# AURIX 2G Reset、Clock、WDT

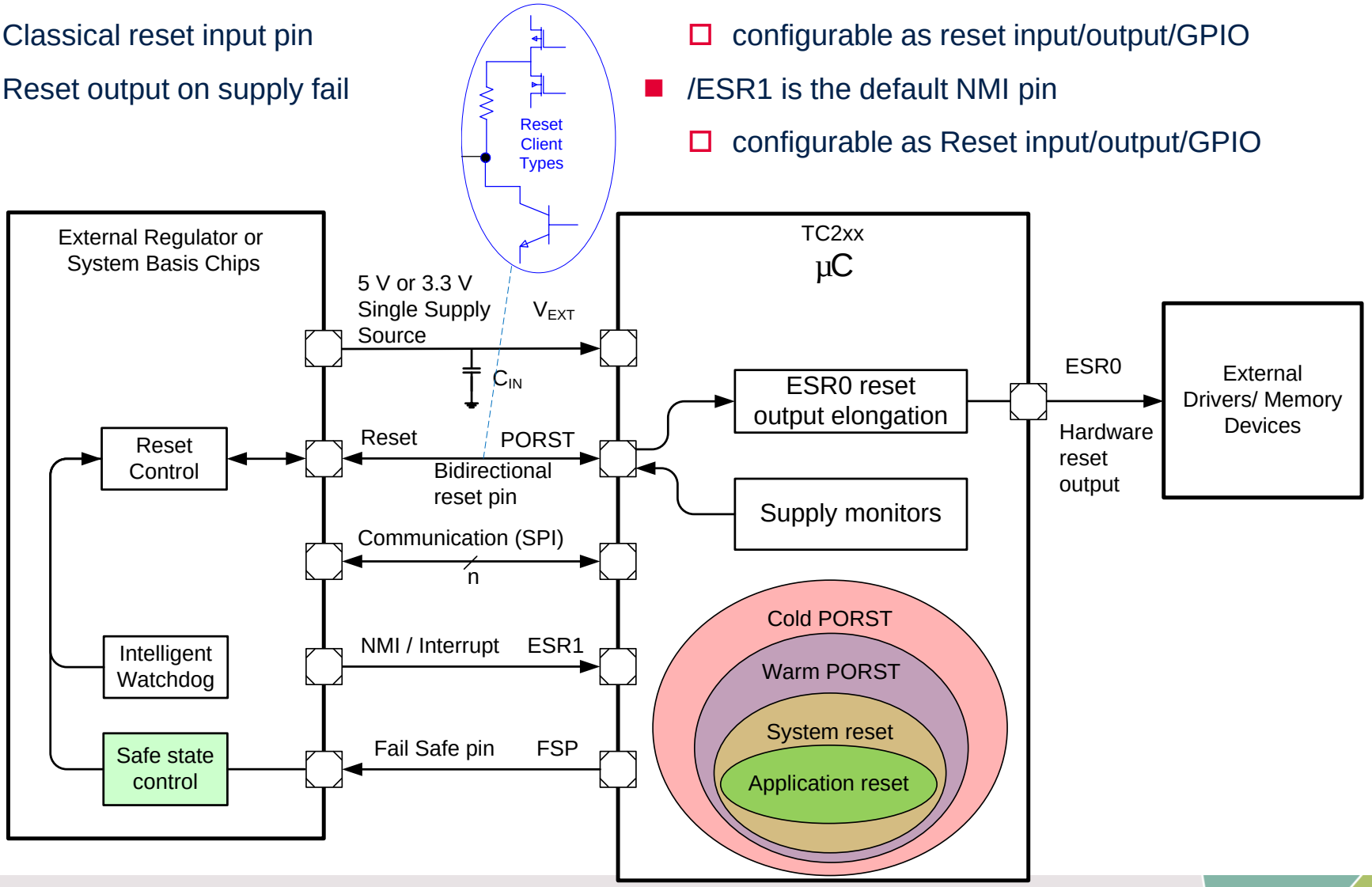
IFCN ATV SMD GC SAE MC

# Content

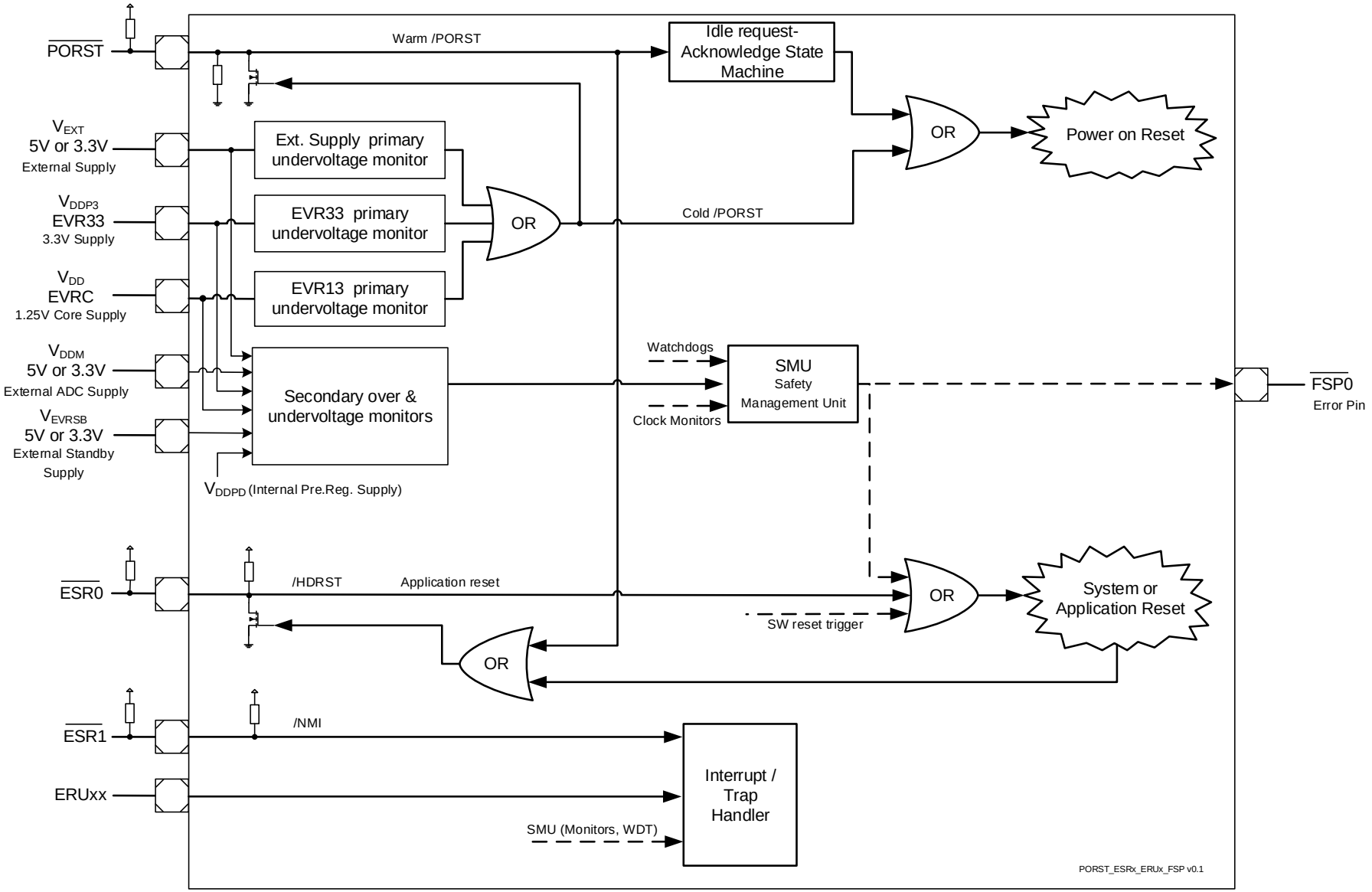
- Reset introduction
- Clock system
- WDT introduction

# TC3xx
# Reset pins and interface

- /PORST is a bidirectional reset pin
  - Classical reset input pin
  - Reset output on supply fail
- /ESR0 is the default reset output pin
  - configurable as reset input/output/GPIO
- /ESR1 is the default NMI pin
  - configurable as Reset input/output/GPIO



# TC3xx
# Reset pins and interface



# TC3xx
# Reset types, triggers and effect

| Reset Type (Class) | Reset Trigger(s) | Additional Modules affected by Cold Power-on | Additional Modules affected by warm PORST | Additional Modules affected by System reset | Modules affected by Application reset |
|---|---|---|---|---|---|
| **Cold Power-on Reset** | ▪Startup<br>▪VEXT supply < 3.0V<br>▪VDDP3 supply < 3.0V<br>▪VDD supply < 1.125V | ▪EVR<br>▪Internal clocks<br>▪RAMs | ▪JTAG interface<br>▪OCDS<br>▪MCDS<br>▪8 bit µC (opt.) | ▪Flash memory<br>▪XTAL/Osc./PLL<br>▪ESRx pins | ▪All CPUs<br>▪All Peripherals<br>▪SCU except EVR<br>▪Port pins except ESRx<br>▪RAMs<br>-Dcache invalid<br>-Pcache invalid |
| **Warm Power-on Reset** | ▪PORST pad asserted | | | | |
| **System Reset** | ▪ESR0/ESR1<br>▪SMU<br>▪STMx<br>▪Watchdog (via SMU)<br>▪Software reset | | | | |
| **Application Reset** | ▪ESR0/ESR1<br>▪SMU<br>▪STMx<br>▪Software reset<br>▪Tuning protection | | | | |

A higher reset encapsulates a lower reset
A higher reset resets all the modules reset by a lower reset

| | | |
|---|---|---|
| **SW Module reset** | ▪<Module>_KRST0.RST<br>▪<Module>_KRST1.RST | ▪Available for All CPUs , Each DMA Channel, QSPI, CAN, ASCLIN, Flexray, Ethermac, MSC, HSSL, GTM, CCU6, GPT12, ADC, SENT, SD_ADC |
| **Debug Reset** | ▪OCDS request trigger<br>▪JTAG reset | ▪OCDS + MCDS reset , All CPUs and peripherals (except SCU) are put into reset. |

# TC3xx
# Clearing reset status register RSTSTAT

**EVRRSTSTAT**
**EVR Reset Status Register** **(0044$_H$)** **Reset Value: Table 491**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | | | RSTSWDOFF | 0 | RST33OFF | 0 | RSTCOFF | RSTSWD | | | | | | | |
| r | | | rh | r | rh | r | rh | rh | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RST33 | | | | | | | | RSTC | | | | | | | |
| rh | | | | | | | | rh | | | | | | | |

- Reset was triggered by Event X
  - $0_B$ The last reset was not requested by this reset trigger X
  - $1_B$ The last reset was requested by this reset trigger X

- Following flags are cleared by **RSTCON2.CLRC** bit

( This is intended so that Cold Power-on reset information is not lost owing to consecutive resets)

  - Supply watchdog (Supply below 3.0V)
  - EVR33 (EVR33 voltage below 3.0V)
  - EVRC (EVR13 voltage below 1.13V)

- All other Flags are cleared with the **next reset**
  - PORST (PORST pad was asserted low)
  - ESR0 (ESR0 configured as reset and pad was asserted low)
  - ESR1 (ESR1 configured as reset and pad was asserted low)
  - Safety Management unit triggered a reset
  - SW - Software reset
  - STMx ( System timer)

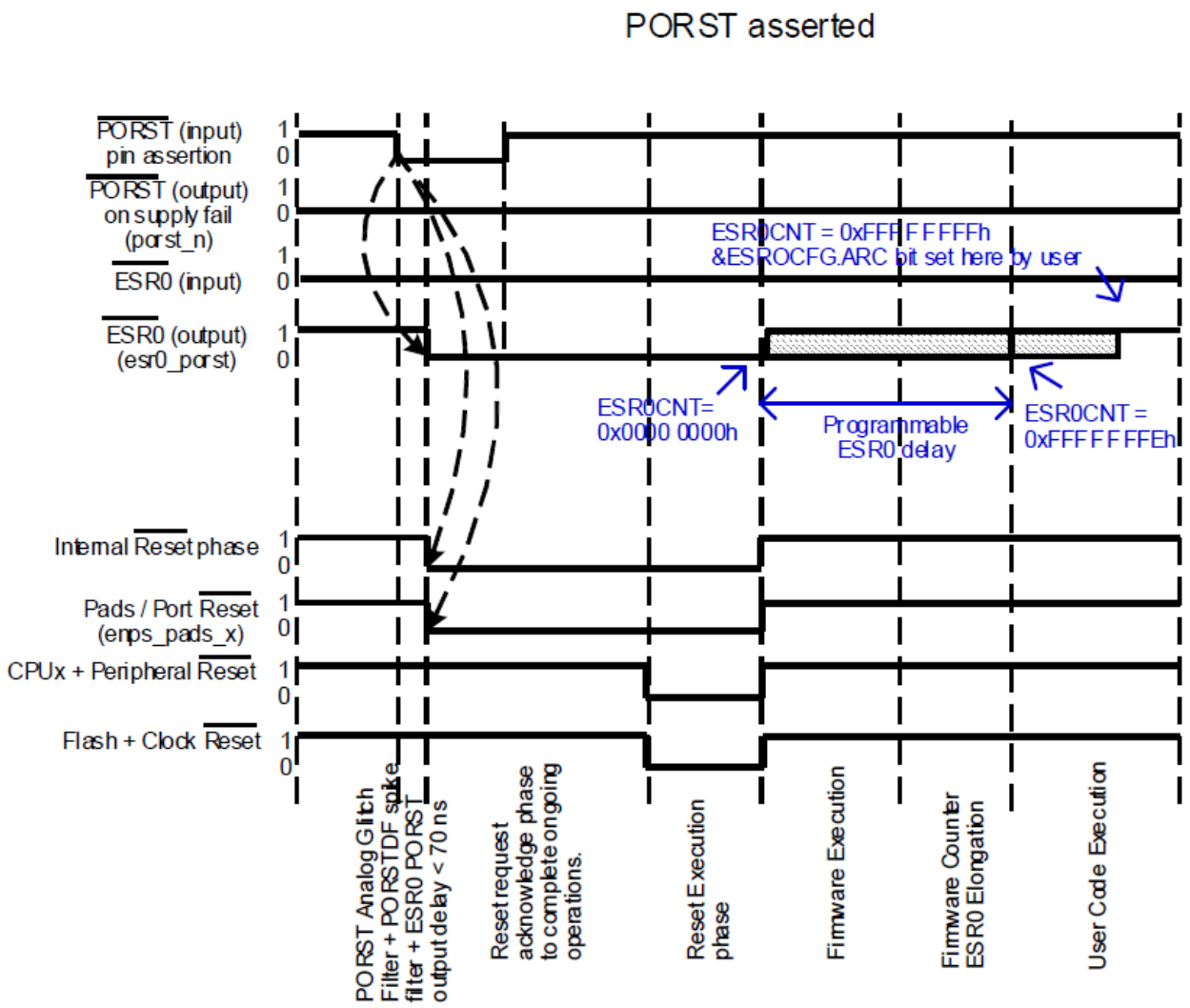
# TC3xx
## Terminology : Warm vs Cold Power On Reset

- A Cold reset is one which is caused owing to
  - Temporary power failures of any of the 3 supply domains
  - Ext. Supply undervoltage < 3.0 V
  - EVR33 undervoltage < 3.0 V
  - EVRC undervoltage < 1.13 V
  - Represents the status at startup

› A **Warm** reset is one which is triggered when system is operational and **supplies are stable** namely
  - RAM data reliability is ensured during Reset phase
  - Ongoing module operations are brought to an end before reset issue

| Reset Type | Reset Trigger(s) |
|---|---|
| **Cold Power-on Reset** | ▪Startup<br>▪Pre reg./ Standby supply falls below 1.13V |
| **Cold Power-on Reset** | ▪EVRC supply falls below 1.13V |
| **Cold Power-on Reset** | ▪EVR33 supply below 3.0V |
| **Cold Power-on Reset** | ▪External supply (SWD) falls below 3.0V |
| **Warm Power-on Reset** | ▪PORST pad asserted |

# TC3xx
# PORST assertion , ESR0 elongation ..

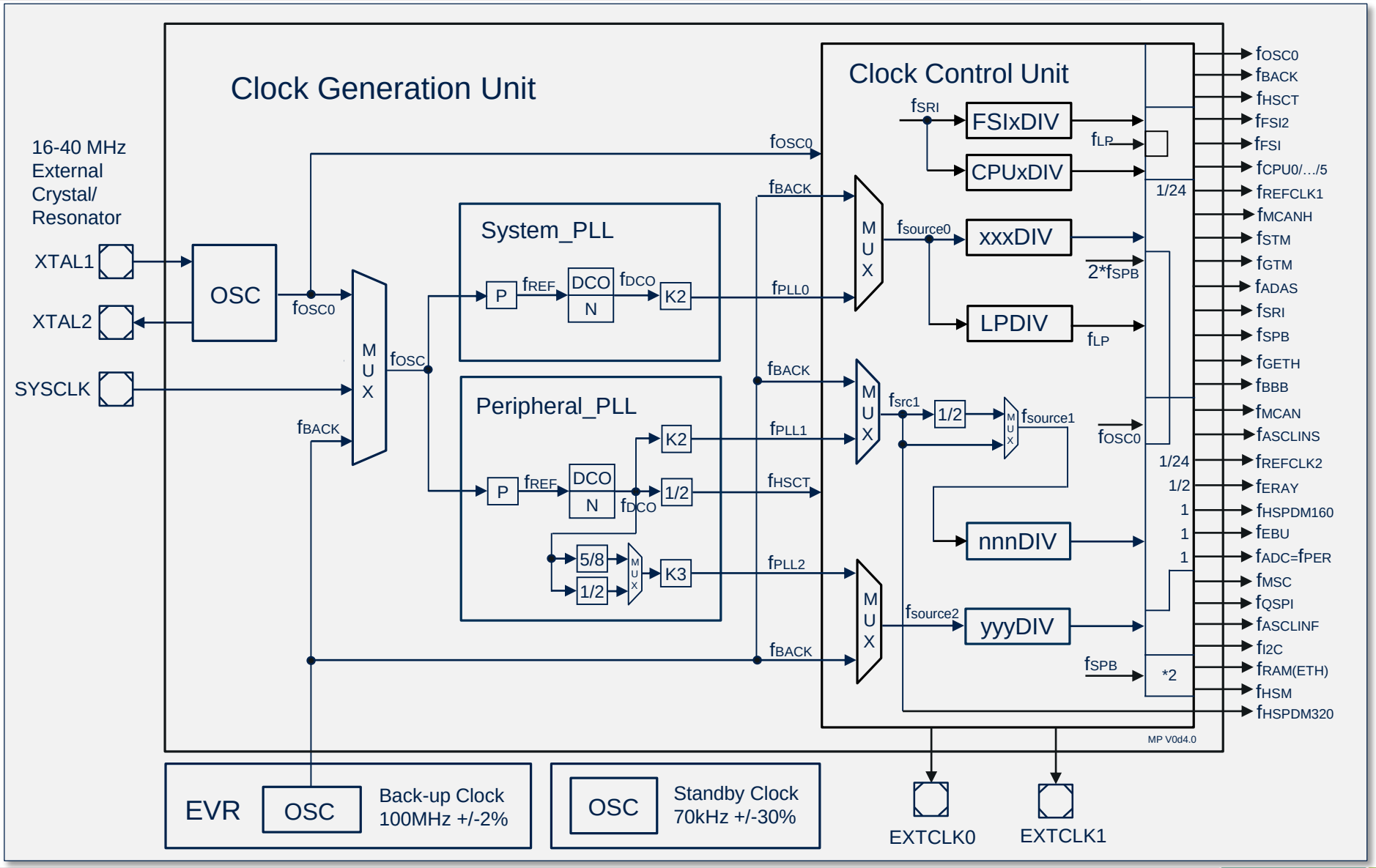



- ESR0 asserted @ PORST recognition

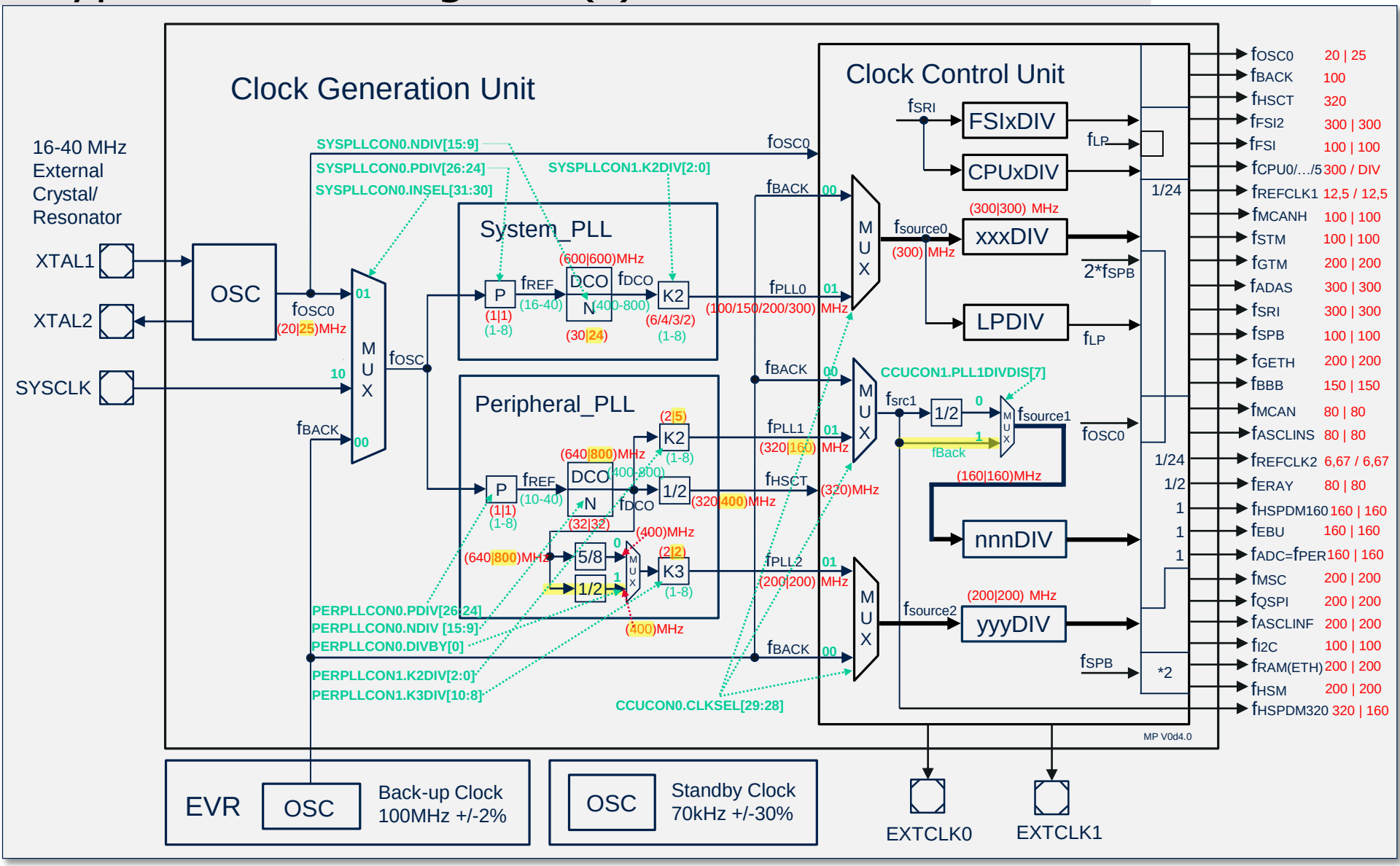
ESR0 deassertion delay after PORST deassertion

Additional PORST Glitch filter – 500ns

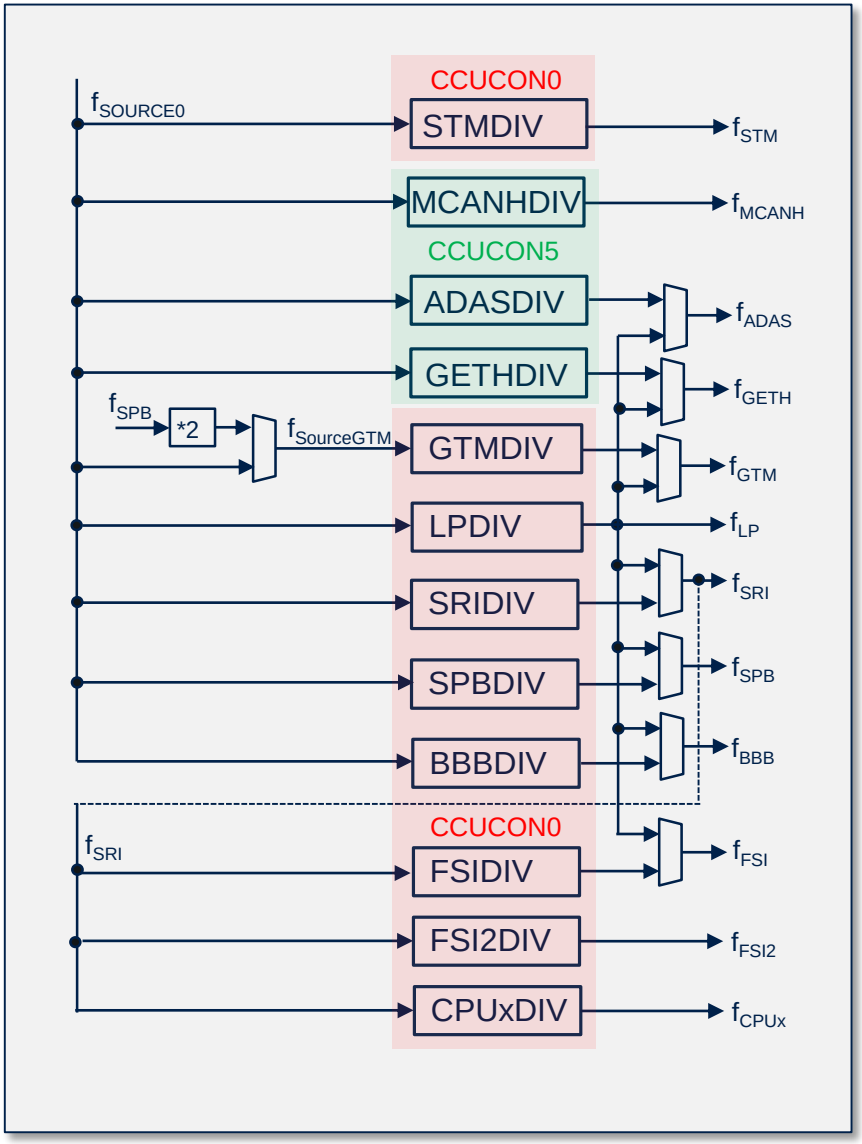
# TC3xx Clock Generation Typical Block Diagram (I)

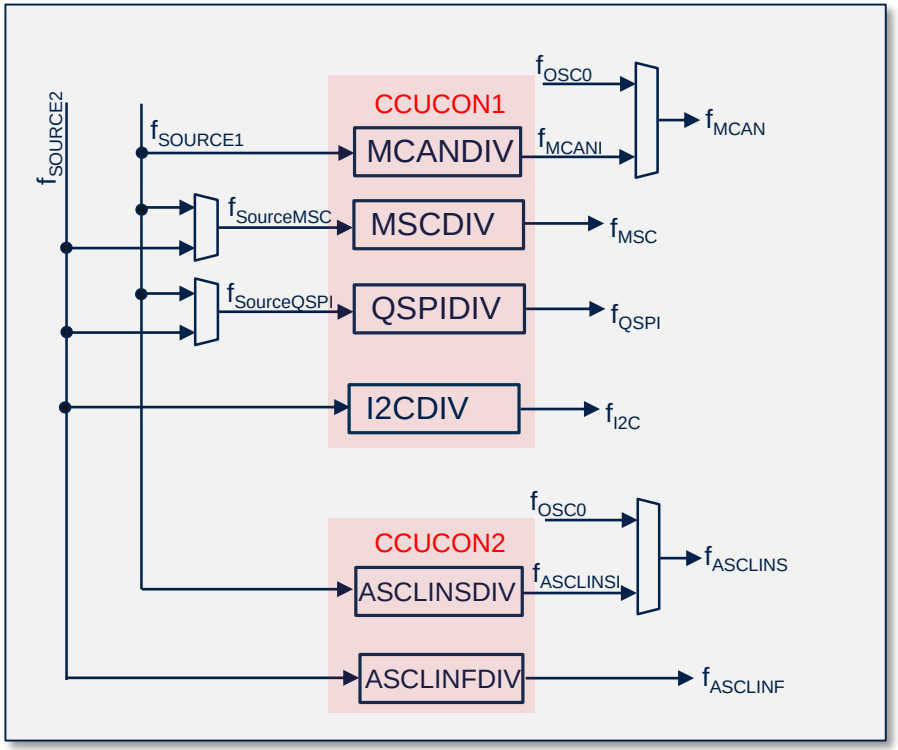


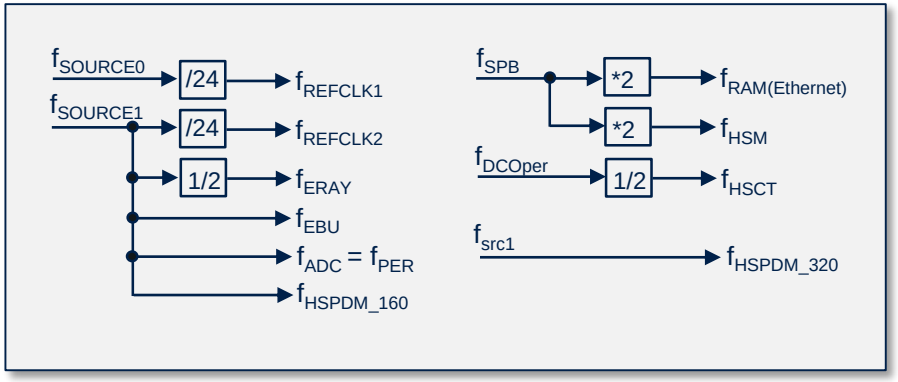
# TC3xx Clock Generation
# Typical Block Diagram (I) with Details



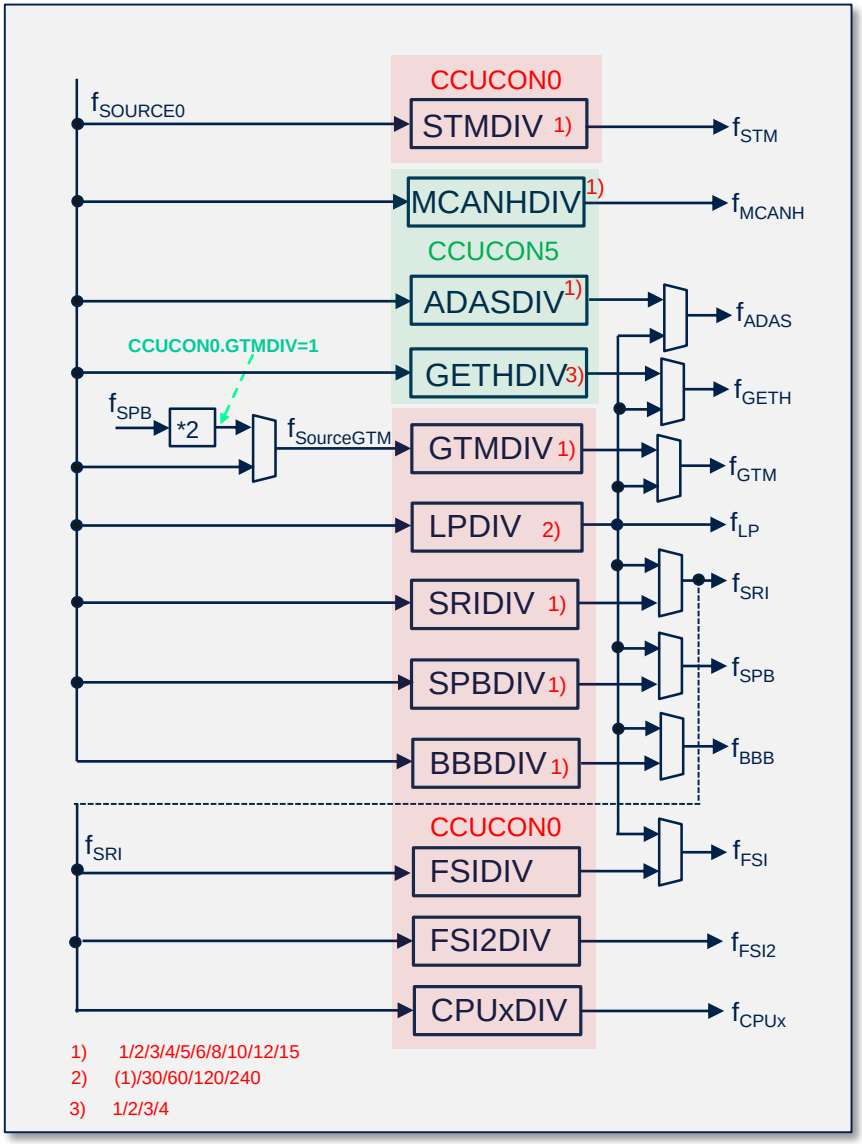
# TC3xx Clock Generation
# Typical Block Diagram (II)

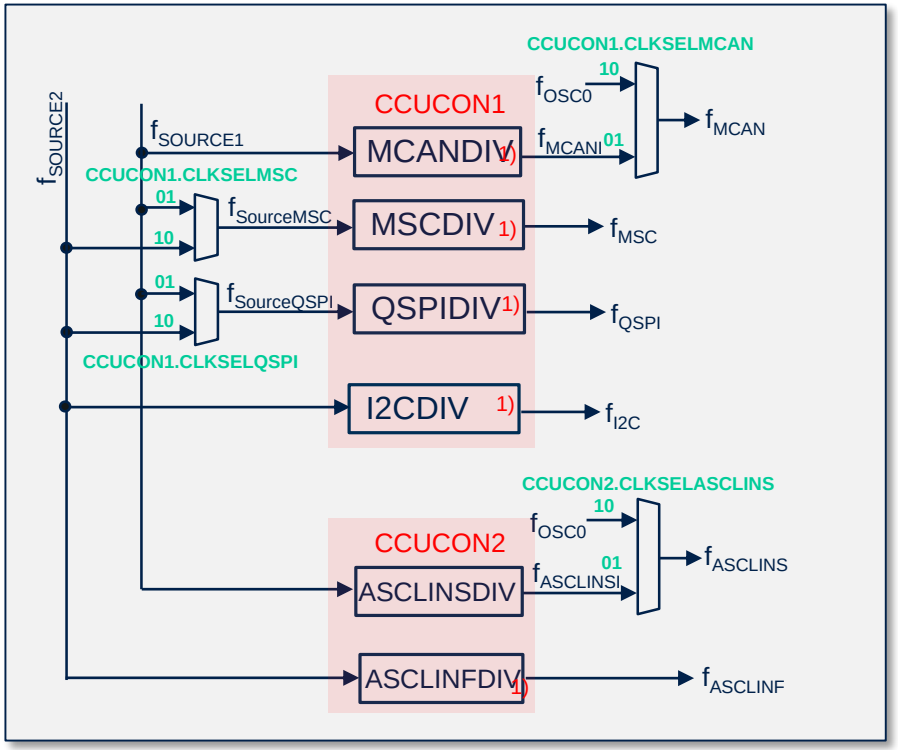


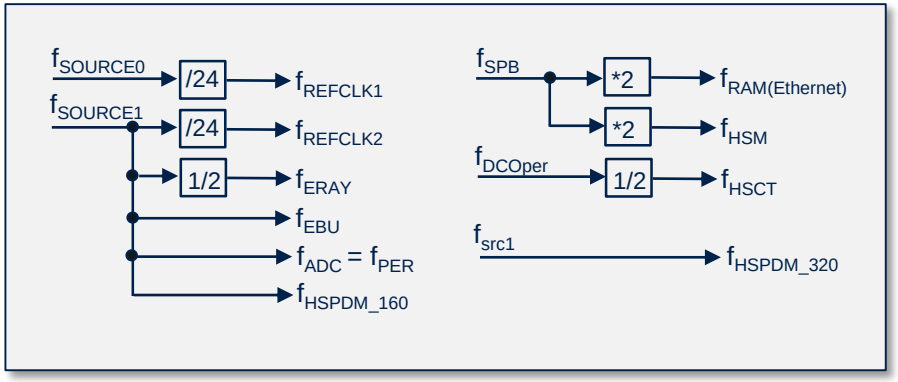



# TC3xx Clock Generation
# Typical Block Diagram (II) with Details



# TC3xx Clock Generation
# Clock Sources

- Different clock sources can be selected
- Separate performance selection for CPU(s), busses, and peripherals. CPU clock frequency (performance) of every single CPU can be changed without changing clock of any peripheral
- Options for constant clock for timers (STM / GTM) & communication interfaces (QSPI / LIN / …) & ADCs while CPU and bus frequencies (performance) can be changed to different values
- Limited current transients resulting from frequency changes in single steps up to system target frequency

# TC3xx
# Oscillator Details

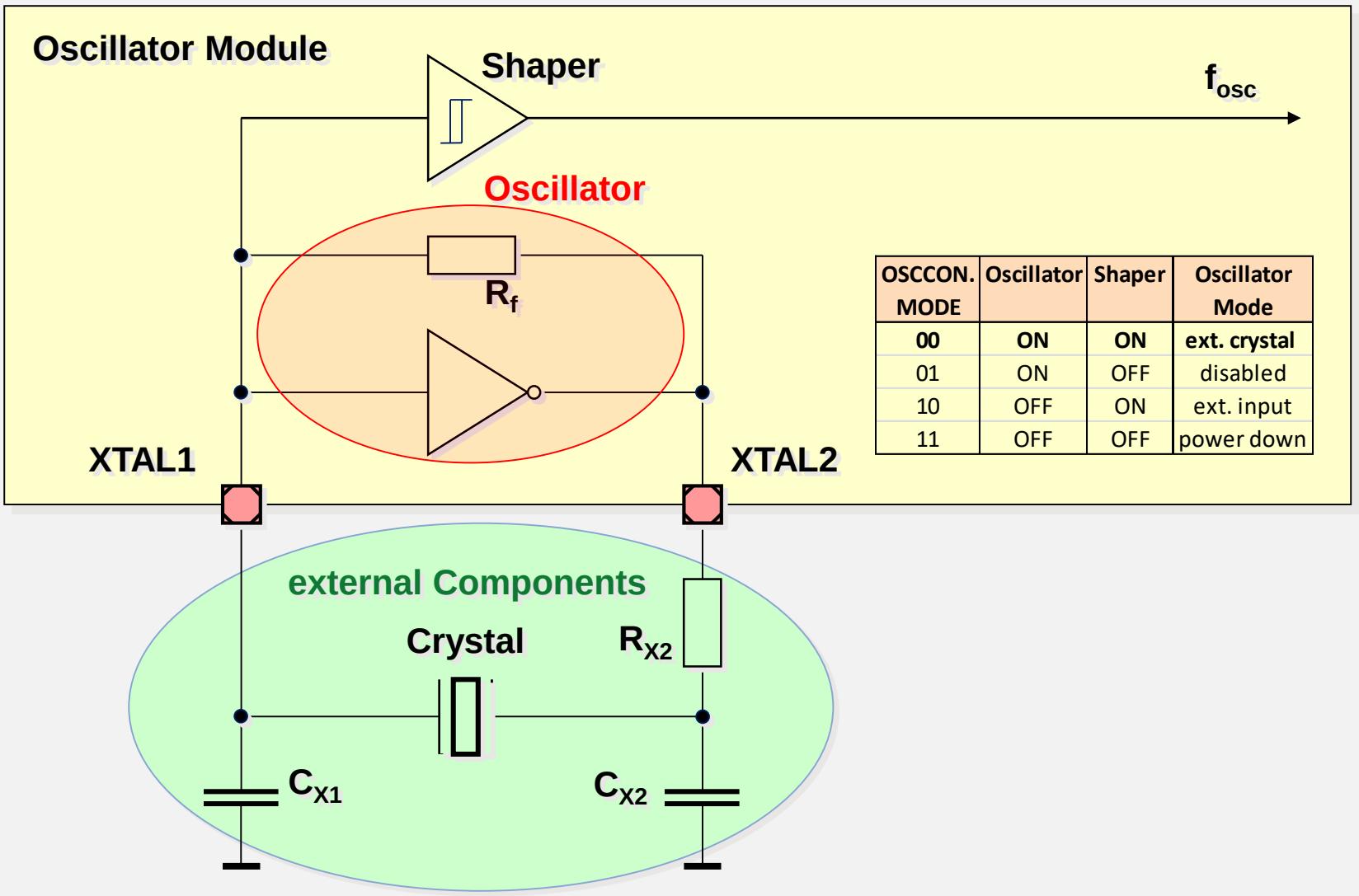
- Crystal oscillator is set to External Input Clock Mode at power-on and has to be set to External Crystal Mode:
  - with direct access to OSCCON.MODE=0
  - or via UCB configuration (➔PROCONDF)
- TC3xx crystal oscillator supports a $f_{OSC}$ frequency range from (16…40)MHz.[1)]
- Oscillator gain of TC3xx oscillator is similar to TC2xx oscillator.[1)]
- Values of the internal load capacitors are changed due to support also crystals with a higher specified load capacitance.[2)]

| Parameter | Symbol | Values | | | Unit | Note / Test Condition |
|---|---|---|---|---|---|---|
| | | Min. | Typ. | Max. | | |
| Internal load capacitor | $C_{L0}$ CC | 1.30 | 1.40 | 1.55 | pF | enabled via bit OSCCON.CAP0EN |
| Internal load capacitor | $C_{L1}$ CC | 3.05 | 3.35 | 3.70 | pF | enabled via bit OSCCON.CAP1EN |
| Internal load capacitor | $C_{L2}$ CC | 7.85 | 8.70 | 9.55 | pF | enabled via bit OSCCON.CAP2EN |
| Internal load capacitor | $C_{L3}$ CC | 12.05 | 13.35 | 14.65 | pF | enabled via bit OSCCON.CAP3EN |

[1)] TC39x AA excluded because of Errata "XTAL_TC.H001 External Oscillator Configuration"

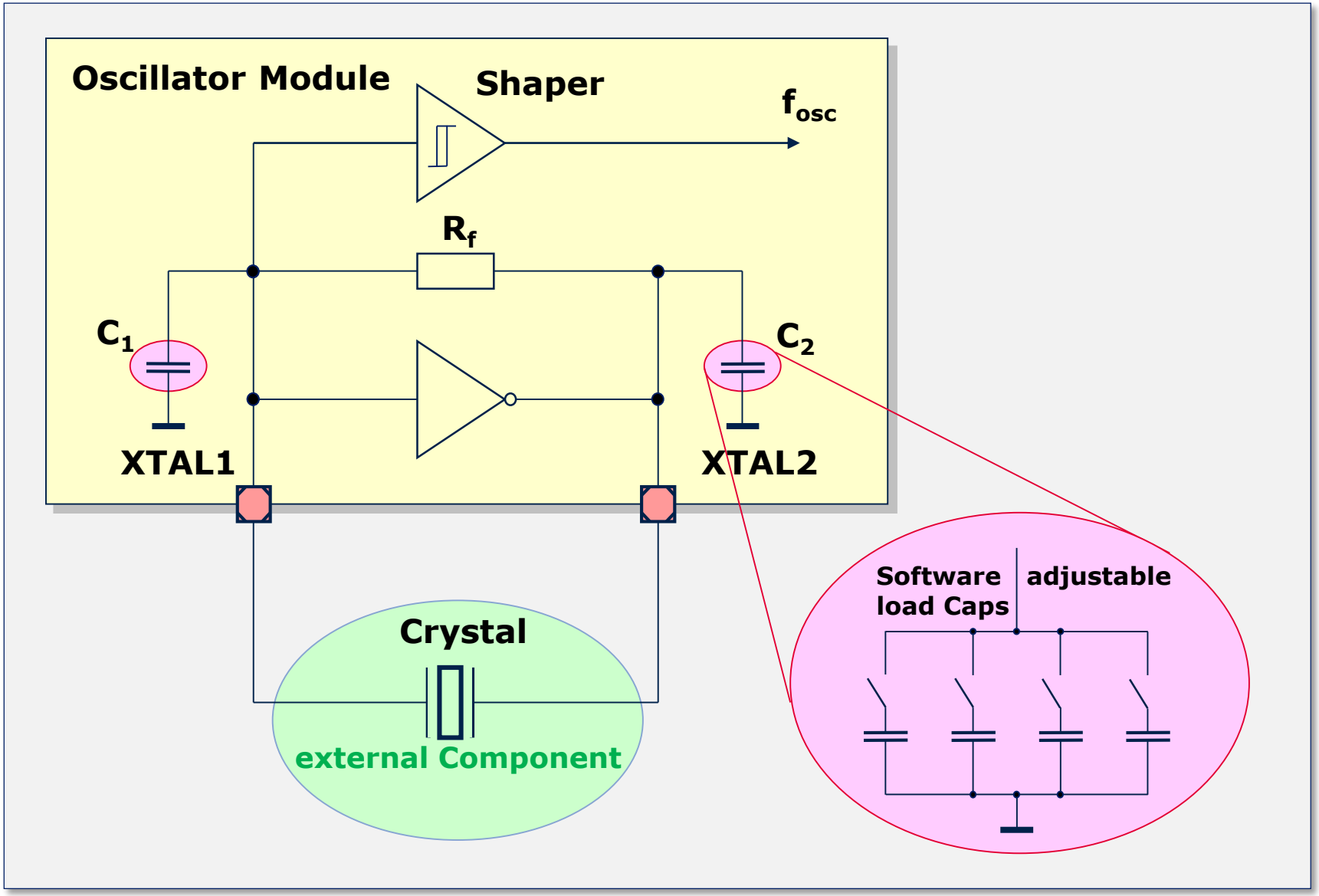
[2)] TC39x AA excluded

# TC3xx Oscillator Circuitry with ext. Load Capacitors (compatible Mode)

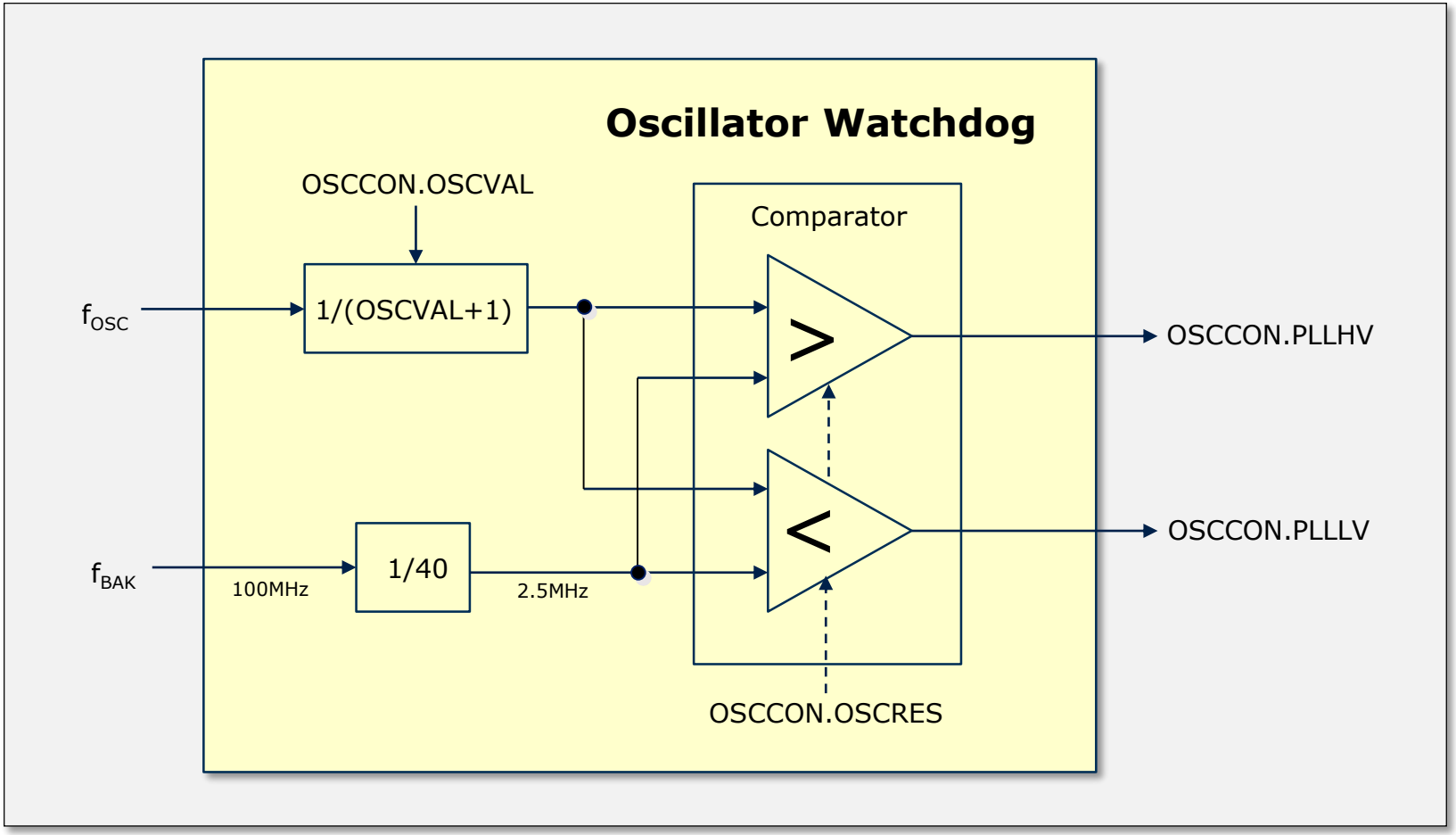



| OSCCON. MODE | Oscillator | Shaper | Oscillator Mode |
|---|---|---|---|
| **00** | **ON** | **ON** | **ext. crystal** |
| 01 | ON | OFF | disabled |
| 10 | OFF | ON | ext. input |
| 11 | OFF | OFF | power down |

# TC3xx Oscillator Circuitry with on-Chip Load Capacitors (Low BOM Mode)

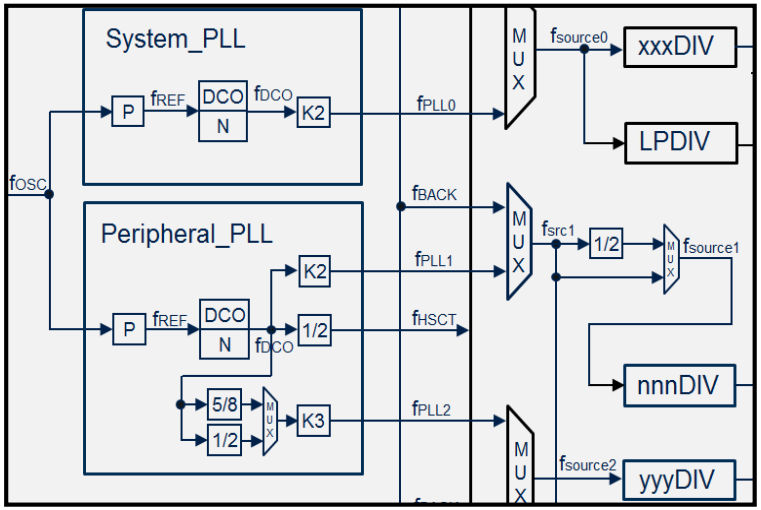


# TC3xx Oscillator Circuitry
# Oscillator Watchdog Block Diagram



# TC3xx
# PLL Configuration

| TC3xx System PLL | | | | | |
|---|---|---|---|---|---|
| crystal [MHz] | P | N | fdco [MHz] | K2 | fpll0 [MHz] |
| 20 | 1 | 30 | 600 | 6/4/3/2 | 100/150/200/300 |
| 25 | 1 | 24 | 600 | 6/4/3/2 | 100/150/200/300 |
| 40 | 2 | 30 | 600 | 6/4/3/2 | 100/150/200/300 |
| 40 | 1 | 15 | 600 | 6/4/3/2 | 100/150/200/300 |




| TC3xx Peripheral PLL | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| crystal [MHz] | P | N | fdco [MHz] | K2 | fpll1 [MHz] | PLL1DIVDIS | fsource1 [MHz] | DIVBY | K3 | fpll2 [MHz] |
| 20 | 1 | 32 | 640 | 2 | 320 | 0 (*1/2) | 160 | 0 (*5/8) | 2 | 200 |
| 25 | 1 | 32 | 800 | 5 | 160 | 1 (*1) | 160 | 1 (*1/2) | 2 | 200 |
| 40 | 2 | 32 | 640 | 2 | 320 | 0 (*1/2) | 160 | 0 (*5/8) | 2 | 200 |
| 40 | 1 | 16 | 640 | 2 | 320 | 0 (*1/2) | 160 | 0 (*5/8) | 2 | 200 |

# TC3xx
# PLL Configuration Principle

- Configuration Principle
  - Enable crystal oscillator (when using an external crystal / resonator)
  - Set Clock Control Unit input clocks to backup clock $f_{BAK}$
  - Select PLL input clock $f_{OSC}$
  - Start and initialize System PLL and Peripheral PLL then wait for lock
  - Initialize CCUCONx register
  - Set Clock Control Unit input clocks to PLL clocks
  - Increase clock frequency $f_{PLL0}$ (CPU clock) step by step to target frequency

- Note, that PLL reference frequency may not exceed specified values:
  - System PLL : $f_{REF\_min}$ = 16MHz, $f_{REF\_max}$ = 40MHz
  - Peripheral PLL: $f_{REF\_min}$ = 10MHz, $f_{REF\_max}$ = 40MHz
  - $f_{REF} = f_{OSC} / P$

# TC3xx Clock Restrictions
# Clock Frequency

| Parameter | Symbol | Values | | | Unit |
|---|---|---|---|---|---|
| | | Min. | Typ. | Max. | |
| SRI frequency | $f_{SRI}$ SR | - | - | 300 | MHz |
| CPU Frequency (All CPUs) | $f_{CPUx}$ SR | - | - | 300 | MHz |
| PLL0 output frequency | $f_{PLL0}$ SR | 50 | - | 300 | MHz |
| SPB frequency | $f_{SPB}$ SR | - | - | 100 | MHz |
| FSI2 frequency | $f_{FSI2}$ SR | - | - | 300 | MHz |
| FSI frequency | $f_{FSI}$ SR | 20 | - | 100 | MHz |
| GTM frequency | $f_{GTM}$ SR | - | - | 200 | MHz |
| STM frequency | $f_{STM}$ SR | - | - | 100 | MHz |
| ERAY frequency | $f_{ERAY}$ SR | - | 80 | - | MHz |
| BBB frequency | $f_{BBB}$ SR | - | - | 150 | MHz |
| VADC frequency | $f_{ADC}$ SR | - | - | 160 | MHz |
| ASCLIN Operating Frequency | $f_{ASCLINx}$ SR | - | - | 200 | MHz |
| CAN frequency | $f_{CAN}$ SR | - | - | 80 | MHz |
| I2C frequency | $f_{I2C}$ SR | - | - | 100 | MHz |
| Operating MSC Frequency | $f_{MSC}$ SR | - | - | 200 | MHz |
| PLL1 output frequency from PER PLL | $f_{PLL1}$ SR | 50 | - | 320 | MHz |
| PLL2 output frequency from PER PLL | $f_{PLL2}$ SR | 25 | - | 200 | MHz |
| QSPI Frequency | $f_{QSPI}$ SR | - | - | 200 | MHz |
| ADAS clock frequency | $f_{ADAS}$ CC | 200 | - | 300 | MHz |
| MCANH frequency | $f_{MCANH}$ CC | - | - | 100 | MHz |
| GETH frequency | $f_{GETH}$ CC | 150 | - | 200 | MHz |
| EBU operating frequency | $f_{EBU}$ SR | - | - | 160 | MHz |

→ only TC39x

# TC3xx Clock Restrictions
# CCU Clock Options

| CCU Clock Output | Clock Source | | | | |
|---|---|---|---|---|---|
| | System PLL ($f_{PLL0}$) | Peripheral PLL ($f_{PLL1}$) | Peripheral PLL ($f_{PLL2}$) | Back-up ($f_{BACK}$) | OSC_XTAL ($f_{OSC0}$) |
| $f_{SRI}$ | ✓ | – | – | Default | – |
| $f_{CPUx}$ | $f_{SRI}$ | – | – | – | – |
| $f_{SPB}$ | ✓ | – | – | Default | – |
| $f_{FSI}$ | $f_{SRI}$ | – | – | – | – |
| $f_{FSI2}$ | $f_{SRI}$ | – | – | – | – |
| $f_{REFCLK1}$ | ✓ | – | – | Default | – |
| $f_{REFCLK2}$ | – | ✓ | – | Default | – |
| $f_{BBB}$ | ✓ | – | – | Default | – |
| $f_{ERAY}$ | – | ✓ | – | – | – |
| $f_{GTM}$ | ✓ | – | – | Default | – |
| $f_{STM}$ | ✓ | – | – | Default | – |
| $f_{MSC}$ | – | ✓ | ✓ | Default | – |
| $f_{GETH}$ | ✓ | – | – | Default | ✓ |
| $f_{ADAS}$ | ✓ | – | – | Default | ✓ |
| $f_{MCANH}$ | ✓ | – | – | Default | ✓ |
| $f_{MCAN}$ | – | ✓ | – | Default | ✓ |
| $f_{ASCLINF}$ | – | – | ✓ | Default | – |
| $f_{ASCLINS}$ | – | ✓ | – | Default | ✓ |
| $f_{QSPI}$ | – | ✓ | ✓ | Default | – |
| $f_{ADC}$ | – | ✓ | – | Default | – |
| $f_{I2C}$ | – | – | ✓ | Default | – |
| $f_{EBU}$ | – | ✓ | – | Default | – |
| $f_{HSPDM\_160}$ | – | ✓ | – | Default | – |
| $f_{HSPDM\_320}$ | – | ✓ | – | Default | – |

# TC3xx Clock Restrictions
# CCU allowed Clock Ratios

| Clock A | Clock B | Allowed Ratios | Notes | Recommended Default |
|---|---|---|---|---|
| $f_{SRI}$ | $f_{SPB}$ | 1 : n | n = 1, 2, 3, 4, 5, 6,... | n = 2 or 3 |
| $f_{SRI}$ | $f_{FSI}$ | 1 : n | n = 1, 2, 3 | n = 2 or 3 |
| $f_{SRI}$ | $f_{FSI2}$ | 1 : n | n = 1, 2, 3 | n = 1 |
| $f_{FSI2}$ | $f_{FSI}$ | 1 : n | n = 1, 2, 3 | n = 2 or 3 |
| $f_{FSI}$ | $f_{SPB}/2$ 1) | 1 : n | n = 1, 2, 3, 4, 5, 6,... | n = 2 |
| $f_{GTM}$ | $f_{SPB}$ | 1 : n | n = 1, 2 | n = 2 |
| $f_{SPB}$ | $f_{GTM}$ | 1 : n | n = 1, 2, 3, 4, 5, 6 | n = 2 |
| $f_{SPB}$ | $f_{STM}$ | 1 : n 2) | n = 1, 2, 3, 4, 5, 6 | n = 1 |
| $f_{STM}$ | $f_{SPB}$ | 1 : n 2) | n = 1, 2, 3, 4, 5, 6,... | n = 1 |
| $f_{SRI}$ | $f_{BBB}$ 3) | 1 : n | n = 1, 2 | n = 2 |
| $f_{SRI}$ | $f_{GETH}$ | 1 : n | n = 1, 2, 3, 4, 5, 6,... | n = 2 |
| $f_{SPB}$ | $f_{MCANH}$ | 1 : n | n = 1 4) | n = 1 |
| $f_{MCANH}$ | $f_{MCAN}$ | 1 : n 5) | n >= 1.0 | n = 1.0 |
| $f_{ADAS}$ | $f_{SRI}$ | 1 : n | n = 1, 2 | n = 1 |
| $f_{ADAS}$ | $f_{BBB}$ | 1 : n 6) | n = 2 | n = 2 |

1) for FSI module, SPBhalf clock is internally used, hence relevant for clock ratios
2) $f_{STM}$ can be faster, slower, or equal to $f_{SPB}$
3) $f_{BBB}$ has to be slower, or equal to $f_{SRI}$
4) restriction only valid when not in pretended networking / LP mode
5) n is not an integer as the related clocks are asynchronous to each other
6) $f_{BBB}$ has to be half the SPU frequency $f_{ADAS}$

*Note: Recommended values did not necessarily reflect the default configuration after a reset event. Instead they should give a hint how to configure the system for an optimal performance. In addition, it may happen that applying an allowed ratio on two selected clocks violates the ratio of others. Due to its complexity, this cannot be shown here in all combinations. The user has to take care not to violate clock ratios depending on the use case, i.e. enabled clocks.*

# TC3xx
# Example of a Clock Ramp-up Sequence (I)

- The example below shows one possibility to initialize the clock system with a XTAL1 input frequency of 20MHz
  - System PLL is set to
    - $f_{SOURCE0}$ = 300MHz (CPU frequency)
  - Peripheral PLL is set to
    - $f_{SOURCE1}$ = 160MHz (ADC frequency)
    - $f_{SOURCE2}$ = 200MHz

**Example of a Clock Ramp-up Sequence:**

- enable crystal oscillator and wait for reliable XTAL1 clock
- select $f_{BACK}$ as CCU input clock
- select $f_{OSC0}$ ($f_{XTAL1}$) as SYS-PLL and PER-PLL input clock
- initialize System PLL:
  - start System PLL and wait 1ms
  - $P_{SYS}$ = 1, $N_{SYS}$ = 30 ($f_{DCO}$ = 600MHz), $K2_{SYS}$ = 6 ($f_{PLL0}$ = 100MHz)
  - wait for lock

# TC3xx
# Example of a Clock Ramp-up Sequence (II)

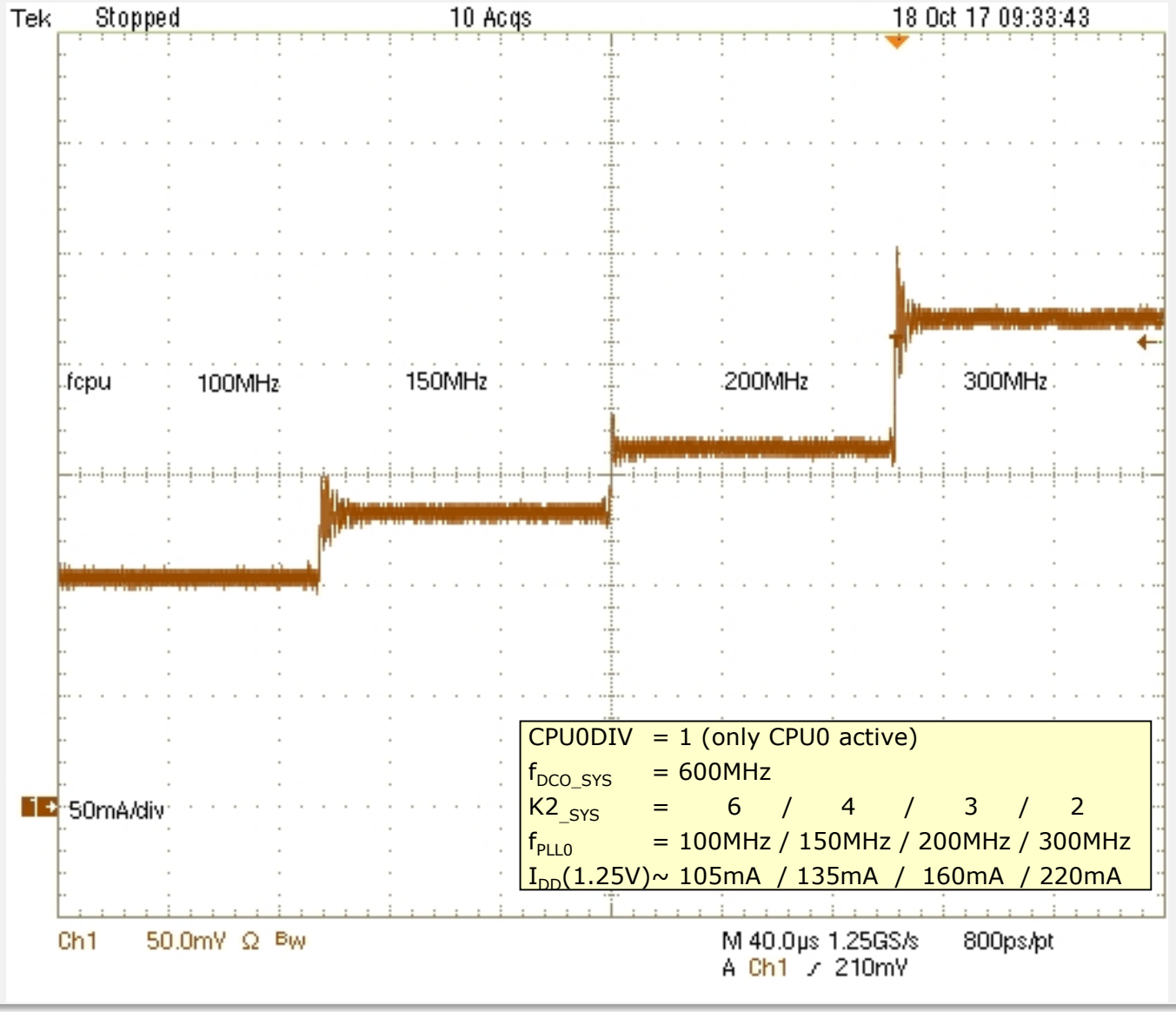
- initialize Peripheral PLL:
  - start Peripheral PLL and wait 1ms
  - $P_{PER}$ = 1, $N_{PER}$ = 32 ($f_{DCO}$ = 640MHz), $K2_{PER}$ = 2 ($f_{PLL1}$ = 320MHz)
  - DIVBY=0, $K3_{PER}$ = 2 ($f_{PLL2}$ = 200MHz)
  - wait for lock
- set CCUCONx to desired values
- switch CCU input clocks $f_{SOURCE0/1/2}$ to $f_{PLL0/1/2}$ via CCUCON0.CLKSEL

- set $K2_{SYS}$ = 4 ($f_{SOURCE0}$ = 150MHz
- wait ~100µs (wait until supply ripple caused by increased supply current is faded away)

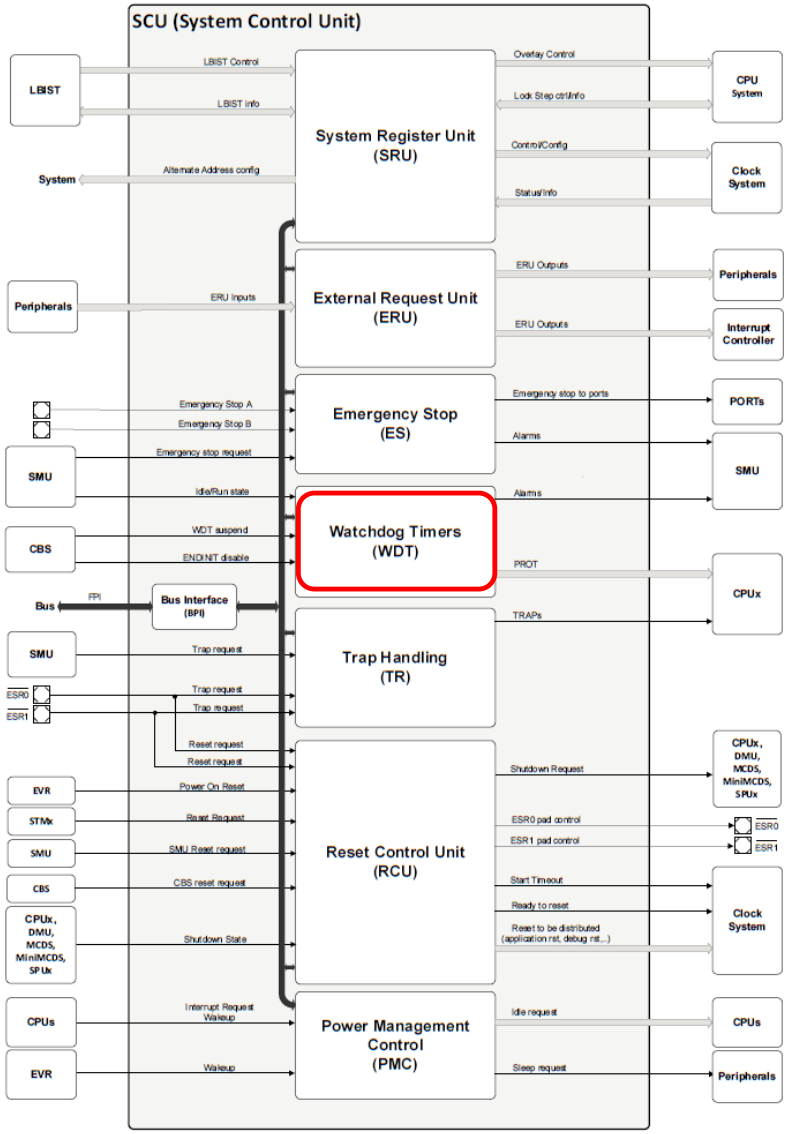
- set $K2_{SYS}$ = 3 ($f_{SOURCE0}$ = 200MHz
- wait ~100µs (wait until supply ripple caused by increased supply current is faded away)

- set $K2_{SYS}$ = 2 ($f_{SOURCE0}$ = 300MHz
- wait ~100µs (wait until supply ripple caused by increased supply current is faded away

# TC389 AA-EES: Clock Ramp-up Sequence
# Typical Supply Current Idd @ Clock Ramp-up

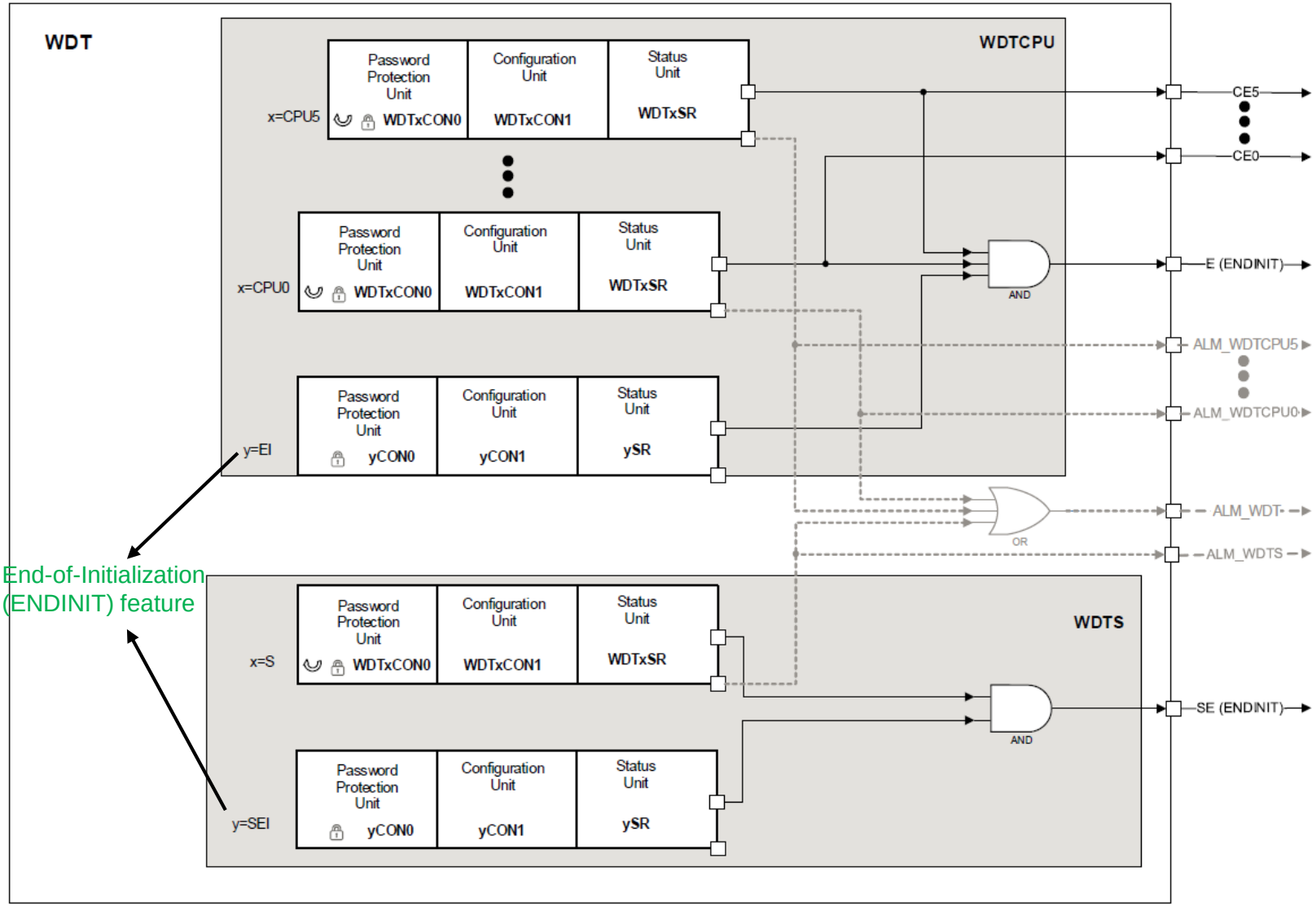


# Watchdog timers

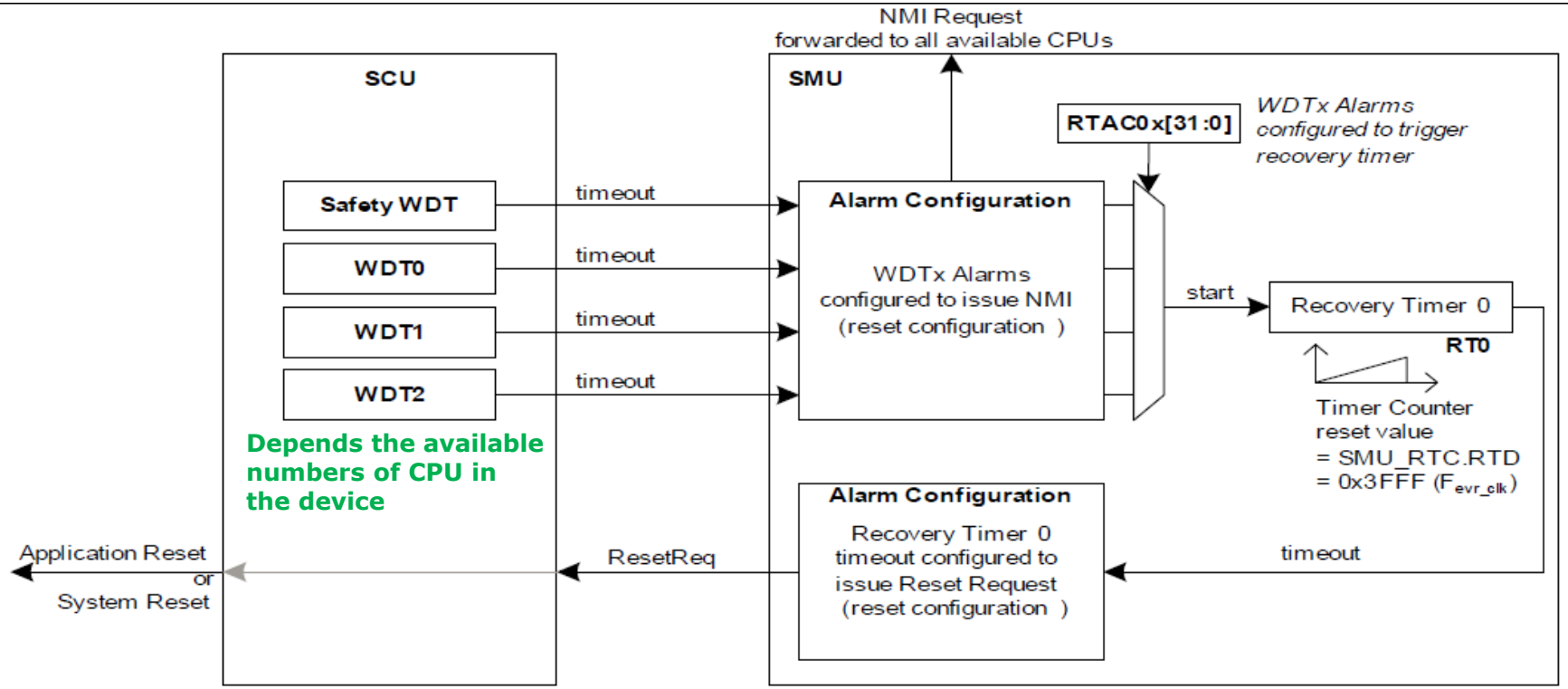



- One Safety Watchdog Timer
- One Watchdog Timer per CPU

# WDT Block Diagram



# Safety and CPU Watchdogs
# Enhanced Software Flow Control monitoring




- Safety watchdog is an overall system level watchdog.
- Individual CPU watchdog monitors the individual CPU execution threads.
- Challenge – Response watchdog service through rolling seed value based on
  - Automatic Password sequencing based on a 14 bit Fibonacci LFSR
  - Temporal Password with actual time stamp or WDT count estimate
- Watchdog interface via pin or SPI to the external monitor

# Watchdog Timer Block

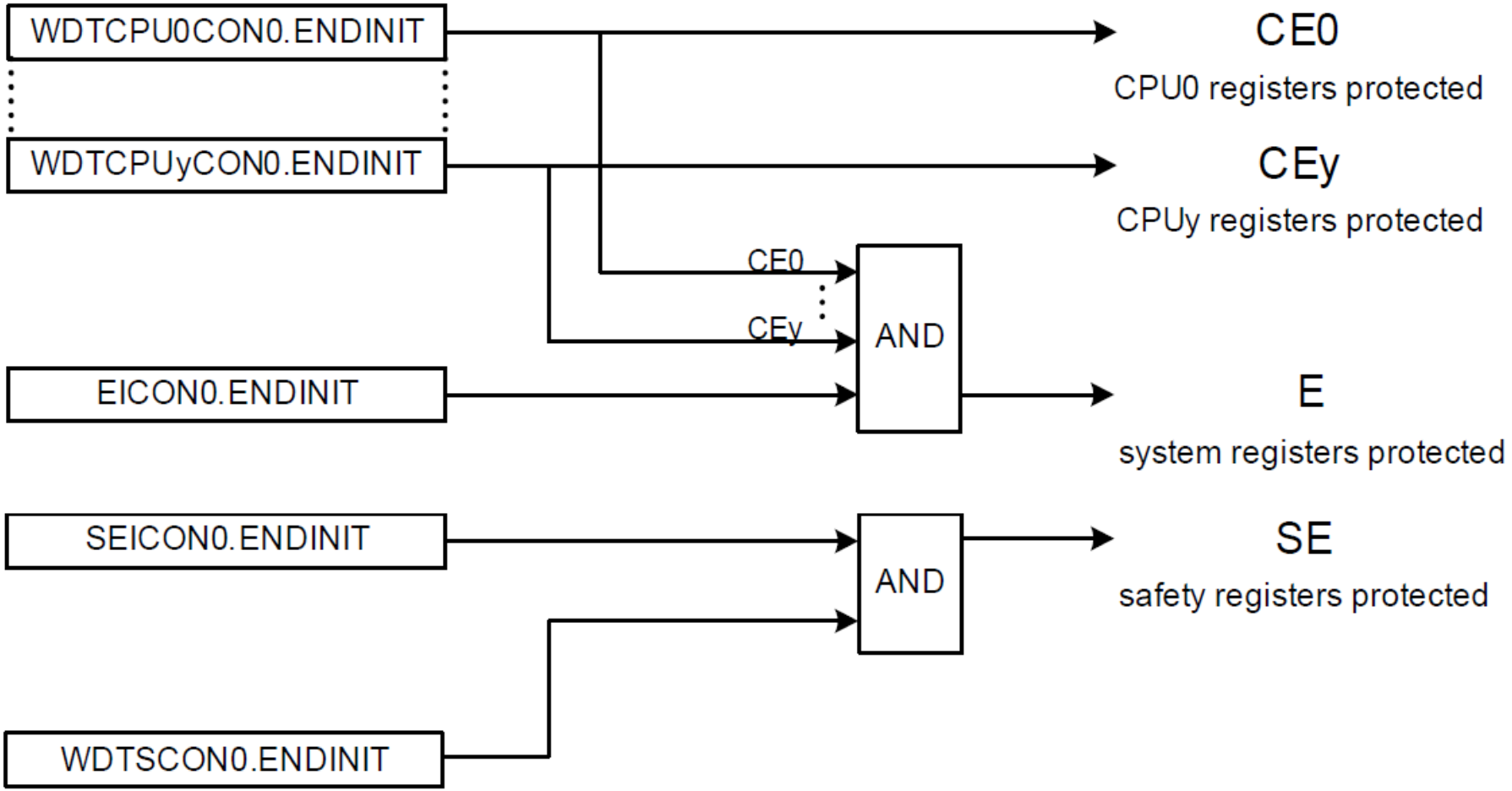
- The Watchdog Timer block controls two functions:
  - EndInit protection : it is a a system wide protection that protects key registers from unwanted write access.
  - Watchdog Timer function : it is a 16 bit timer. It can be disabled.
- Configuration options are available which enable a Watchdog service to additionally check code execution sequence and intermediate code execution time.
  - If these checks are enabled then any incorrect sequence or out-of-limit execution time will also result in a Safety Management Unit (SMU) alarm request.

# Endinit Protection

Endinit protection is the write protection mechanism of key system registers

- Endinit type
  - "CEy"- CPU critical registers. Writeable only when CPUy WDT ENDINIT=0 (y=CPU number)
  - "E" - System critical registers - Writeable when any (one or more) CPUy Watchdog Timer ENDINIT=0 or EICON0.ENDINIT =0
  - "SE" - Safety critical registers - Writeable only when Safety Watchdog Timer ENDINIT=0 or SEICON0.ENDINIT=0
- It can only be *temporarily* disabled via a complex access sequence to a WDT control registers
- The clearing of the ENDINIT bit takes some time.
  - Accesses to EndInit-protected registers after the clearing of the ENDINIT bit must only be done when bit ENDINIT is really cleared.
  - WDT_CON0 (the register with the ENDINIT bit) should be read back once before EndInit-protected registers are accessed the first time after bit ENDINIT has been cleared.

# ENDINIT Control Registers



# Endinit Protected Registers(examples)

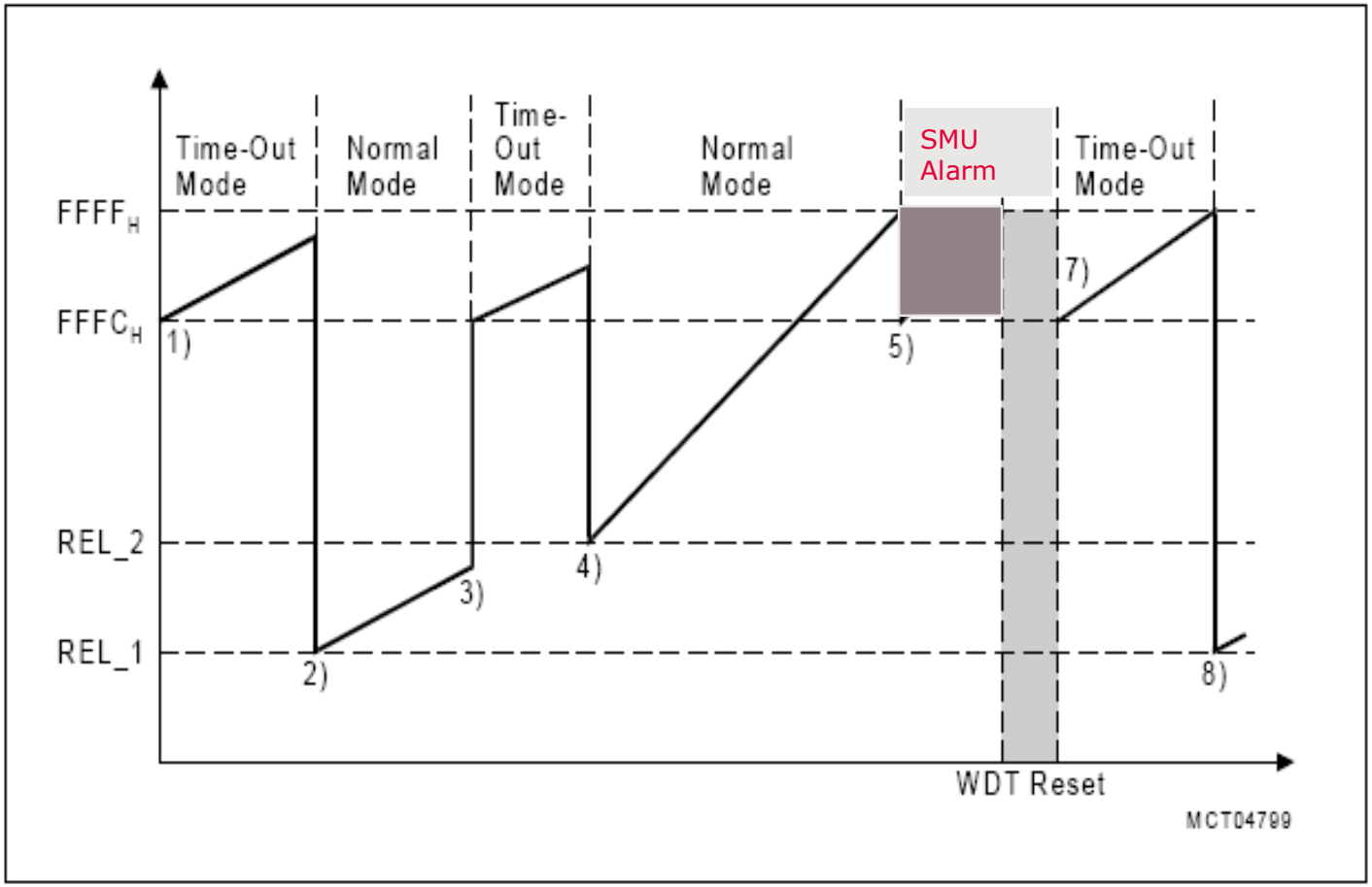
| Short Name | Long Name | Offset Address | Access Mode | | Reset | Page Number |
|---|---|---|---|---|---|---|
| | | | Read | Write | | |
| OSCCON | OSC Control Register | $0010_H$ | U,SV | SV,SE,P0 | See page 6 | 6 |
| SYSPLLSTAT | System PLL Status Register | $0014_H$ | U,SV | BE | See page 13 | 13 |
| SYSPLLCON0 | System PLL Configuration 0 Register | $0018_H$ | U,SV | SV,SE,P0 | System Reset | 14 |
| SYSPLLCON1 | System PLL Configuration 1 Register | $001C_H$ | U,SV | SV,SE,P0 | System Reset | 15 |
| SYSPLLCON2 | System PLL Configuration 2 Register | $0020_H$ | U,SV | SV,SE,P0 | System Reset | 16 |
| PERPLLSTAT | Peripheral PLL Status Register | $0024_H$ | U,SV | BE | System Reset | 19 |
| PERPLLCON0 | Peripheral PLL Configuration 0 Register | $0028_H$ | U,SV | SV,SE,P0 | System Reset | 20 |
| PERPLLCON1 | Peripheral PLL Configuration 1 Register | $002C_H$ | U,SV | SV,SE,P0 | System Reset | 21 |
| CCUCON0 | CCU Clock Control Register 0 | $0030_H$ | U,SV | SV,SE,P0 | See page 27 | 27 |

# Watchdog Timer

Key Features :

- 16 bit counter
- Selectable input frequency: fsys/64, fsys/256 or fsys/16384
- User definable reload value
- Complex password access mechanism
- Access error detection and overflow error detection triggers
  - SMU alarm
  - Interrupt or NMI
  - Reset or stop CPU
- Double reset detection (for safety WDT only)

# Watchdog Timer: Operation Overview .



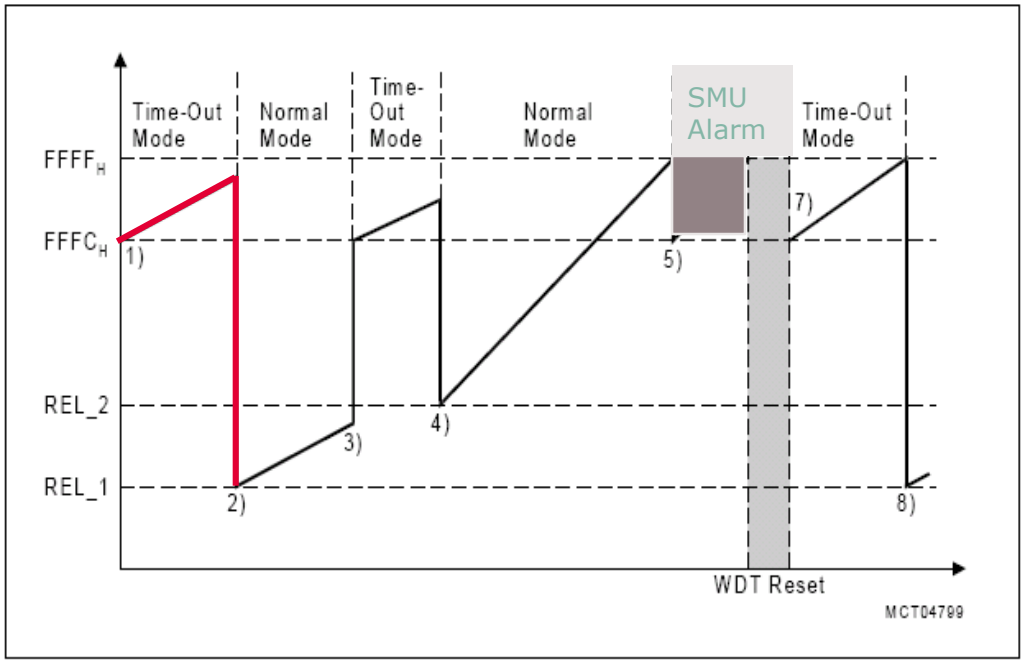
# Watchdog Timer: Operation Overview .

① Time-Out Mode is automatically entered after reset.

- Timer counts with slowest input clock.

② Time-Out Mode terminated and Normal Mode is entered

- Correct password access to WDT_CON0
- Modify access to WDT_CON0 sets the reload value to REL_1 and ENDINIT to 1



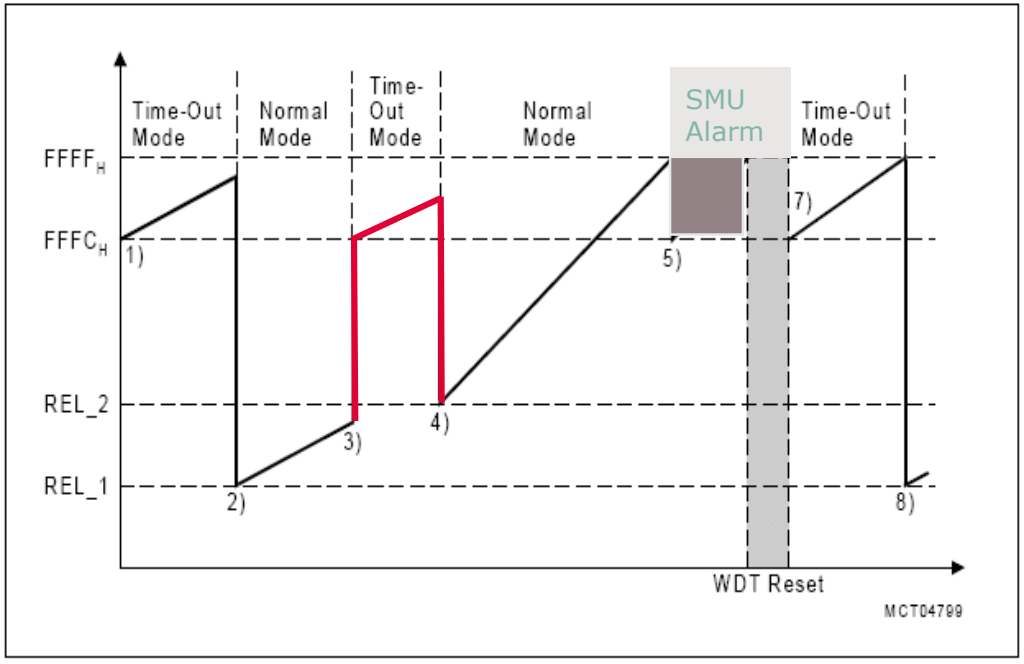
# Watchdog Timer: Operation Overview .

③ Correct password access to WDT_CON0.

- Normal Mode is terminated and Time-Out Mode is entered

④ Modify access to WDT_CON0

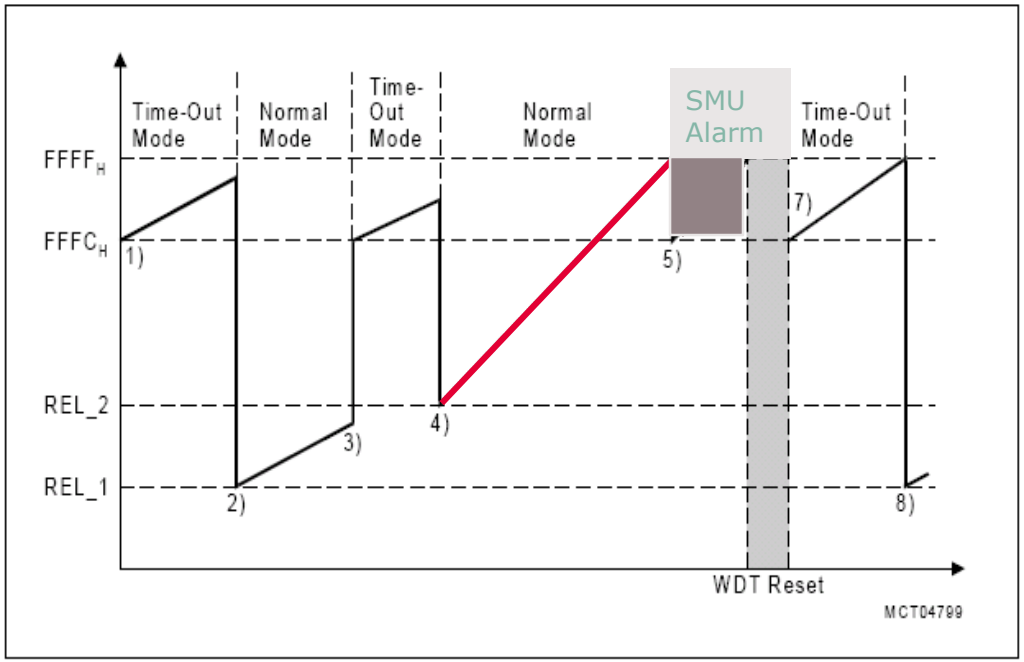
- ENDINIT set to 1
- The reload value WDTREL has been changed to REL_2 and the timer input clock was set to the fast clock.
- Time-Out Mode is terminated and Normal Mode entered again



# Watchdog Timer: Operation Overview

⑤ If the Watchdog Timer is not serviced

- It continues to count upwards until it overflows
- SMU Alarm Mode is entered. Timer counts with selected fast input clock. Watchdog operation cannot be altered or stopped in this mode.



# Access WDT_CON0

- In order to modify WDT_CON0, two write accesses are needed :
  - A *password* access to unlock WDT_CON0.
    - A valid password access is based on the present value of WDT_CON0, the value of WDT_CON1 and some guard bits.
  - A *modify* access to modify WDT_CON0 (e.g. to remove EndInit protection).
    - During the modify access, guard bits are also implemented.

# Access WDT_CON0(cont.)

- Password Access

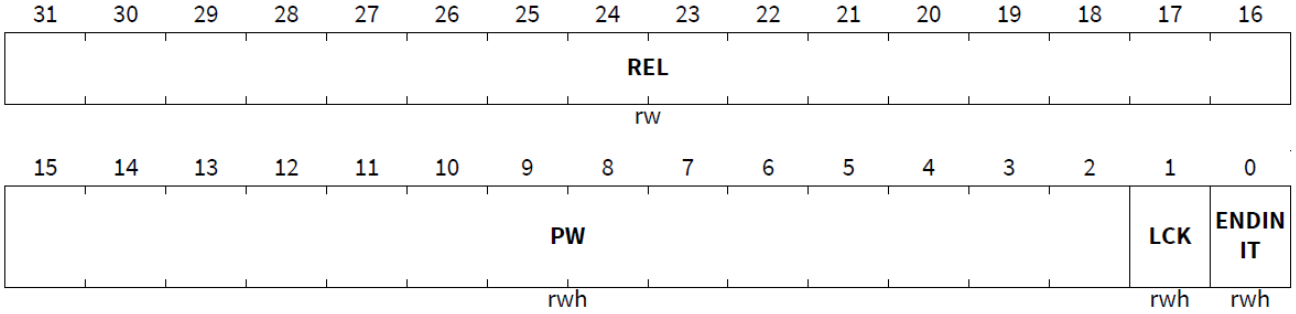



**Table 429 Password Access Bit Pattern Requirements**

| Bit Position | Required Value |
|---|---|
| **[1:0]** | Fixed; must be written to $01_B$ |
| **[15:2]** | If PAS=0: WDTxCON0.PW[7:2] must be written with inverted current value read from WDTxCON0.PW[7:2]<br>WDTxCON0.PW[15:8] must be written with non-inverted current value read from WDTxCON0.PW[15:8]<br><br>If PAS=1:<br>Must be written with Expected Next Sequence Password |
| **[31:16]** | If TCS=0: Must be written with current value of user-definable reload value, WDTxCON0.REL<br>If TCS=1:<br>Must be written with inverted estimate of the WDT count value, WDTxSR.TIM. This value must be within +/- WDTxSR.TCT of the actual value |

# Access WDT_CON0(cont.)

- Modify Access

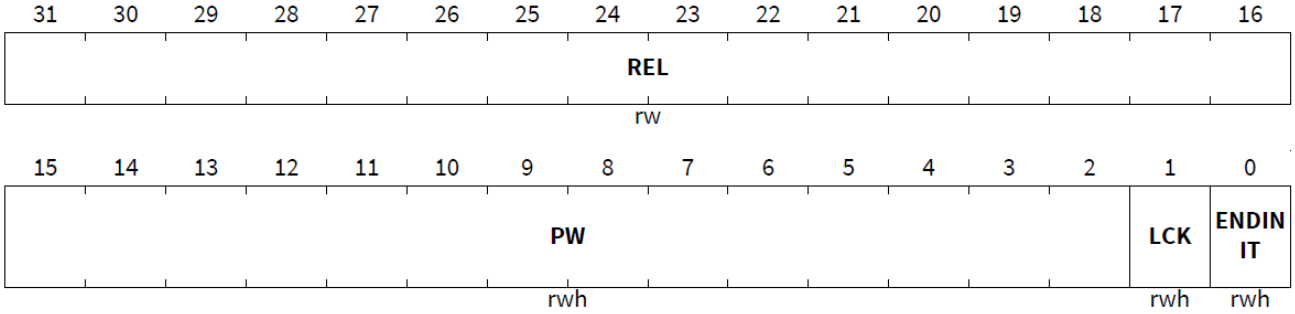


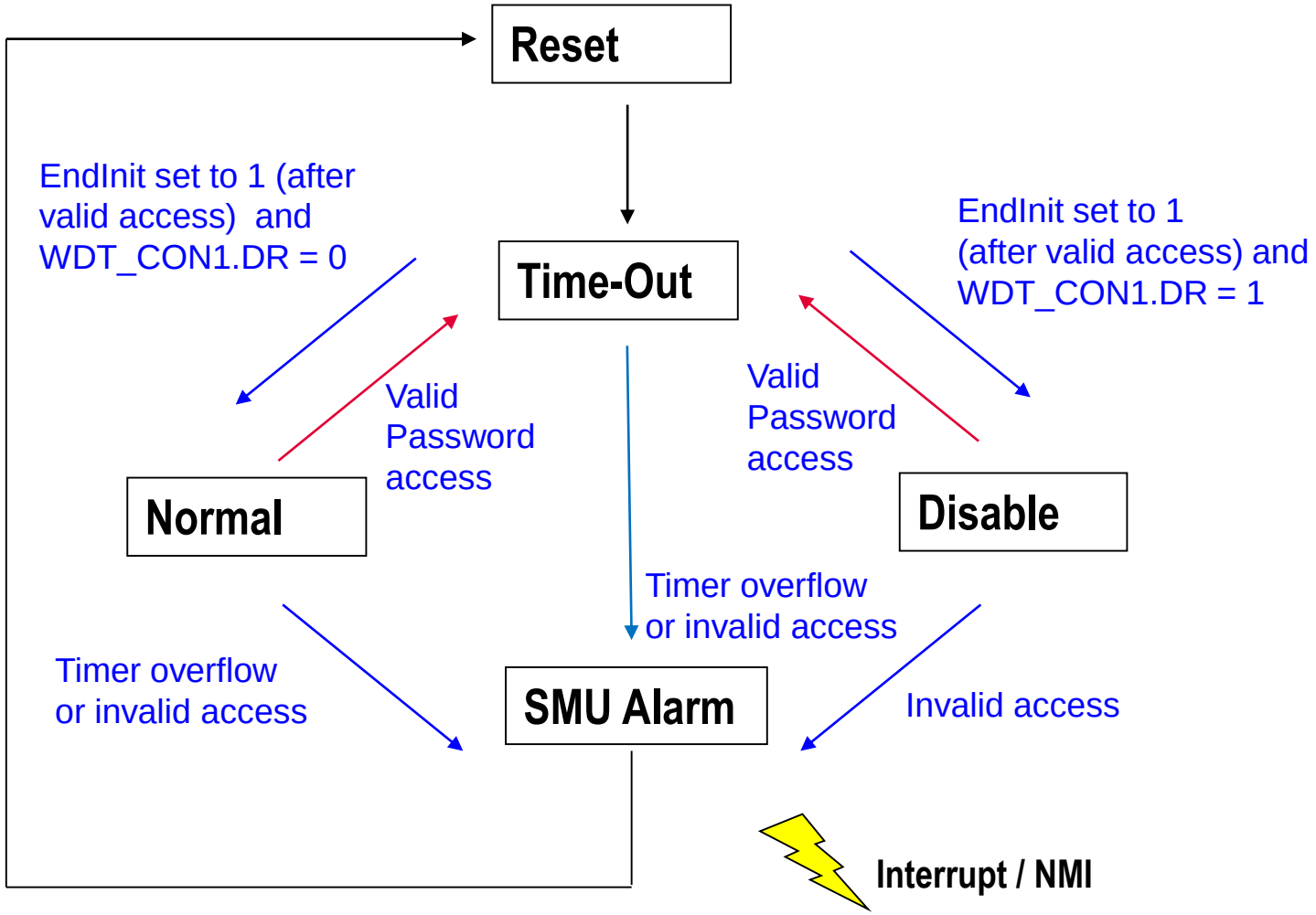

**Table 431 Modify Access Bit Pattern Requirements**

| Bit Position | Value |
|---|---|
| **0** | User-definable; desired value for bit WDTxCON0.ENDINIT. |
| **1** | Fixed; must be written with $1_B$. |
| **[15:2]** | User-definable; desired value of user-definable password field, WDTxCON0.PW. |
| **[31:16]** | User-definable; desired value of user-definable reload value, WDTxCON0.REL. |

# Access WDT_CON0 (cont.)

- When a valid password access is performed, WDT_CON0 is unlocked only for a short period of time (Time-Out mode).
  - Time Out mode can only be exited after a valid modify access that sets EndInit to 1.
  - After a valid modify access, WDT_CON0 is automatically locked again.
- If either an invalid password or modify access is performed, the register WDT_CON0 remains unchanged and the Watchdog timer enters the Pre-Warning Mode and eventually triggers a reset.
  - This protection scheme makes it very unlikely that WDT_CON0 is incorrectly modified.

# State Diagram of the Modes of the WDT



Part of your life. Part of tomorrow.

Infineon