ATC 2008

MSP430 Advanced Technical Conference

Using the flexible ‘5xx Universal Clock System (UCS)

Stefan Schauer

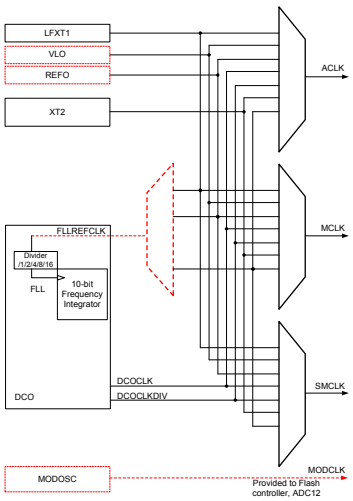
Texas Instruments

Agenda

- Introduction into the UCS system
- Oscillators (Overview, Characteristics, typical usage)
- Frequency Locked Loop (FLL)
- Low Power Mode support from the UCS
- Summary

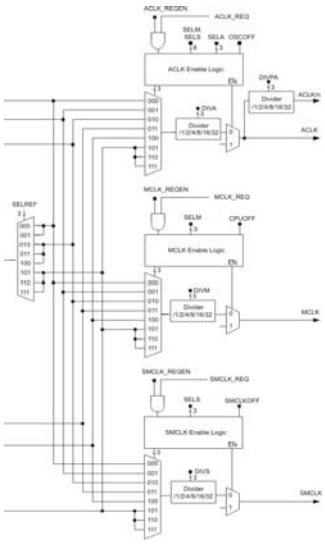
F5xx Unified Clock System (UCS)

- Three low-freq sources
  - LFXT1
  - VLO
  - REFO
- FLL reference selectable from LFXT1, REFO, or XT2
- ACLK/SMCLK/MCLK can all be driven from any source
- MODOSC provided to modules
  - Example: Flash controller and ADC
- PLL for USB devices only
  - Up-converts 4-24MHz XT1/2 to internal 48MHz for USB communication



UCS Comparison to 4xx and 1xx

- Using best of FLL and Basic Clock
  - FLL
  - High clock flexibility
- Existing clocks essentially unchanged
  - XT1/XT2/DCO/FLL
- Increased clock orthogonality
  - Any source can drive any system clock
- Crystal pins muxed with I/O function, defaulting as I/O
  - Must be initialized to crystal function
- Clock divider on all clock tree outputs



Low Frequency Clock Sources

Range of choices to fit application needs

	POWER	*PRECISION*	*COST*
XTAL	1uA	HIGH	COMPONENT
REFO	3uA	MEDIUM	ZERO
VLO	<500nA	LOW	ZERO

(Current included in Active and LPM0-3 current if clock is used for ACLK)

High Frequency Clock Sources

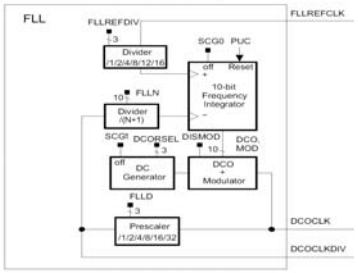
Range of choices to fit application needs

	POWER	*PRECISION*	*COST*
XTAL (XT1)	60uA @ 12MHz 150uA @ 20MHz 300uA @ 32MHz	HIGH	COMPONENT
XTAL (XT2)	60uA @ 12MHz 150uA @ 20MHz 300uA @ 32MHz	HIGH	COMPONENT
DCO	60uA @ 1MHz	Depends on Ref + Jitter	ZERO

(DCO Current included in Active and LPM0 current)

5xx FLL Overview

- FLL: Adjust DCO frequency in reference to a lower clock source (similar to PLL)
- Normally the FLL is used as source for the MCLK (CPU)
- Very flexible scaling of the output frequency
- Sources for Reference: REFO / LFXT1/XT1 / XT2
- Output frequency: 100kHz - >32Mhz



Frequency Locked Loop (FLL) functionality

- The System Clock of controllers has to meet different requirements, according to the application and system conditions:
  - High frequency, to react fast onto system hardware requests or events
  - Low frequency, to minimize current consumption, EMI, ...
  - Stable frequency for timer applications e.g. real time clock RTC
  - Low-Q oscillators to enable start-stop operation with 'zero' delay to operation.
- All these conflicting but essential requests can not be handled, with
  - high-Q, fast frequency crystals
  - low-Q RC-type oscillators
- Lowest current consumption and frequency stability require the use of a low frequency crystal.
- The compromise used in the MSP430 is to use a low frequency crystal, and to multiply its frequency up to the nominal operating range.

Agenda

- Introduction into the UCS system
- Oscillators (Overview, Characteristics, typical usage)
- Frequency Locked Loop (FLL)
- Low Power Mode support from the UCS
- Summary

Oscillators

- LF oscillator
- HF oscillator
- VLO
- Reference Oscillator
- Start up sequence

LF oscillator

- Support for 32kHz Crystal
- Startup time < 1000ms
- Internal Load Caps for Crystal:
  2pF, 5.5pF, 8.5pF, 12pF (effective)
- Adjustment of drive strength (0- 3)
  - Default: highest setting for highest safety factor
- Oscillator Allowance:
  - 210 kOhm at 6pF (Drive Strength: 0 / Safety Factor: 5)
  - 300 kOhm at 12pF (Drive Strength: 1 / Safety Factor: 5)
- Separate fault flag for LFXT Oscillator
- Many improvement for stability have been added compared to older families
- Bypass mode to feed in external digital clock

HF oscillator

- Support for 4 -32 MHz Crystal
- Startup time < 10ms (6MHz Crystal)
- Oscillator Allowance:
  450 Ohm at 6MHz
  320 Ohm at 12MHz
  200 Ohm at 20MHz
  200 Ohm at 32MHz
- No internal Load Caps for Crystal:
  (add 1pF from Bond Pads)
- Adjustment of drive strength (default highest setting)
- Bypass mode to feed in external clock
- XT1 and XT2 identical

VLO (Very LP/LF Oscillator)

- Very low-power, low-cost alternative for 32kHz crystal in apps that don't require precision
- Power draw figures are included in $I_{LPM3, VLO}$
- Introduced on 2xx

Internal Very-Low-Power Low-Frequency Oscillator (VLO)

over recommended ranges of supply voltage and operating free-air temperature (unless otherwise noted)

PARAMETER		TEST CONDITIONS	V_{CC}	MIN	TYP	MAX	UNIT
f_{VLO}	VLO frequency	Measured at ACLK	1.8 V to 3.6 V	4	12	20	kHz
df_{VLO}/d_T	VLO frequency temperature drift	Measured at ACLK(1)	1.8 V to 3.6 V		0.5		%/°C
df_{VLO}/dV_{CC}	VLO frequency supply voltage drift	Measured at ACLK(2)	1.8 V to 3.6 V		4		%/V
Duty cycle		Measured at ACLK	1.8 V to 3.6 V	40	50	60	%

(1) Calculated using the box method: (MAX(-40...85°C) - MIN(-40...85°C))/MIN(85°C - (-40°C))
(2) Calculated using the box method: (MAX(1.8...3.6V) - MIN(1.8...3.6V))/MIN(1.8...3.6V)/(3.6V - 1.8V)

Reference Oscillator

- Factory calibrated Oscillator
- Accuracy sufficient for UART Communication (up to 9600 Baud)
- Current Higher then LF Oscillator
- alternative to 32kHz crystal
- Moderate frequency tolerance over voltage/temp
  - Similar to DCO, much better than VLO
  - Less accurate than 32kHz crystal
- Power draw is higher than crystal or VLO
- Is the default FLL reference clock

Internal Reference, Low-Frequency Oscillator (REFO)

over recommended ranges of supply voltage and operating free-air temperature (unless otherwise noted)

PARAMETER		TEST CONDITIONS	V_{CC}	MIN	TYP	MAX	UNIT
I_{REFO}	REFO oscillator current consumption	T_A = 25°C	1.8 V to 3.6 V		3		µA
f_{REFO}	REFO frequency calibrated	Measured at ACLK	1.8 V to 3.6 V		32768		Hz
	REFO absolute tolerance calibrated		1.8 V to 3.6 V			±3.5	%
		T_A = 25°C	3 V		±TBD		
Duty cycle		Measured at ACLK	1.8 V to 3.6 V	40	50	60	%
t_{START}	REFO startup time	40%/60% duty cycle	1.8 V to 3.6 V		0.4		ms

What Can You Do With REFO?

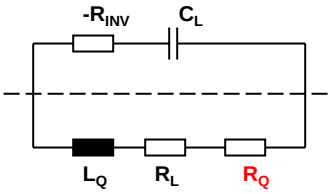
- Periodic wakeup for apps in which these are true…
  - Don't need crystal accuracy…
  - But need better accuracy than VLO
  - More cost-sensitive than power-sensitive
- Can you do RTC?
  - Not really -- +/-2% error means ~ +/- 1/2 hour error every day
  - But not bad as a 'walking wounded' RTC mode in event of crystal failure!

MODOSC

- Internal oscillator to help automate operation of some modules
  - Substitute for source clock in Flash module
    No configuration of f_{FTG} required
    No Risk of bad programming due to wrong Flash clock
  - Serves as ADC12_A's internal oscillator (ADC12OSC)
- ~ 5MHz
- Not available to system clocks – direct to modules
- Generally for applications in which drift isn't critical
- Activation on demand
  - Flash activates it automatically when programming or erasing
  - ADC12 activates it when chosen as conversion clock

Oscillator Allowance

- Load Capacitance CL contains C1, C2 and CS
- The amplification capability of the oscillator inverter is replaced with a negative resistance –RINV
- The quartz crystal is replaced by the load resonance resistance RL (effective resistance) and the effective Reactance LQ.



Condition for oscillation:

$|-R_{INV}| = R_L + R_{Qmax}$

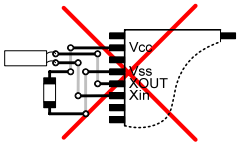
Safety Factor:

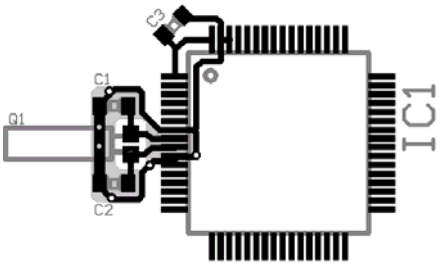
$$SF = \frac{R_{Qmax}}{R_{Lmax}}$$

Safety Factor	Qualification
SF < 1.5	unsuitable
1.5 ≤ SF < 2	risky
2 ≤ SF < 3	suitable
3 ≤ SF < 5	safe
SF ≥ 5	very safe

Crystal Layout

- Crystal as close the to MSP430 as possible
- Short and direct traces, no traces underneath
- Keep away switching signals
- Ground crystal can, use guard ring around leads
- Ground plane underneath crystal





Oscillator startup sequence

```
void LFXT_START(void)
{
  P7SEL |= 0x03;              // enable XT1 for LFXTAL
  UCSCTL6_L |= XT1DRIVE1_L+XT1DRIVE0;
              // Highest drive setting for XT1 startup

  while (SFRIFG1 & OFIFG) {   // check OFIFG fault flag
    while (SFRIFG1 & OFIFG) { // check OFIFG fault flag
      UCSCTL7 &= ~(DCOFFG+XT1LFOFFG+XT1HFOFFG+XT2OFFG);
                              // Clear OSC fault Flags
      SFRIFG1 &= ~OFIFG;      // Clear OFIFG fault flag
    }
  }
}
```

- Ports have to be enabled for Oscillator usage (PxSEL)
- Select the required drive strength and load XCAP (default is highest)
- No Software delay loop is required anymore (done in Hardware)

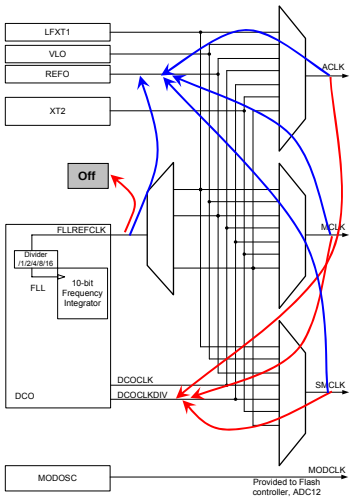
Oscillator Fault / Fail-Safe Modes

- Fault detection (XT1, DCO, XT2)
  - Flag set if oscillator enabled but not operating properly
  - Cristal Oscillator Clocks will switch to save backup clock
- Flags must be reset by software: Not Automatic!
- Fail-safe modes ensure minimal operation if primary clock source fails
- For MCLK/SMCLK/ACLK:
  - If LFXT1 is selected and it fails: reverts to REFO
  - If HFXT1/XT2 is selected and it fails: reverts to DCO
- During an oscillator fault, DCOCLK active even at lowest DCO tap, to provide clock for the CPU

Oscillator Fail-Safes vs 2xx/4xx

- Similar to 2xx/4xx, except….
- If LF crystal fails, REFO now takes over
  - In 2xx/4xx, DCO takes over (only for WDT+)
  - Robust, but large freq difference can affect operation
- REFO and crystal have same nominal frequency, allowing similar functionality
- Remember:
  - REFO tolerance isn't as tight – not a replacement for crystal in all cases
  - Current draw is higher than crystal -- 3uA (typ)

Oscillator failsafe - Backup clocks



Original Source: LFXT1

Original Source: Other

Oscillator Fault Handling

- Write NMI oscillator fault handlers for robustness!
- Fault detection outputs are “flags” and therefore latched
  - LFOF, DCOF are now XT1LFOFFG, DCOFFG, etc.
  - Specific OF Flags feed into OFIFG, which is also latched (as it was in 2xx/4xx)
  - Source flags must be cleared manually
  - Difference to 2xx/4xx: Self-cleared when condition ceased, and were not called “flags”
- OFIE no longer automatically cleared (Nested NMI interrupts of same level are not accepted by hardware)

Agenda

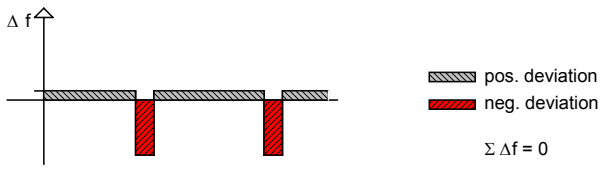
- Introduction into the UCS system
- Oscillators (Overview, Characteristics, typical usage)
- Frequency Locked Loop (FLL)
- Low Power Mode support from the UCS
- Summary

FLL

- Understanding a FLL , difference to an PLL
- FLL: Regulation and Modulation
- Clock Accuracy (cycle by cycle, average, stability)
- Setting for certain Clock frequency
- Understanding the error of an FLL

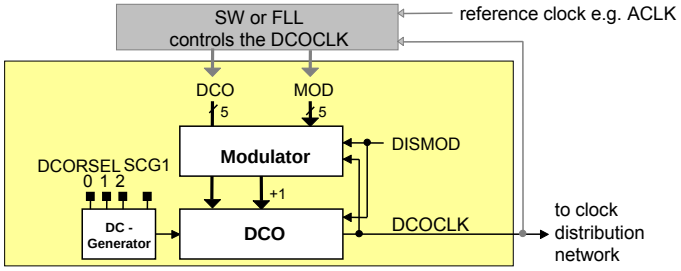
Understanding an FLL

- The FLL aligns the frequency (and phase) of the DCO to the low-frequency clock, in order to provide increased stability and determinability of the frequency.
- The FLL operates as a continuous frequency integrator. An up/down counter that follows the loop control corrects permanently the multiplication factor N. The follow-up or up-date rate is identically to the crystal's frequency rate.
  Using a 32,768 kHz crystal the rate is 30.5µs.
- The accumulated frequency error is the same as that of the crystal's. The time deviation from one machine cycle to another is typically less than 10%.

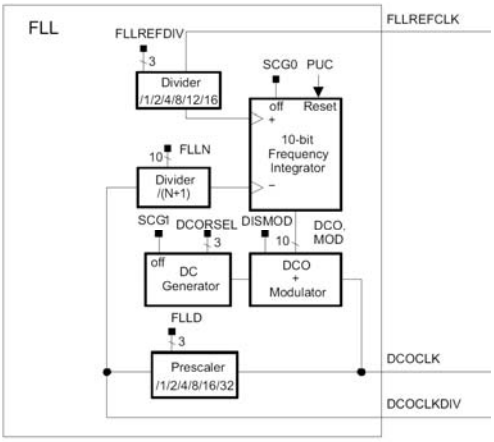


Digital Controlled Oscillator

- The operating range is controlled by:
  - DCORSEL0…DCORSEL2
  - Wide Range Area
- Digital Controlled Oscillator is controlled by
  - DCO0…DCO5
  - Frequency Tap for fine adjustment
- Five modulation bits MOD0 to MOD4 to define the timing interval



FLL: Blockdiagram



Note: The SCG0 bit in the Status Register (SR) controls the FLL loop (open or closed).

FLL: Digital Oscillator Loop Control

The content of N defines the system frequency if the FLL is active.

$$f_{(DCOCLK)} = D \times (N + 1) \times f_{(FLLRefClock)} / n$$

- D: FLL Loop Divider in UCSCTL2 (FLLD bits)
- N: Multiplier Bits in UCSCTL2 (FLLN bits) (must be greater then 0)
- n: DCO tap selection in UCSCTL 0 (Modified automatically by the FLL)

FLL: Selection of Nominal Frequency

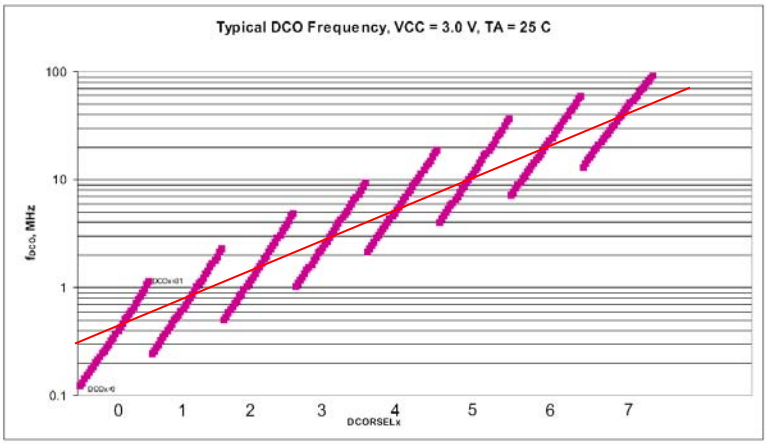
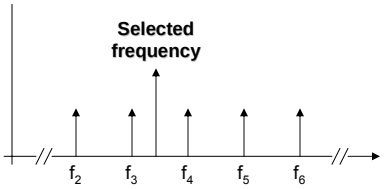


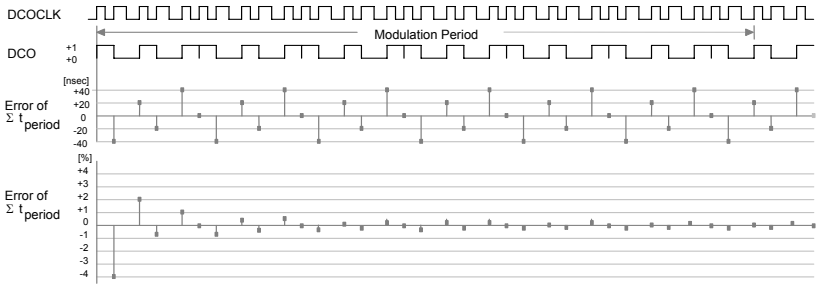
Figure 10. Typical DCO Frequency

FLL: Modulation

	Frequency	Cycle time
selected:	1000 kHz	1000 nsec
f3:	943 kHz	1060 nsec
f4:	1042 kHz	960 nsec

MOD = 19

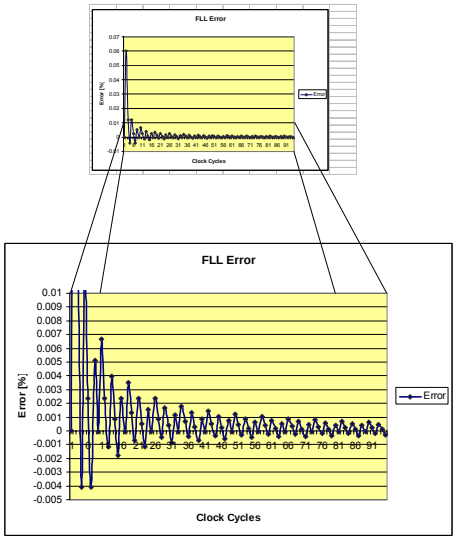




FLL: Regulation and Modulation

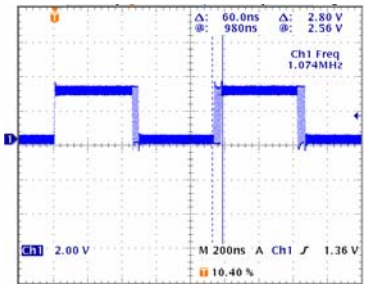
- On each Ref Clock Cycle the DCO tap and the modulation is updated
- The DCO could get one tap up or down.
- If the Frequency is locked the Tap will stay almost the same and only the Modulation is changed.
- The Modulation allows to change the DCO with each DCO clock cycle to the adjusted frequency and the frequency of the Tap +1 to get less time for zero frequency error.

Understanding the Error of an FLL



- Clock Accuracy: Average stability
  - Example for the 'Lock time' of the FLL
  - Shown:
    1MHz required Frequency
    DCO = 943000 MHz
    DCO+1 = 1037540 MHz
  - Clock Error < 0.1% after 50 clock cycles
  - Clock Error < 0.003% after 100 clock cycles

Understanding the Error of an FLL



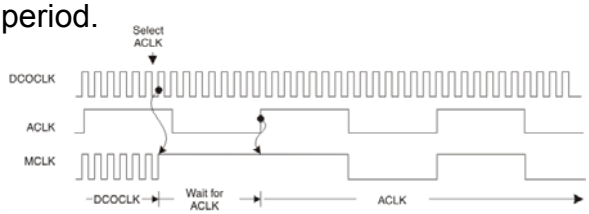
- Clock Accuracy: Cycle by cycle
  - FLL could change the DCO frequency with each FLLREF clock cycle
  - Modulation could change the DCO frequency with each DCO clock cycle

FLL versus PLL

	FLL	PLL
Cycle by Cycle Accuracy	Jitter of ~ 10%	Very small
Frequency step size	~ 10%	-
Long time Freq. Error	~ 0	~ 0
Startup time	< 5us	>100 clock cycles
Overshoot possible	Limited	Possible
Support for Low Power Mode	Very good	Limited due to long startup time
Switch on/off	Simple	Lock in required

Synchronization on Clock Switching

- Hardware controlled clock switching between asynchronous sources to avoid Glitches.
- The current clock cycle continues until the next rising edge.
- The clock remains high until the next rising edge of the new clock.
- The new clock source is selected and continues with a full high period.



Setting for certain Clock frequency

```
void init_fll(unsigned int fsystem, const unsigned int fcrystal)
{

  UCSCTL2 &= ~(0x3FF);                  // Reset FN bits
  // Choose the system frequency divider
  UCSCTL2= FLLD__x | ((fsystem/fcrystal)  - 1);
                                        // Set Loop Controll and feedback devider

  UCSCTL0 = 0x000;                      // Set DCO to lowest Tap
  UCSCTL1= DCORSEL_x ;                  // Set DCO to required Range

} // End of fll_init()
```


Agenda

- Introduction into the UCS system
- Oscillators (Overview, Characteristics, typical usage)
- Frequency Locked Loop (FLL)
- Low Power Mode support from the UCS
- Summary

Low Power Mode support from the UCS

- The FLL provides the fastest clock on and off switching with a stabilized clock.
  (LDO on + 6 clocks)
- Dynamic change of clock sources to select lowest possible clock for the application/module.
- Any clock request from a peripheral module will cause its respective clock off signal to be overridden.
  - Clocks are just on as required.
  - Clock could be switched on without CPU wake up.
- Keep in mind: Current consumption for a certain task is independent from the clock, if the System is in LPM during the remaining time, but a e.g. a Timer needs a higher current when it is running on a higher speed then required.

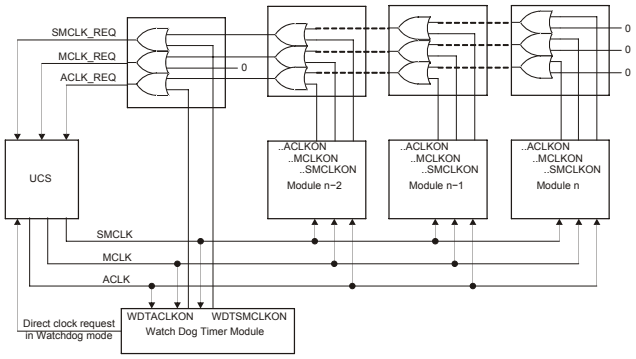
Getting the Application more robust

- The watchdog, due to its security requirement, actively selects the VLOCLK source if the originally selected clock source is not available.
- Many security aspects are already covered by hardware but take respect of the Fail save mechanism in your application to take the proper actions.
- Implement OSC Fault Interrupt Service Routine
- Using an input clock divider could prevent system locks or errors due to spikes
  (esp. for external clocks).

Fully Automatic Clock Requests

- A module can use a *clock request* to force its source to stay active, even when entering LPMx
- LPMx otherwise goes into effect

- When clock request goes away, clock shuts down & LPMx fully implemented
- Used much more in 5xx than in previous families



Review of Available Clocks

Clock	Frequency (nominal)	Precision	Current Draw	Crystal Required
High-Frequency				
DCO	**100kHz – 32MHz**	**Low**	**60uA**	
HFXT1/2	**4 - 32MHz**	**High**	**60uA @ 12MHz**	**X**
MODOSC	**5MHz**	**n/a**	**n/a**	
Low-Frequency				
LFXT1	**32kHz**	**High**	**300nA**	**X**
VLO	**12kHz**	**Low**	**0nA***	
REFO	**32kHz**	**Medium/High**	**3uA**	

* Included in $I_{LPM3, VLO}$ spec (~1.2uA)

Summary

- Many focus was set on safety and flexibility to meet your application requirements as good as possible.
- Configuration may needs a few more things to consider due to the higher flexibility.
- Default settings are already set to meet most of the common requirements.
- Crystal less operation possible in many cases.

Thank you