Agenda

- Oscillators, the source of precise timing
  - Internal Oscillators DCO, VLO, REFO
  - Crystal oscillators LFXT1 and XT2
  - FLL (Frequency locked loop) and calibration

- Precise Timing with Peripherals
  - Temperature measurement with Comparator_A
  - Software UART with Timer_A
  - USART and USCI
  - Sampling with ADC
  - Sample rate of DAC
**MSP430F2xx Basic Clock Module**

- **VLO**
  - Internal very low power, low frequency oscillator
  - ≈12kHz

- **LFXT1**
  - Crystal Oscillator
  - 32768Hz
  - 400kHz ... 16MHz

- **XT2**
  - Crystal Oscillator
  - 400kHz ... 16MHz
  - On high end MSP430

- **DCO**
  - Digitally Controlled Oscillator
  - ~100kHz ... >8MHz

---

**MSP430F4xx FLL+ Clock Module**

- **FLL**
  - Frequency Locked Loop

- **LFXT1**
  - Crystal Oscillator
  - 32768Hz
  - 400kHz ... 8MHz

- **DCO**
  - Digitally Controlled Oscillator
  - ~100kHz ... >8MHz

- **XT2**
  - Crystal Oscillator
  - 400kHz ... 8MHz
  - On high end MSP430
MSP430F5xx Unified Clock System (UCS)

- **LFXT1**: Crystal oscillator 32.768kHz, 400kHz ... >25MHz
- **VLO**: Internal very low power, low frequency oscillator ~12kHz
- **REFO**: Internal 32.768kHz Oscillator
- **XT2**: Crystal Oscillator 400kHz ... >25MHz
- **FLL**: Frequency Locked Loop
- **DCO**: Digitally Controlled Oscillator ~100kHz ... >25MHz
- **MODOSC**: Module Oscillator e.g. for ADC, Flash Controller etc.

Low Frequency Clock Sources

*Range of choices to fit application needs*

<table>
<thead>
<tr>
<th></th>
<th>POWER</th>
<th>PRECISION</th>
<th>COST</th>
</tr>
</thead>
<tbody>
<tr>
<td>XTAL</td>
<td>1μA</td>
<td>high</td>
<td>External Components</td>
</tr>
<tr>
<td>REFO</td>
<td>3μA</td>
<td>medium</td>
<td>$0</td>
</tr>
<tr>
<td>VLO</td>
<td>&lt;500nA</td>
<td>low</td>
<td>$0</td>
</tr>
</tbody>
</table>

ATC 2008
MSP430 Advanced Technical Conference
Agenda

• Oscillators, the source of precise timing
  – Internal Oscillators DCO, VLO, REFO
  – Crystal oscillators LFXT1 and XT2
  – FLL (Frequency locked loop) and calibration

• Precise Timing with Peripherals
  – Temperature measurement with Comparator_A
  – Software UART with Timer_A
  – USART and USCI
  – Sampling with ADC
  – Sample rate of DAC

DCO (Digitally Controlled Oscillator)

• Available on all MSP430
• Internal high speed oscillator
• Very fast wakeup time
  – MSP430F1xx: 6us
  – MSP430F2xx: 1us
  – MSP430F4xx: 6us
  – MSP430F5xx: 5us
  – In most applications CPU runs from DCO
• Different DCO frequency range on different MSP430 families
  MSP430F1xx: ~100kHz … 5MHz or ~8Mhz with $R_{OSC} = 100\,\Omega$
  MSP430F2xx: ~100kHz … 16MHz
  MSP430F4xx: ~100kHz … 8MHz
  MSP430F5xx: ~100kHz … 25MHz
• Different DCO frequency accuracy on different MSP430 families
  See details see on next slides
DCO frequency adjustment

<table>
<thead>
<tr>
<th>RSEL Bits</th>
<th>DCO Bits</th>
</tr>
</thead>
<tbody>
<tr>
<td>$s_{RSEL} = f_{RSEL+1} / f_{RSEL}$</td>
<td>$s_{DCO} = f_{DCO+1} / f_{DCO}$</td>
</tr>
<tr>
<td>MSP430F149</td>
<td>1.35 … 2</td>
</tr>
<tr>
<td>MSP430F249</td>
<td>&lt; 1.55</td>
</tr>
<tr>
<td>MSP430F449</td>
<td>1.49 … 2 (FN_x+1 / FN_x)</td>
</tr>
<tr>
<td>MSP430F5xx</td>
<td>1.5 … 2.1</td>
</tr>
</tbody>
</table>

MSP430F5xx values are preliminary!

ATC 2008
MSP430 Advanced Technical Conference

DCO Modulation
Question: Does the DCO have jitter?

DCO Modulation allows fine tuning of the DCO frequency.

The modulator mixes two DCO frequencies, $f_{DCOx}$ and $f_{DCOx+1}$ to produce an intermediate frequency between $f_{DCOx}$ and $f_{DCOx+1}$ and spread the clock energy, reducing electromagnetic interference (EMI).

Jitter Definitions

- **Cycle-to-cycle jitter**: (JEDEC Standard JESD65B)
  The variation in cycle time of a signal between adjacent cycles, over a random sample of adjacent cycle pairs.

  MSP430 modulation can generate 5% ... 17% cycle-to-cycle jitter.

- **Period jitter**: (JEDEC Standard JESD65B)
  Period jitter measures the maximum change in a clock’s output transition from its ideal over a large number of cycles (e.g. 1000 cycles)

  MSP430 modulation adds 0% long term jitter for n×32 clock cycles.
DCO accuracy

<table>
<thead>
<tr>
<th></th>
<th>$D_V$</th>
<th>$D_V$</th>
<th>$D_L$</th>
<th>$D_L$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Drift with $V_{CC}$ variation</td>
<td>1.8V ... 3.6V maximum over entire range</td>
<td>Temperature drift $V_{CC} = 3V$</td>
<td>0°C ... 85°C $V_{CC} = 3V$ maximum over entire range</td>
</tr>
<tr>
<td>MSP430F149</td>
<td>0 ... 10%/V</td>
<td>0 ... 18%</td>
<td>-0.33 ... -0.43%/°C</td>
<td>-28 ... -37%</td>
</tr>
<tr>
<td>MSP430F249</td>
<td>0 ... -2.5%/V (from graph in DS)</td>
<td>-3% ... +3% (Cal 1MHz ... 12MHz)</td>
<td>-0.03% ... +0.03%/°C</td>
<td>-2.5% ... +2.5% (Cal 1MHz ... 12MHz)</td>
</tr>
<tr>
<td>MSP430F449</td>
<td>0 ... 15%/V</td>
<td>0 ... -27%</td>
<td>-0.2 ... -0.4%/°C</td>
<td>-17 ... -34%</td>
</tr>
<tr>
<td>MSP430F5xx</td>
<td>0 ... 10%/V</td>
<td>0 ... 18%</td>
<td>-0.3% ... +0.3%/°C</td>
<td>-25% ... -25%</td>
</tr>
</tbody>
</table>

MSP430F4xx: LFXT1 & FLL stabilize DCO frequency

MSP430F5xx: ~ ± 3.5% over all with REFO & FLL

Influence of $R_{Osc}$ MSP430F1xx

MSP430F1xx:
- DCO frequency stability over temperature clearly improved with external resistor $R_{Osc}$
- Without $R_{Osc}$ $\Rightarrow$ maximum $f(DCO) \sim 5$Mhz
- With $R_{Osc} = 100k\Omega$ $\Rightarrow$ maximum $f(DCO) \sim 8$Mhz
Influence of $R_{\text{osc}}$ MSP430F2xx

**MSP430F2xx:**

- Slight improvement on temperature stability with external $R_{\text{osc}}$
- DCO without $R_{\text{osc}}$ already has a great frequency stability over temperature ($\pm 2.5\%$) and $V_{\text{cc}}$ ($\pm 3\%$)
- Maximum $f(DCO) > 16$MHz $\Rightarrow R_{\text{osc}}$ not necessary to increase frequency range.

Calibrated DCO Frequencies

**MSP430F2xx:**

- Calibrated DCO with values stored in Information Memory segment A for 1MHz, 8MHz, 12MHz and 16MHz:
- Code example:

  ```c
  if ((CALDCO_16MHZ != 0xFF) && (CALBC1_16MHZ != 0xFF))
  {DCOCTL = CALDCO_16MHZ;                // DCO = 16MHz calibrated
   BCSCTL1 = CALBC1_16MHZ;               // DCO = 16MHz calibrated
  }
  ```

**MSP430F5xx:**

- REFO ($\pm 3.5\%$) & FLL $\Rightarrow$ DCO ($\sim \pm 3.5\%$)
Agenda

• Oscillators, the source of precise timing
  – Internal Oscillators DCO, VLO, REFO
  – Crystal oscillators LFXT1 and XT2
  – FLL (Frequency locked loop) and calibration

• Precise Timing with Peripherals
  – Temperature measurement with Comparator_A
  – Software UART with Timer_A
  – USART and USCI
  – Sampling with ADC
  – Sample rate of DAC

VLO (Very Low Speed and Very Low Power Oscillator)

• Fixed frequency ~12KHz
• Enables LPM3 operation like a 32kHz crystal, but without external components
  ☑ Lower cost
  ☑ Less board-space
  ☑ Less sensitive to EMI and crosstalk
• Lower LPM3 supply current consumption compared to 32kHz crystal
• Considerable higher frequency tolerance than LFXT1 with 32kHz crystal
VLO specification in data sheet

Internal very low power, low frequency oscillator (VLO)

<table>
<thead>
<tr>
<th>PARAMETER</th>
<th>TEST CONDITIONS</th>
<th>VCC</th>
<th>MIN</th>
<th>TYP</th>
<th>MAX</th>
<th>UNIT</th>
</tr>
</thead>
<tbody>
<tr>
<td>( f_{\text{VLO}} ) VLO frequency</td>
<td>See Note 6</td>
<td>2.2 V/3 V</td>
<td>0.5</td>
<td>%/°C</td>
<td></td>
<td>kHz</td>
</tr>
<tr>
<td>( \frac{df_{\text{VLO}}}{dT} ) VLO frequency temperature drift</td>
<td>See Note 7</td>
<td>1.8 V to 3.6 V</td>
<td>4</td>
<td>%/V</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( \frac{df_{\text{VLO}}}{V_{\text{CC}}} ) VLO frequency supply voltage drift</td>
<td>See Note 6</td>
<td>2.2 V/3 V</td>
<td></td>
<td>%/V</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

NOTES:
6. Calculated using the box method:
   - i version: \( \frac{(\text{MAX}(\text{MAX}(40 \text{ to } 85\degree C)) - \text{MIN}(\text{MIN}(40 \text{ to } 85\degree C)))}{\text{MIN}(\text{MIN}(40 \text{ to } 85\degree C))} \)
   - T version: \( \frac{(\text{MAX}(\text{MAX}(40 \text{ to } 105\degree C)) - \text{MIN}(\text{MIN}(40 \text{ to } 105\degree C)))}{\text{MIN}(\text{MIN}(40 \text{ to } 105\degree C))} \)
7. Calculated using the box method: \( \frac{(\text{MAX}(1.8 \text{ to } 3.6 \text{ V}) - \text{MIN}(1.8 \text{ to } 3.6 \text{ V}))}{\text{MIN}(1.8 \text{ to } 3.6 \text{ V})} \)

VLO is ultra low power oscillator for LPM3, for cost sensitive applications where frequency accuracy is not important.

Example: Supply current MSP430F249:
- \( \text{ICC (VLO, LPM3, } V_{\text{CC}}=3\text{V, } 25\degree C): \) typ. 0.4uA, max. 1uA
- \( \text{ICC (LFXT1, LPM3, } V_{\text{CC}}=3\text{V, } 25\degree C): \) typ. 1uA, max. 1.4uA

Agenda

- Oscillators, the source of precise timing
  - Internal Oscillators DCO, VLO, REFO
  - Crystal oscillators LFXT1 and XT2
  - FLL (Frequency locked loop) and calibration
- Precise Timing with Peripherals
  - Temperature measurement with Comparator_A
  - Software UART with Timer_A
  - USART and USCI
  - Sampling with ADC
  - Sample rate of DAC
**REFO (Trimmed Reference Oscillator)**

- Available on MSP430F5xx
- Fixed frequency 32768Hz
- Moderate frequency tolerance over voltage/temp
  - Similar to calibrated DCO, much better than VLO
  - Less accurate than 32kHz crystal
- Power draw is higher than crystal or VLO
- Is the default FLL reference clock

---

**REFO specification in data sheet**

<table>
<thead>
<tr>
<th>PARAMETER</th>
<th>TEST CONDITIONS</th>
<th>VCC</th>
<th>MIN</th>
<th>TYP</th>
<th>MAX</th>
<th>UNIT</th>
</tr>
</thead>
<tbody>
<tr>
<td>fREFO (REFO oscillator current consumption)</td>
<td>T_A = 25°C</td>
<td>1.8 V to 3.6 V</td>
<td>3</td>
<td></td>
<td></td>
<td>μA</td>
</tr>
<tr>
<td>fREFO (REFO frequency calibrated)</td>
<td>Measurement at ACLK</td>
<td>1.8 V to 3.6 V</td>
<td>32768 Hz</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>REFO absolute tolerance calibrated</td>
<td>T_A = 25°C</td>
<td>3 V</td>
<td>±3.5</td>
<td></td>
<td>±TBD</td>
<td>%</td>
</tr>
<tr>
<td>Duty Cycle</td>
<td>Measurement at ACLK</td>
<td>1.8 V to 3.6 V</td>
<td>40</td>
<td>50</td>
<td>60</td>
<td>%</td>
</tr>
<tr>
<td>tSTART (REFO startup time)</td>
<td>40%/60% duty cycle</td>
<td>1.8 V to 3.6 V</td>
<td>0.4</td>
<td></td>
<td></td>
<td>ms</td>
</tr>
</tbody>
</table>

- REFO is a low power oscillator for LPM3, without external components, if moderate frequency accuracy is sufficient.

Example: Supply current MSP430F5xx:
- ICC (REFO, LPM3, 25°C): typ. ~5μA
- ICC (LFXT1): typ. ~2.6μA
What Can You Do With REFO?

• Improve DCO frequency stability with REFO & FLL

• Periodic wakeup for apps in which these are true...
  – Don’t need crystal accuracy
  – Need better accuracy than VLO
  – No external crystal required
    ☰ 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!

Agenda

• Oscillators, the source of precise timing
  – Internal Oscillators DCO, VLO, REFO
  – Crystal oscillators LFXT1 and XT2
  – FLL (Frequency locked loop) and calibration

• Precise Timing with Peripherals
  – Temperature measurement with Comparator_A
  – Software UART with Timer_A
  – USART and USCI
  – Sampling with ADC
  – Sample rate of DAC
Standard Pierce Oscillator

- The characteristics of the "Inverter" defines the oscillators' performance and crystal constraints.
- The serial resistor $R_D$ helps to suppress the crystal's overtone mode and can help to reduce the drive level.
- The chosen crystal and its specification should match the oscillator design and constraints.
- The load-capacitors $C_{L1}$ and $C_{L2}$ (regarded as in series) form the oscillator's effective load-capacitance.

- Loop gain: At steady state, the closed-loop gain = 1
- Phase shift: At the frequency of oscillation, the closed loop phase shift = $2\pi$

How to check a crystal oscillator?

- Effective Load-Capacitance
  (frequency accuracy)

- OA Oscillation allowance
  (negative-resistance model, Crystal constraints)

- Board Layout
  (minimize crosstalk and negative EMI effects)
ΔF_L dependent on C_L: good match

Required Effective Load Capacitance

Actual Effective Load Capacitance

Frequency FL dependent on C_L: bad match

Required Effective Load Capacitance

Actual Effective Load Capacitance

Correct frequency 32768 Hz

Wrong frequency ~32770 Hz
### 32kHz oscillator $C_L$ configurations

<table>
<thead>
<tr>
<th>MSP430F1xx</th>
<th>Internal $C_L$ ((\text{effective load capacitance}))</th>
<th>External $C_L$</th>
</tr>
</thead>
<tbody>
<tr>
<td>fixed 6pF</td>
<td>In addition to internal $C_L$</td>
<td></td>
</tr>
<tr>
<td>MSP430F2xx</td>
<td>~0pF, 6pF, 10pF, 12pF</td>
<td>free choice</td>
</tr>
<tr>
<td>MSP430F4xx</td>
<td>~0pF, 6pF, 8pF, 10pF</td>
<td>free choice</td>
</tr>
<tr>
<td>MSP430F5xx</td>
<td>2pF, 5.5pF, 8.5pF, 12pF</td>
<td>free choice</td>
</tr>
</tbody>
</table>

**Determine correct $C_L$**

- Measure crystal oscillator frequency with frequency counter
- Don’t measure with probes directly at the crystal pins
  - Typical 32kHz load capacitance is 6pF … 15pF
  - Ultra-low-power oscillator input impedance > 5MΩ
  - Probe capacitance and impedance heavily influences oscillator
- Measure at digital ACLK output pin
  
  ```
  P2DIR = BIT0; // MSP430F24x pin ‘P2.0/ACLK/CA2’
  P2SEL = BIT0; // ‘P2.0/ACLK/CA2’ => ACLK output
  ```
- Result on the frequency counter must be 32768 Hz
- This measurement includes parasitic capacitance of board and pins
ESR (Equivalent Series Resistance)

The ESR (Equivalent Series Resistance) is given in a crystal’s data sheet and can be calculated with the following formula from the equivalent circuit:

$$ESR = R_M \left( 1 + \frac{C_0}{C_L} \right)^2$$

Oscillator Allowance

The Oscillation Allowance test is a method to measure the Oscillator-performance:
- Resistor Rx is placed in series to the crystal.
- Increase Rx until oscillation stops / starts.

Example: $OA = 200k\Omega + 50k\Omega$
Board Layout examples

Layout without and with external load capacitors
(XIN and XOUT neighboring pins are standard function pins)

Layout with external capacitors and ground guard ring
(XIN and XOUT neighboring pins are NC pins)
Examples for MSP430F41x and MSP430F1232IRHB

Temperature characteristics

curve for 0ppm crystal

AT-Cut

Tuning Fork

32768Hz crystals

source: MicroCrystal
ppm = parts per million

- 1ppm ⇒ 1 second off, after 11 days 13:46:40 hours
  ~31.5 seconds off per year
- 150ppm ⇒ 1 second off, after 1:51:07 hours
  ~12 seconds off per day
- 32768 Hz crystal:
  1ppm ⇒ Δ = 0.0328Hz
  150ppm ⇒ Δ = 4.9152Hz

Temperature compensation in software

- Measure temperature with Comparator_A or ADC
- Method 1:
  - Timer_A always divides by 32768
  - Depending on the temperature profile over time, add or subtract a second if the time is expected to be >0.5 second off.
- Method 2:
  - Divide Timer_A by 32768 or 32769, depending on temperature
    - Divide by 32768 ⇒ 0 ppm correction
    - Divide by 32769 ⇒ +30 ppm correction
  - 1ppm correction with modulation of Timer_A in software:
    Within 30 cycles:
    - ’n’ cycles divide by 32769
    - ’30-n’ cycles divide by 32768

This procedure needs frequent CPU interaction ⇒ Higher power consumption!
DMA can do modulation with much less power consumption.
Agenda

- Oscillators, the source of precise timing
  - Internal Oscillators DCO, VLO, REFO
  - Crystal oscillators LFXT1 and XT2
  - FLL (Frequency locked loop) and calibration

- Precise Timing with Peripherals
  - Temperature measurement with Comparator_A
  - Software UART with Timer_A
  - USART and USCI
  - Sampling with ADC
  - Sample rate of DAC

HF oscillator $C_L$ configurations

<table>
<thead>
<tr>
<th>Internal $C_L$</th>
<th>External $C_L$</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSP430F1xx</td>
<td>no</td>
</tr>
<tr>
<td>MSP430F2xx</td>
<td>no</td>
</tr>
<tr>
<td>MSP430F4xx</td>
<td>no</td>
</tr>
<tr>
<td>MSP430F5xx</td>
<td>no</td>
</tr>
</tbody>
</table>
Temperature characteristics

**Agenda**

- Oscillators, the source of precise timing
  - Internal Oscillators DCO, VLO, REFO
  - Crystal oscillators LFXT1 and XT2
  - FLL (Frequency locked loop) and calibration

- Precise Timing with Peripherals
  - Temperature measurement with Comparator_A
  - Software UART with Timer_A
  - USART and USCI
  - Sampling with ADC
  - Sample rate of DAC
MSP430F4xx and MSP430F5xx FLL

32768Hz
LFXT1

~ 32768Hz

DCO / (FDLLx \times (N+1))

Divider

- FLL compares frequencies at '+' and '-' input
- If \( f('+') > f('-') \) then increment DCO+MOD bits
  else decrement DCO+MOD bits

FLL Frequency adjustment

Increasing DCO+MOD by '1' means within 32 clock cycles

⇒ one more clock cycle at the frequency DCOx+1

⇒ one clock cycles less at the frequency DCOx

no impact on Cycle-to-cycle and Period jitter
**FLL at power-on**

- FLL loop control is turned on by default
  - Status Register Bit SCG0 = 0 (default value)
- Directly after power-on the 32768Hz crystal oscillator starts oscillating and frequency < 32768Hz
  - FLL reduces the DCO frequency
- After start-up of the 32786Hz crystal oscillator
  - FLL increases the DCO frequency until it reaches the programmed value
- If the application requires to avoid DCO frequency reduction at start-up
  - Set SCG0=1 at the beginning of your software
  - Reset SCG0 to 0 when the oscillator fault detection indicates a stable 32768Hz crystal oscillator

**MSP430F1xx/F2xx DCO Calibration with SW**

4096Hz

// Partial SW FLL Code
if (244 < Compare ) // DCO too fast
DCCON--;
else DCCON++;     // DCO too slow

Periodic loop can adjust DCO

'Software FLL' in case Hardware FLL is not present
MSP430x1xx/F2xx Software FLL Example

Goal:
- Set \( f(DCO) = 1,000,000 \text{Hz} \) with \( f(ACLK) = 4096 \text{Hz} \)

Steps to take:
- \( f(DCO) = 1,000,000 \text{Hz} \)  \( \Rightarrow \) DCO is clock source for Timer_A
- \( f(ACLK) = 4096 \text{Hz} \)  \( \Rightarrow \) Triggers 244us capture of \( f(DCO) \) on CCI2B
- Adjust \( f(DCO) \) until capture value is 244 (= \( 1,000,000 \div 4096 \))

Code Library ‘Using the DCO’

This library encapsulates routines used for setting the DCO to a specific speed based on a multiplication of a known clock, such as a 32-kHz crystal. These functions are written in assembly to be optimized for the MSP430 but can be called from any C program that includes their header files.

VLO Calibration

- The VLO frequency is fixed and cannot be changed
- Temperature and V\textsubscript{CC} influence the VLO frequency
- Based on a calibrated DCO, the actual frequency of the VLO can be measured with a Timer
- Once the VLO frequency is known, software can adjust timing

**Code Library ‘Using the VLO’**

The VLO library contains only the function TI\_measureVLO(), when called, performs the following actions:

1. Save the current clock settings in registers and on the stack.
2. Set the DCO to the 1MHz calibrated value stored in flash.
4. Measure the number of 1-MHz clock pulses in 1 ACLK (VLO/8) pulse.
5. Store the measured result in the variable TI\_8MHz\_Counts\_Per\_VLO\_Clock.
6. Reload the previous clock settings.
7. Return variable TI\_8MHz\_Counts\_Per\_VLO\_Clock from the function.

Agenda

• Oscillators, the source of precise timing
  – Internal Oscillators DCO, VLO, REFO
  – Crystal oscillators LFXT1 and XT2
  – FLL (Frequency locked loop) and calibration

• Precise Timing with Peripherals
  – Temperature measurement with Comparator_A
  – Software UART with Timer_A
  – USART and USCI
  – Sampling with ADC
  – Sample rate of DAC

Precise Timing

• Precise timing ⇒ Triggering in hardware
  – MSP430 peripherals offer trigger signals between peripherals
    e.g. Comparator_A ⇒ Timer_A
    Timer_A ⇒ ADC10 or ADC12
    Timer_A ⇒ DAC12
  – MSP430 peripherals have additional hardware for precise timing
    e.g. Timer_A SCCI latch
    Group load logic in Timer_B and DAC12

• Triggering in software ⇒ This is not precise
  – Interrupts add delay to program execution
    ⇒ Turn off Interrupts if timing is based on program execution
  – Depending on the addressing mode used, MSP430 instruction execution needs between 1 and 6 MCLK clock cycles
    ⇒ Interrupt latency uncertainty of 5 MCLK clock cycles
Agenda

- Oscillators, the source of precise timing
  - Internal Oscillators DCO, VLO, REFO
  - Crystal oscillators LFXT1 and XT2
  - FLL (Frequency locked loop) and calibration

- Precise Timing with Peripherals
  - Temperature measurement with Comparator_A
  - Software UART with Timer_A
  - USART and USCI
  - Sampling with ADC
  - Sample rate of DAC

Comparator_A+ ⇔ Timer_A

Timer_A can record a time-stamp of a Comparator_A input slope in TACCR
Slope Conversion (simple method)

- Slope conversion is an excellent method to measure resistors e.g. measure temperature
- Result of simple slope measurement depends on R, C, \( CA_{REF} \) and \( V_{CC} \)

\[
t = - (R \times C) \times \ln\left(\frac{CA_{REF}}{V_{CC}}\right)
\]

Counts = \( t \times f_{\text{Timer}} \)

Ratiometric Slope Conversion

Independent from C, \( CA_{REF} \) and \( V_{CC} \)

\[
\begin{align*}
R_{NTC} &= 10k \times \frac{t_{NTC}}{t_{10k}} \\
\frac{t_{NTC}}{t_{10k}} &= \frac{-R_{NTC} \times C \times \ln V_{CAREF}}{V_{CC}}
\end{align*}
\]

\( CA_{REF} = 0.25 \times V_{CC} \)
Ratiometric Slope Conversion

• Relatively slow, but very low cost measurement method for resistors
  ⇒ Perfect for temperature measurement
• Measuring discharge time from $V_{CC}$ to $0.25 \times V_{CC}$ removes influence of $V_{CC}$ and comparator threshold $V_{REF}$
• Comparison measurement of discharge time with a known resistor removes influence of capacitor $C$
• If made perfect, accuracy of more than 12 bits is realistic

For details please see:
• Implementing An Ultralow-Power Thermostat With Slope A/D Conversion (slaa129b)
• Family User's Guides, chapter Comparator_A and/or ComparatorA+ for details

Agenda

• Oscillators, the source of precise timing
  – Internal Oscillators DCO, VLO, REFO
  – Crystal oscillators LFXT1 and XT2
  – FLL (Frequency locked loop) and calibration
• Precise Timing with Peripherals
  – Temperature measurement with Comparator_A
  – Software UART with Timer_A
  – USART and USCI
  – Sampling with ADC
  – Sample rate of DAC
Timer_A3

- Async 16-Bit timer/counter with four modes
- Int/ext clock source
- Three capture/compare registers
- Outputs with PWM capability
- SCCI asynchronous input signal latching
- Interrupt vector register for fast decoding
- DMA enabled

Low-Overhead UART Implementation

- 100% hardware bit latching and output
- Full speed from LPM3 and LPM4
- Low CPU Overhead
- App Note SLAA078
Timer_A UART TX SW

Timer output logic modifies TXD

| X | ST | "1" | "1" | "0" | "1" | "1" | "0" | "0" | "1" | SP |

Interrupts

```
TA0_ISR  add.w #Bitime,&CCR0     ; Time to Next Bit
UART_TX  dec.w BitCnt           ; Dec bit counter
        jne TX_Next                 ; Next bit?
        bic.w #CCIE,&CCTL0         ; Done, disable int
        reti                       ;
TX_Next  bic.w #OUTMOD2,&CCTL0  ; TX Mark
        rra.w RXTXData             ; LSB->Carry
        jc TX_Test                 ; Jump if bit = 1
TX_Space bis.w #OUTMOD2,&CCTL0  ; TX Space
TX_Test  reti                   ;
```

Timer_A UART RX SW

1. Capture mode:
   Wait for falling edge of start bit and store a time stamp \( t_s \) of its occurrence in TACCR

2. Compare mode:
   1. Program TACCR to \( t_1 = t_s + 150\% \) of a bit time
      This loads 1st bit of RX signal into SCCI latch
   2. Program TACCR to \( t_2 = t_1 + 100\% \) of a bit time
      This loads 2nd bit of RX signal into SCCI latch
   3. Program TACCR to \( t_3 = t_2 + 100\% \) of a bit time
      This loads 3rd bit of RX signal into SCCI latch
      ..... (continue until all bits are captured)
### Timer_A UART RX SW

Timer captures first edge, then switches to compare mode

- **TA0_ISR**: `add.w #Bitime,&CCR0 ; Time to Next Bit`
- **bit.w #CAP,&CCTL0 ; Capture mode?**
- **jz RX_Bit ; Is start bit edge**
- **RX_Edge**: `bic.w #CAP,&CCTL0 ; Enable comp. mode`
- **add.w #Bitime_5,&CCR0 ; Add 0.5 T_bit**
- **reti**
- **RX_Bit**: `bit.w #SCCI,&CCTL0 ; Read RX latch`
- **rrc.b RXTXData ; Store received bit**
- **RX_Test**: `dec.w BitCnt ; All bits RXed?`
- **jnz RX_Next ; No, next bit**
- **RX_Comp**: `bic.w #CCIE,&CCTL0 ; All bits RXed!`
- **RX_Next**: `reti`

---

### CPU Load Using Timer_A UART

**19,200 Baud RX, 8 MHz MCLK**

- Avg. CPU Load: **7.4%**
  - 2X Baud = 2X CPU Load
  - 2X MCLK = ½ CPU Load
  - Optimized Assembly ISRs

**19,200 Baud TX, 8 MHz MCLK**

- Avg. CPU Load: **7.0%**
Agenda

• Oscillators, the source of precise timing
  – Internal Oscillators DCO, VLO, REFO
  – Crystal oscillators LFXT1 and XT2
  – FLL (Frequency locked loop) and calibration

• Precise Timing with Peripherals
  – Temperature measurement with Comparator_A
  – Software UART with Timer_A
  – USART and USCI
  – Sampling with ADC
  – Sample rate of DAC

USART

The USART supports 2 or 3 operating modes:

- Asynchronous RS232 Mode
- Synchronous SPI Mode (3-Wire & 4 Wire)
- I²C interface – MSP430F15x/16x/16xx (Master & Slave)
**USART Asynchronous Mode**

- Software selectable UART or SPI
- Auto-start from any LPMx
- Double buffered RX and TX shift registers
- Baud-rate generator
- 7 or 8-bit data
- 9-bit addressing mode available
- Parity generation and detection
- Error detection and suppression

---

**Example 1:**

SMCLK = 1MHz  
Baud rate = 19200  

\[
1,000,000/19,200 = 52.08 \equiv 0x34
\]

UBR1 | UBR0 = 0x00 0x34; \( \equiv +0.16\% \) error

---

**Example 2:**

ACLK = 32,768Hz  
Baud rate = 9600  

\[
32,768/9,600 = 3.413
\]

UBR1 | UBR0 = 0x00 0x03; \( \equiv +13.8\% \) error  
UBR1 | UBR0 = 0x00 0x04; \( \equiv -14.7\% \) error

---

**USART Baud Rate Generator**

**Prescaler Factor UxBRx \( \geq 3 \)**

---

**ATC 2008**

MSP430 Advanced Technical Conference
**Example:**

ACLK = 32,768

Baud = 9,600 = 32,768/9,600 = 3.48

UBR1 | UBR0 | UMCTL = 0x00 0x3 0xA

**USART Baud Rate Modulation**

- Modulator mixes adjacent clock dividers to enable high baud rates even with low frequency XTAL.

**Table with 'Commonly Used Baud Rates, Baud Rate Data, and Errors' in User's Guide, chapter 'USART' in 'UART Mode'*

**Agenda**

- Oscillators, the source of precise timing
  - Internal Oscillators DCO, VLO, REFO
  - Crystal oscillators LFXT1 and XT2
  - FLL (Frequency locked loop) and calibration

- Precise Timing with Peripherals
  - Temperature measurement with Comparator_A
  - Software UART with Timer_A

- USART and USCI
  - Sampling with ADC
  - Sample rate of DAC
USCI Baud Rate Generator

Oversampling mode disabled
- Same behavior like USART Baud Rate Generator
- Modulation according to the tables ‘BITCLK Modulation Pattern’ and ‘Commonly Used Baud Rates, Settings, and Errors, UCOS16 = 0’

USCI Baud Rate Modulation

Commonly Used Baud Rates, Settings, and Errors, UCOS16 = 0

<table>
<thead>
<tr>
<th>BRCLK frequency [Hz]</th>
<th>Baud Rate [Baud]</th>
<th>UCBRx</th>
<th>UCBRx6</th>
<th>UCBFRx</th>
<th>Max. TX Error [%]</th>
<th>Max. RX Error [%]</th>
</tr>
</thead>
<tbody>
<tr>
<td>32,768</td>
<td>1200</td>
<td>27</td>
<td>2</td>
<td>0</td>
<td>-2.8</td>
<td>1.4</td>
</tr>
<tr>
<td>32,768</td>
<td>2400</td>
<td>13</td>
<td>6</td>
<td>0</td>
<td>-4.8</td>
<td>6.0</td>
</tr>
<tr>
<td>32,768</td>
<td>4800</td>
<td>6</td>
<td>7</td>
<td>0</td>
<td>-12.1</td>
<td>5.7</td>
</tr>
<tr>
<td>32,768</td>
<td>9600</td>
<td>3</td>
<td>3</td>
<td>0</td>
<td>-21.1</td>
<td>15.2</td>
</tr>
<tr>
<td>1,048,576</td>
<td>9600</td>
<td>109</td>
<td>2</td>
<td>0</td>
<td>-21.1</td>
<td>18.3</td>
</tr>
<tr>
<td>1,048,576</td>
<td>19200</td>
<td>54</td>
<td>5</td>
<td>0</td>
<td>-4.8</td>
<td>6.0</td>
</tr>
<tr>
<td>1,048,576</td>
<td>38400</td>
<td>27</td>
<td>2</td>
<td>0</td>
<td>-2.8</td>
<td>1.4</td>
</tr>
</tbody>
</table>

BITCLK Modulation Pattern

Both Tables are in the User’s Guide

Number of “1” in Modulation Pattern
USCI Baud Rate Generator

Oversampling mode enabled
- Two modulators
- RX sampled using BITCLK16
- Majority vote always at the same location within one RX bit
- BITCLK16 supports IrDA TX pulse generation
- Quasi-standard for UART, IrDA & LIN

Agenda

- Oscillators, the source of precise timing
  - Internal Oscillators DCO, VLO, REFO
  - Crystal oscillators LFXT1 and XT2
  - FLL (Frequency locked loop) and calibration

- Precise Timing with Peripherals
  - Temperature measurement with Comparator_A
  - Software UART with Timer_A
  - USART and USCI
  - Sampling with ADC
  - Sample rate of DAC
Timer Triggers For ADC / DAC

- Software ADC trigger
  Time uncertainty ⊆ Voltage error
- Hardware ADC trigger with Timer eliminates phase error
  Precise timing ⊆ Precise voltage

ADC12

- 12 Bit SAR
- 200ksps+
- Autoscan
- Single, sequence, repeat-single, repeat-sequence
- Int/ext V_{Ref}
- Temp sensor
- Batt measure
  - TA/TB trigger
- 16-word conversion buffer
- DMA enabled

ATC 2008
MSP430 Advanced Technical Conference
**Agenda**

- Oscillators, the source of precise timing
  - Internal Oscillators DCO, VLO, REFO
  - Crystal oscillators LFXT1 and XT2
  - FLL (Frequency locked loop) and calibration
- Precise Timing with Peripherals
  - Temperature measurement with Comparator_A
  - Software UART with Timer_A
  - USART and USCI
  - Sampling with ADC
  - Sample rate of DAC

**Analog wave output with DAC**

- Timer_A generates sampling rate and triggers DMA
- DMA loads new digital value from Flash to DAC

Wrong trigger sequence because of timing uncertainty for DMA data and address bus access!
MSP430 DAC12

- 12-bit monotonic output
- 8- or 12-bit voltage output resolution
- Programmable settling time vs power consumption
- Internal or external reference selection
- Straight binary or 2’s compliment data format
- Self-calibration option for offset correction
- Synchronized update capability for multiple DAC12s

Analog wave output with DAC

- Timer_A generates sampling rate and triggers DAC
- DAC loads output register ‘ADC12_xLatch’ with value of shadow register ‘ADC12_xDat’
- DAC with empty shadow register ‘ADC12_xDat’ triggers DMA to get next audio sample from Flash
- DMA copies digital audio data from Flash to DAC
Summary

• The complete clock chain is important for precise timing:
  – Oscillators are the source of precise timing
    • Internal Oscillators DCO, VLO, REFO
    • Crystal oscillators LFXT1 and XT2
    • FLL (Frequency locked loop) and calibration

• Peripherals offer features for precise timing
  – Comparator_A can trigger Timer
  – SCCI latch in Timer_A to sample at a predefined time
  – USART and USCI offer special features to fine tune timing
  – Precise ADC sampling with hardware triggering
  – Sample rate of DAC

Thank you