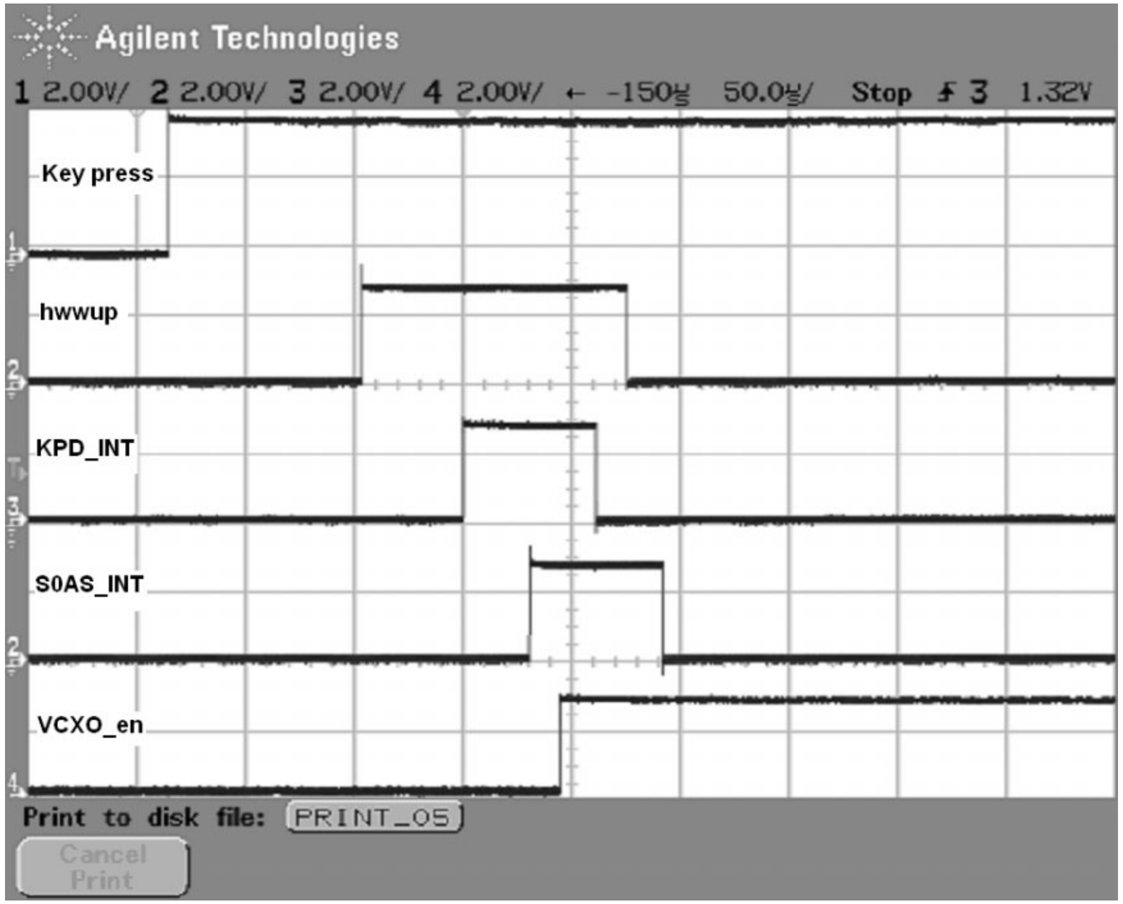


Power Saving – Clock

Autostart block function – 32 kHz wakeup on keypress

- Ch.1 – Keypress event
- Ch.2a – hwwup signal (031D)
- Ch.3 – KPD_INT signal (052F)
- Ch.2b – S0AS_INT signal (0531)
- Ch.4 – VCXO_en signal

ICU to MCU is disabled (not useful)

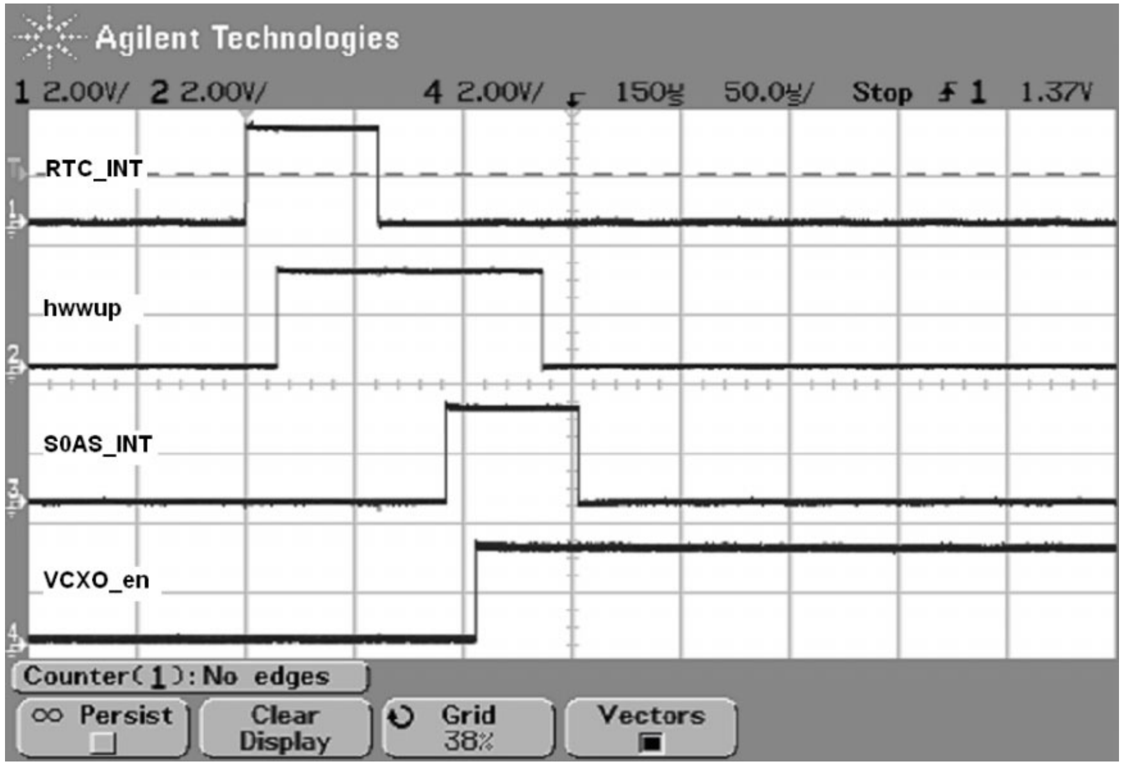

Agilent Technologies
1 2.00V/ 2 2.00V/ 3 2.00V/ 4 2.00V/ ← -150µs 50.0µs/ Stop ⨍ 3 1.32V
Key press
hwwup
KPD_INT
S0AS_INT
VCXO_en
Print to disk file: PRINT_05
Cancel Print

Note: monitor signals within parentheses

Power Saving – Clock

Autostart block function – 32 kHz wakeup on RTC interrupt

- Ch.1 – RTC_INT signal (051B)
- Ch.2 – hwwup signal (031D)
- Ch.3 – S0AS_INT signal (0531)
- Ch.4 – VCXO_en signal

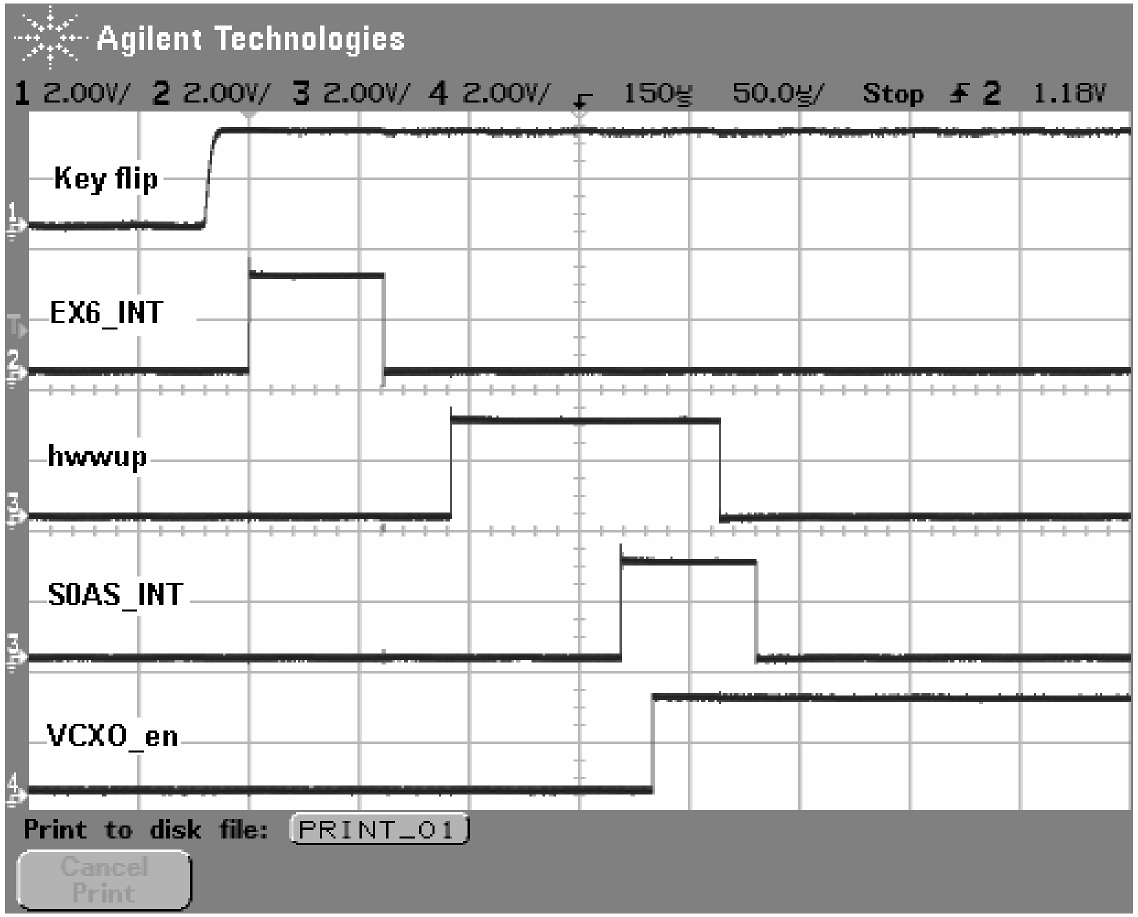


Note: monitor signals within parentheses

Power Saving – Clock

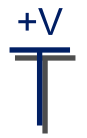
Autostart block function – 32 kHz wakeup on key flip opening

- Ch.1 – Flip open event
- Ch.2 – EX6_INT signal (054A)
- Ch.3a – hwwup signal (031D)
- Ch.3b – S0AS_INT signal (0531)
- Ch.4 – VCXO_en signal


Agilent Technologies
1 2.00V/ 2 2.00V/ 3 2.00V/ 4 2.00V/ 150µs 50.0µs/ Stop 2 1.18V
Key flip
EX6_INT
hwwup
S0AS_INT
VCXO_en
Print to disk file: PRINT_01
Cancel Print

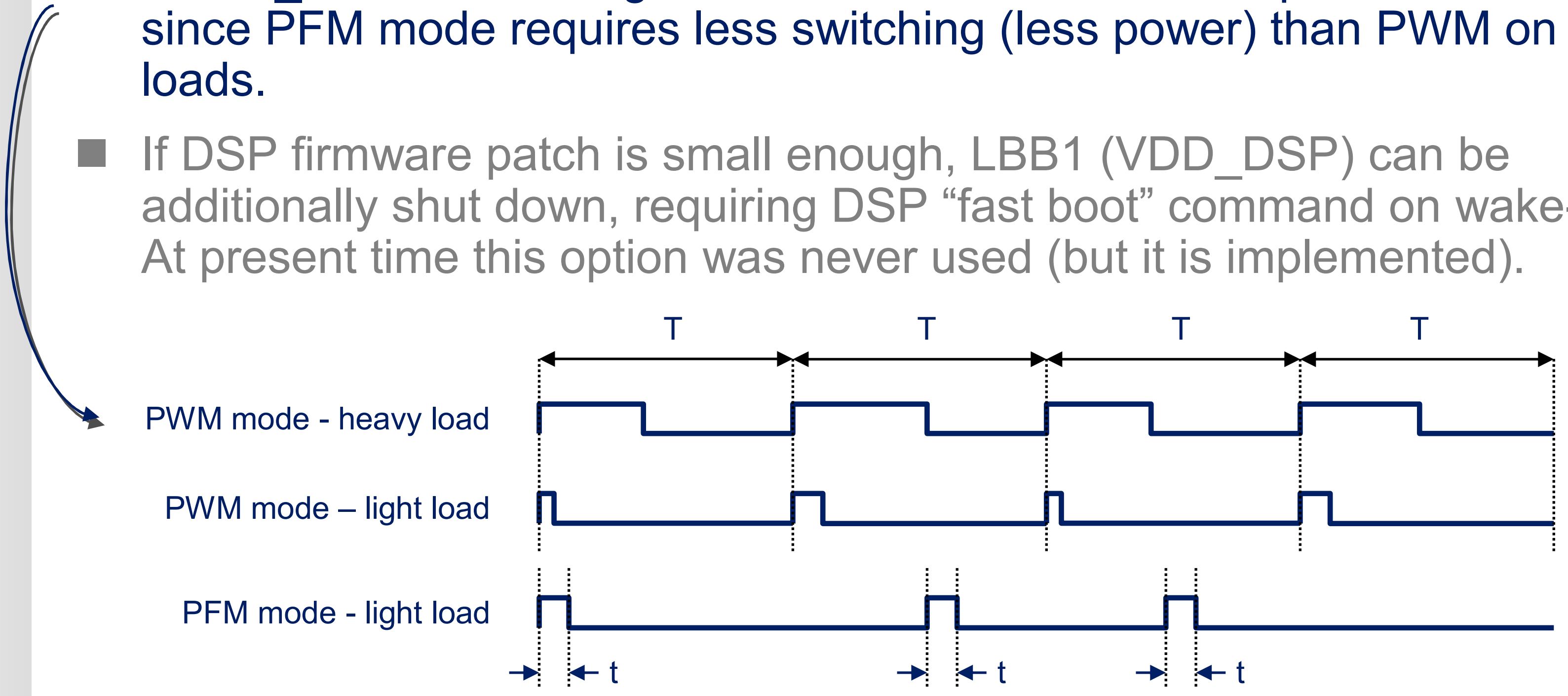
Note: monitor signals within parentheses

Power Saving – Power Supply



E-POWERlite

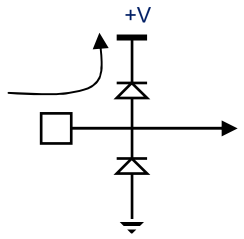
- Less the voltage, Less the power.
- VCXO_en control voltage is applied to E-POWERlite to selectively shutdown supply voltage lines LRF0, LRF1 and LRF2 for radio part.
- VCXO_en control voltage is used in PWM ⇔ PFM StepDown converter, since PFM mode requires less switching (less power) than PWM on light loads.
- If DSP firmware patch is small enough, LBB1 (VDD_DSP) can be additionally shut down, requiring DSP “fast boot” command on wake-up. At present time this option was never used (but it is implemented).

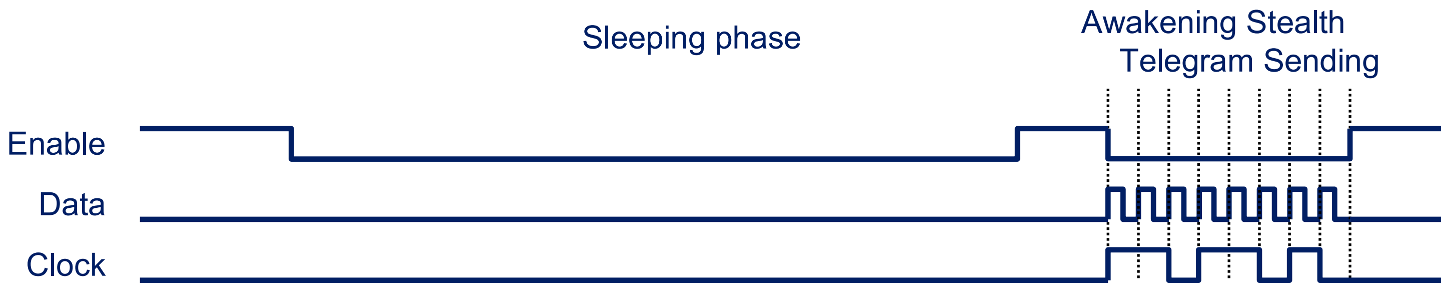


Power Saving – Power Supply

SMARTi 3-wire serial bus

- SMARTi and BaseBand are linked via a 3-wire bus (Clock, Data, Enable)
- Since Enable is active low (and inactive high) it is forced low before enter power saving, else SMARTi will be pseudo supplied via Enable **input junctions**. **<u>This may be very harmful for relevant BaseBand output port HW and consumes power</u>**.
- Enable is restored high when power saving exits. Moreover, SMARTi has also to be reinitialized, just like the first power up (stealth telegrams).





Power Saving – Power Supply

+V

Other Supplies

- Led supply needs DC-DC switched capacitor step-up converter, since blue leds require more than 3 V supply. This converter is very simple but **inefficient** (only half of the energy is stored in caps - 5 mA typ. current without output load) so it **shall be shut down** from APOXI calls if no light is required at all.
- Low led brightness (PWM of 2% or so) draws more current than expected since fast clock is needed to drive PWM, and 32 kHz system clock is inhibited.
- E-POWERlite built-in **audio amplifier** shall also shutted down when unused via proper I^2C messages.
- External audio amplifier (option) may need additional low-drop regulator for FM-Radio (option) and another DC-DC converter for balanced voltage to got strong audio level. Also in this case high-level commanded shutdown is highly recommended.

Power Saving – Peripherals

Peripherals Shutdown and Restart Operations

- Blanked LCD saves power at its best. LCD black-and-white low-detailed **small-bandwidth** image (large numbers, for example) can help to **reduce** power consumption by a factor of 2.
- So-called “Screensaver” effects are power-wasting since CPU animation activity support is required and 200 ms software timer (frame-aligned) is required continuously to run. **End user shall be warned on this side effect**.
- Camera shutdown programming should also include camera clock swinging removal from relevant output pin (CLKOUT0 – pad M1).
- All General Purpose I/O (GPIOs) shall be initialized to proper direction / output logic level A.S.A.P. If supply is removed form external device, relevant GPIOs should be placed at low logic level, or decoupled.

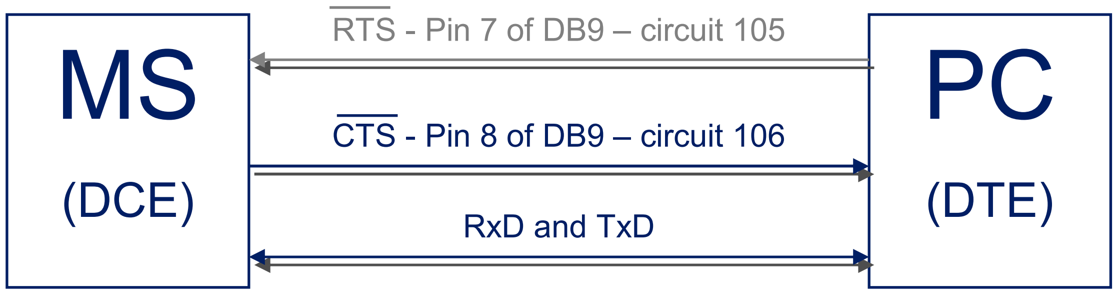
For more information please refer to Vojko PAHOR and Stefano MARCHESIN.

Power Saving – Peripherals

Serial Ports and Power Saving

- It is **impossible to receive** and send on Asynchronous Serial Communication (ASC) ports at standard speeds in power saving.
- DCE serial port relies on **hardware flow control** logic. DTE shall cache serial data flow if it senses CTS inactive (high logic). CTS is set inactive while power saving and set active while running at full speed and ready to receive. Since power saving exits at least every 2 s, DTE shall be capable to cache up to 2 s of information flow (host polling mode). System waits on a software timer for 5000 frames (23.077 s) after the last valid character received, allowing immediate response, before re-enter power saving.
- DCE may also sense DTE's RTS inactive-to-active driven **transition** to generate external interrupt that forces power saving exit. DCE Serial port will be ready to receive only after a couple of frames. Is up to the DTE application to correctly manage handshake lines. **This mode of operation is actually disabled**.
- Hardware flow control is currently supported **only on ASC0**, not on ASC1.

Power Saving – Peripherals


$\overline{\text{RTS}}$ - Pin 7 of DB9 – circuit 105
MS
(DCE)
$\overline{\text{CTS}}$ - Pin 8 of DB9 – circuit 106
PC
(DTE)
RxD and TxD

Serial port handling with power saving

- PC shall cache data to be transmitted under hardware flow control if power saving is not disabled.
- “Polled mode” is used: when power saving exits (e.g. for normal paging activity) CTS is made active (logic low level = +V_{RS232})
- Interrupt initiated by RTS can also be implemented if requested for faster response.

Power Saving – Peripherals

DSP StandBy Power Down (SBPD)

- DSP shall execute IDLE instruction before power saving. During sleep phase, DSP and analog supply domain are **kept in reset** state and decoupled from BaseBand core. This allows to significantly to reduce VDD_MAIN current.
- DSP shall be restarted after a reset. If VDD_DSP was not removed firmware patches are still available: simply BRANCH to boot code start address and call `l1d_modu_init` function to reinitialize GMSK modulator parameters once for all.
- If firmware patch size is less than 1334 words **VDD_DSP** can also be **removed**. Additional PW_DOWN command is used to save firmware in all the Shared Memory (not lost). When wake-up, FAST boot command restores P-RAM of DSP in only 77 µs. <u>This feature was never tested since firmware patch is actually too large</u>.
- Additionally, **VDD_ANA** can also be **removed** if VDD_DSP is removed. Take this into account while distributing supplies in hardware project.

Power Saving – Software

L1d_dsp.c Driver - Public Functions

- l1d_psv_setup_clock – is responsible of overall clock setups at very first startup.
- l1d_psv_init – is responsible of SCCU block once-for-all presets.
- l1d_psv_sleep – provides to enter power saving on specified number of frames. It supports complete 52 MHz to 32 kHz switch down process. It is called, with interrupt disabled, from l1c_pwr_sleep_mode_handler to enter power saving mode.
- l1d_psv_idle_loop – executes CPU IDLE instruction waiting for any interrupt. This code executes from internal SRAM with VISIBLE bit at 0.
- l1d_psv_wake_up – supports fast system clock switch from 32 kHz to 26 MHz. Shall be called A.S.A.P. within any LISR requiring immediate service. It uses l1d_psv_enable_fast_clock.

Power Saving – Software

L1d_dsp.c Driver – Public Functions (cont)

- l1d_psv_tick – called from framer LISR provides, via psv_wake_up private function, to complete the wake-up process. It also controls the RTC calibration process.
- l1d_psv_disable –disables power saving to be entered by setting selectively 1 specified bit over possible 32 on a bit mask. If the bit mask is non-zero, power saving is inhibited.
- l1d_psv_enable – Resets specified bit in bit mask, possibly re-enabling power saving operation.

L1d_dsp.c Driver – Private Functions

- psv_wake_up – called from l1d_psv_tick, provides for the complete wake-up process up to PLL turn-on and normal 52 MHz system clock running.
- l1d_psv_set_rf_strobe_to_high_active – handles SMARTi on sleeping
- l1d_psv_set_rf_strobe_to_low_active – handles SMARTi on awakening

Power Saving – Software

L1c_pwr.c Driver – Public Functions

- l1c_pwr_sleep_mode_handler – provides to overall manage power saving possible duration and to eventually use these periods for EEP data saving. If power saving is possible it is started. Then CPU IDLE is executed. It also provides for ASC0 power save function calls. Called from deep assembler OSE idle_loop.
- l1c_pwr_calc_sleep_time – provides to evaluate the maximum number of frames that can be slept. If power saving was not disabled, a poll through Channel Objects (CO) is performed and the minimum allowed sleeping frame value is evaluated.

Actually no public entry point function is provided to put in sleep and awake devices other than SMARTi. Power saving clock switching is mainly associated to Channel Objects scheduled activity, indeed.