

## 8 Clock control (CLK)

The clock controller is designed to be powerful, very robust, and at the same time easy to use. Its purpose is to allow you to obtain the best performance in your application while at the same time get the full benefit of all the microcontroller's power saving capabilities.

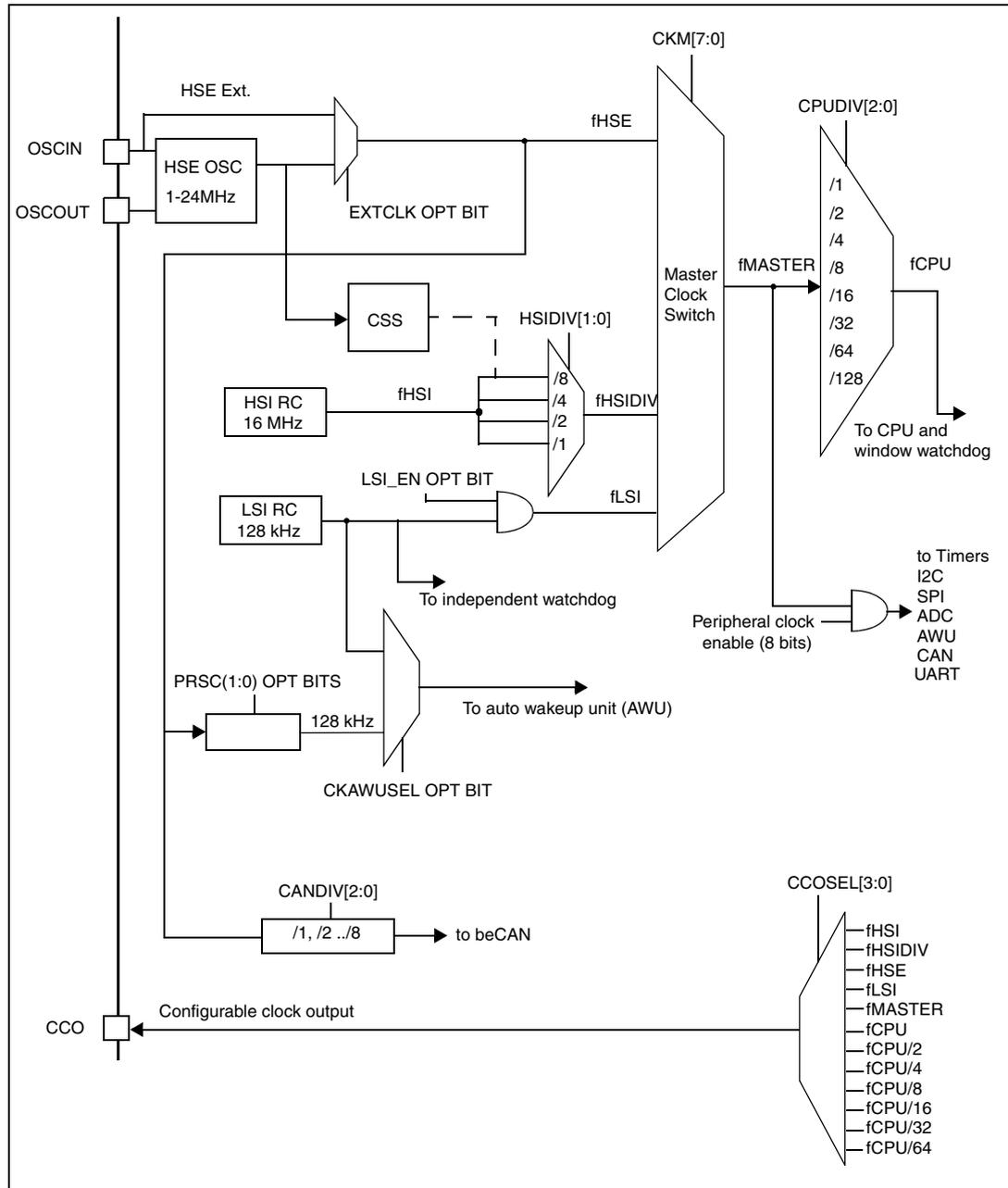
You can manage all the different clock sources independently and distribute them to the CPU and to the various peripherals. Prescalers are available for the master and CPU clocks.

A safe and glitch-free switch mechanism allows you to switch the master clock on the fly from one clock source to another one.

EMS-hardened clock configuration registers

To protect the application against spurious write access or system hang-up, possibly caused by electromagnetic disturbance, the most critical CLK registers are implemented as two bitfields that must contain complementary values. Mismatches are automatically detected by the CLK, triggering an EMS reset and allowing the application to cleanly recover normal operations. See [CLK register description](#) for more details.

Figure 13. Clock tree



1. Legend: HSE = High speed external clock signal; HSI = High speed internal clock signal; LSI = Low Speed internal clock signal.

## 8.1 Master clock sources

4 different clock sources can be used to drive the master clock:

- 1-24 MHz high speed external crystal oscillator (HSE)
- Up to 24 MHz high speed user-external clock (HSE user-ext)
- 16 MHz high speed internal RC oscillator (HSI)
- 128 kHz low speed internal RC (LSI)

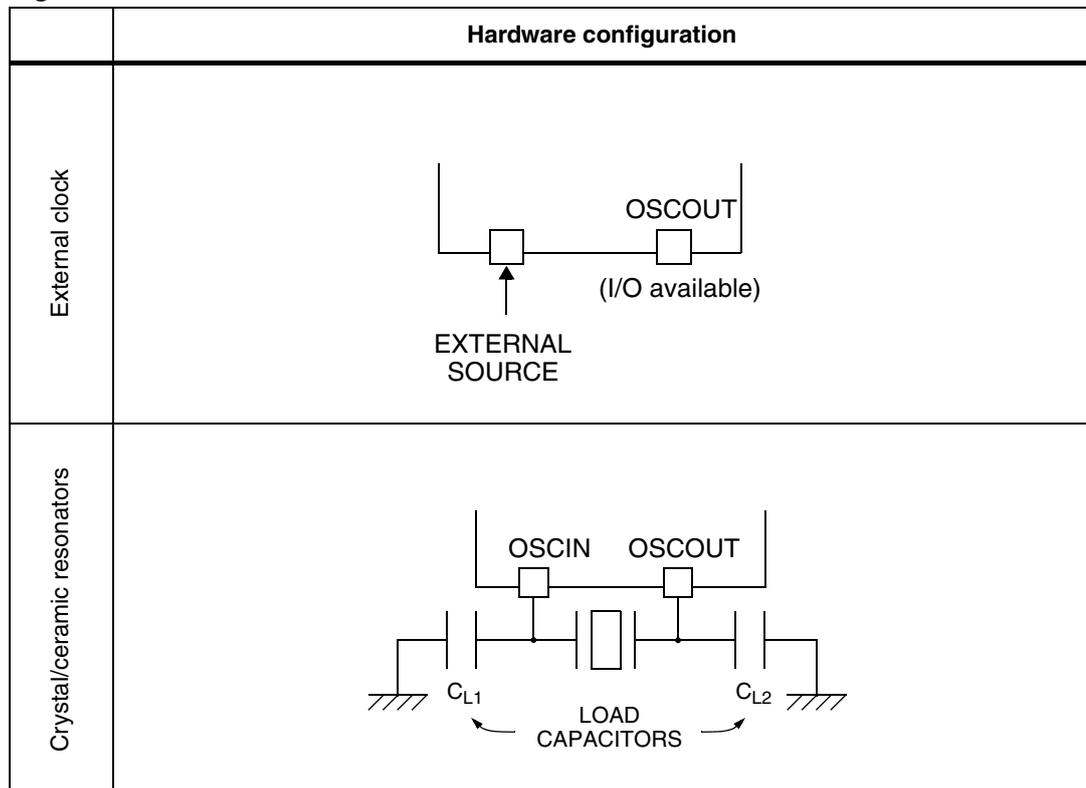
Each clock source can be switched on or off independently when it is not used, to optimize power consumption.

### 8.1.1 HSE

The high speed external clock signal (HSE) can be generated from two possible clock sources:

- HSE external crystal/ceramic resonator
- HSE user external clock

**Figure 14. HSE clock sources**



The resonator and the load capacitors have to be placed as close as possible to the oscillator pins in order to minimize output distortion and start-up stabilization time. The loading capacitance values must be adjusted according to the selected oscillator.

### External crystal/ceramic resonator (HSE crystal)

The 1 to 24 MHz external oscillator has the advantage of producing a very accurate rate on the main clock with 50% duty cycle.

The associated hardware configuration is shown in [Figure 14](#). Refer to the electrical characteristics section for more details.

At start up the clock signal produced by the oscillator is not stable, and by default a delay of 2048 osc cycles is inserted before the clock signal is released. You can program a shorter stabilization time in the HSECNT option byte, please refer to option bytes section in the datasheet.

The HSERDY flag in the [External clock register \(CLK\\_ECKR\)](#) indicates if the high-speed external oscillator is stable or not. At startup, the clock is not released until this bit is set by hardware.

The HSE Crystal can be switched on and off using the HSEEN bit in the [External clock register \(CLK\\_ECKR\)](#).

#### External source (HSE user-ext)

In this mode, an external clock source must be provided. It can have a frequency of up to 24 MHz. You select this mode by programming the EXTCLK option bit. Refer to the option bytes section of the datasheet. The external clock signal (square, sinus or triangle) with ~50% duty cycle has to drive the OSCIN pin while the OSCOUT pin is available as standard I/O (see [Figure 13](#)).

*Note:* For clock frequencies above 16 MHz, Flash /data EEPROM access must be configured for 1 wait state. This is enabled by the device option byte. Refer to the datasheet option byte section.

### 8.1.2 HSI

The HSI clock signal is generated from an internal 16 MHz RC oscillator together with a programmable divider (factor 1 to 8). This is programmed in the [Clock divider register \(CLK\\_CKDIVR\)](#).

*Note:* At startup the master clock source is automatically selected as HSI RC clock output divided by 8 ( $f_{HSI}/8$ ).

The HSI RC oscillator has the advantage of providing a 16 MHz master clock source with 50% duty cycle at low cost (no external components). It also has a faster startup time than the HSE crystal oscillator however, even with calibration the frequency is less accurate than an external crystal oscillator or ceramic resonator.

The HSIRDY flag in the [Internal clock register \(CLK\\_ICKR\)](#) indicates if the HSI RC is stable or not. At startup, the HSI RC output clock is not released until this bit is set by hardware.

The HSI RC can be switched on and off using the HSIEN bit in the [Internal clock register \(CLK\\_ICKR\)](#).

#### Backup source

The HSI/8 signal can also be used as a backup source (Auxiliary clock) if the HSE crystal oscillator fails. Refer to [Section 8.6: Clock security system \(CSS\)](#).

### Fast wakeup feature

If the FHWU bit in the *Internal clock register (CLK\_ICKR)* is set, this automatically selects the HSI clock as master clock after MCU wakeup from halt or active halt (see Low power chapter).

### Calibration

Each device is factory calibrated by ST.

After reset, the factory calibration value is automatically loaded in an internal calibration register.

If the application is subject to voltage or temperature variations this may affect the RC oscillator speed. You can trim the HSI frequency in the application using the *HSI clock calibration trimming register (CLK\_HSTRIMR)*. In this register there are 3 or 4 bits providing an additional trimming value that is added to the internal HSI calibration register value.

## 8.1.3 LSI

The 128 kHz LSI RC acts as a low power, low cost alternative master clock source as well as a low power clock source that can be kept running in halt mode for the independent watchdog (IWDG) and auto-wakeup unit (AWU).

The LSI RC can be switched on and off using the LSIEN bit in the *Internal clock register (CLK\_ICKR)*.

The LSIRDY flag in the *Internal clock register (CLK\_ICKR)* indicates if the low-speed internal oscillator is stable or not. At startup, the clock is not released until this bit is set by hardware.

### Calibration

Like the HSI RC, the LSI RC device is factory calibrated by ST. However, it is not possible to perform further trimming.

*Note: When using the independent watchdog with the LSI as clock source, in order to guarantee that the CPU will never run on the same clock in case of corruption, the LSI clock cannot be the master clock if LSI\_EN option bit is reset. Refer to the option bytes section in the datasheet.*

## 8.2 Master clock switching

The clock switching feature provides an easy to use, fast and secure way for the application to switch from one master clock source to another.

### 8.2.1 System startup

For fast system startup, after a reset the clock controller configures the master clock source as HSI RC clock output divided by 8 (HSI/8). This is to take advantage of the short stabilization time of the HSI oscillator. The /8 divider is to ensure safe start-up in case of poor  $V_{DD}$  conditions.

Once the master clock is released, the user program can switch the master clock to another clock source.

## 8.2.2 Master clock switching procedures

To switch clock sources, you can proceed in one of two ways:

- Automatic switching
- Manual switching

### Automatic switching

The automatic switching enables the user to launch a clock switch with a minimum number of instructions. The software can continue doing other operations without taking care of the switch event exact time.

Refer to the flowchart in [Figure 15](#).

1. Enable the switching mechanism by setting the SWEN bit in the [Switch control register \(CLK\\_SWCR\)](#).
2. Write the 8-bit value used to select the target clock source in the [Clock master switch register \(CLK\\_SWR\)](#). The SWBSY bit in the CLK\_SWCR register is set by hardware, and the target source oscillator starts. The old clock source continues to drive the CPU and peripherals.

As soon as the target clock source is ready (stabilized), the content of the CLK\_SWR register is copied to the [Clock master status register \(CLK\\_CMSR\)](#).

The SWBSY bit is cleared and the new clock source replaces the old one. The SWIF flag in the CLK\_SWCR is set and an interrupt is generated if the SWIEN bit is set.

### Manual switching

The manual switching is not as immediate as the automatic switching but it offers to the user a precise control of the switch event time.

Refer to the flowchart in [Figure 16](#).

1. Write the 8-bit value used to select the target clock source in the [Clock master switch register \(CLK\\_SWR\)](#). Then the SWBSY bit is set by hardware, and the target source oscillator starts. The old clock source continues to drive the CPU and peripherals.
2. The software has to wait until the target clock source is ready (stabilized). This is indicated by the SWIF flag in the CLK\_SWCR register and by an interrupt if the SWIEN bit is set.
3. The final software action is to set, at the chosen time, the SWEN bit in the CLK\_SWCR register to execute the switch.

In both manual and automatic switching modes, the old master clock source will not be powered off automatically in case it is required by other blocks (the LSI RC may be used to drive the independent watchdog for example). The clock source can be powered off using the bits in the [Internal clock register \(CLK\\_ICKR\)](#) and [External clock register \(CLK\\_ECKR\)](#).

If the clock switch does not work for any reason, software can reset the current switch operation by clearing the SWBSY flag. This will restore the CLK\_SWR register to its previous content (old master clock).

Figure 15. Clock switching flowchart (automatic mode example)

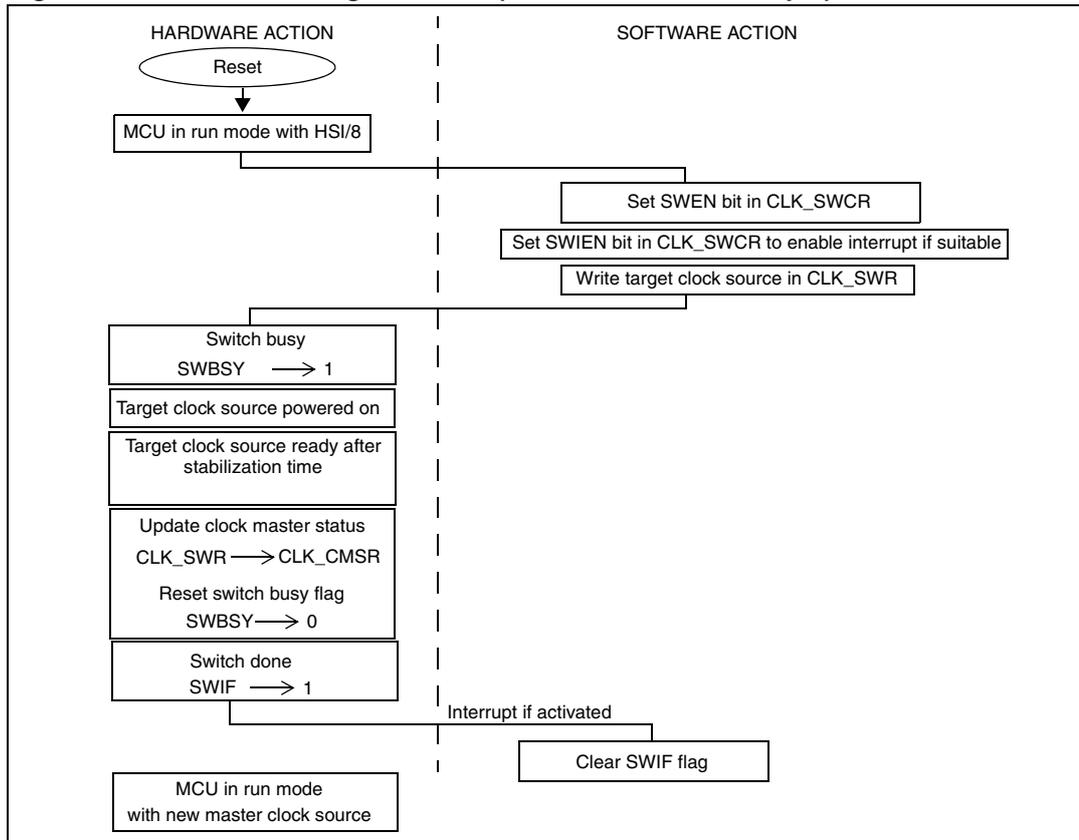
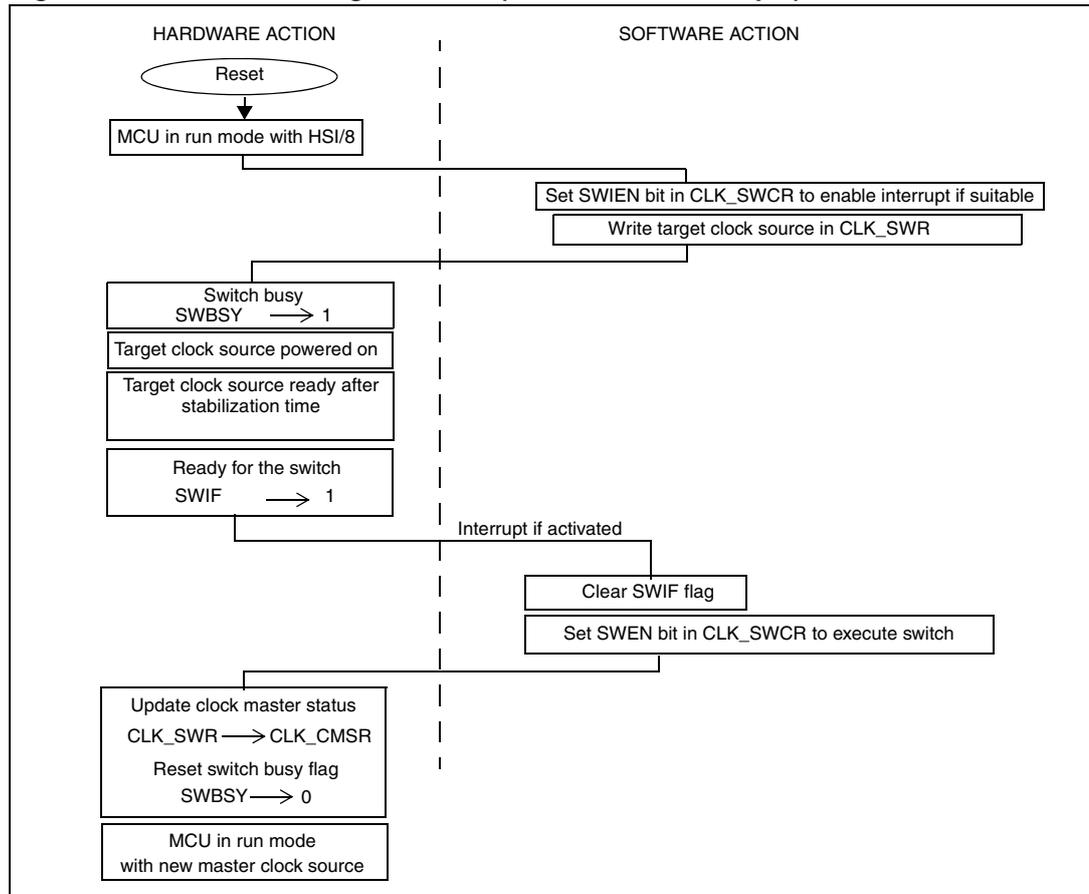


Figure 16. Clock switching flowchart (manual mode example)



### 8.3 Low speed clock selection

The Low speed clock source for the AWU or the independent watchdog can be LSI or HSE divided according to the CKAWUSEL option bit. Refer to option bytes section in the datasheet.

The division factor for HSE has to be programmed in the HSEPRSC[1:0] option bits Refer to in the option bytes section of the datasheet. The goal is to get 128 kHz at the output of the HSE prescaler.

### 8.4 CPU clock divider

The CPU clock ( $f_{CPU}$ ) is derived from the master clock ( $f_{MASTER}$ ), divided by a factor programmed in the CPUDIV[2:0] bits in the *Clock divider register (CLK\_CKDIVR)*. Seven division factors (1 to 128 in steps of power of 2) can be selected. Refer to *Figure 13*.

The  $f_{CPU}$  signal is the clock for both the CPU and the window watchdog.

## 8.5 Peripheral clock gating (PCG)

Gating the clock to unused peripherals helps reduce power consumption. Peripheral clock Gating (PCG) mode allows you to selectively enable or disable the  $f_{\text{MASTER}}$  clock connection to the following peripherals at any time in run mode:

- ADC
- I2C
- AWU (register clock, not counter clock)
- SPI
- TIM[4:1]
- UART
- CAN (register clock, not CAN clock)

After a device reset, all peripheral clocks are enabled. You can disable the clock to any peripheral by clearing the corresponding PCKEN bit in the *Peripheral clock gating register 1 (CLK\_PCKENR1)* and in the *Peripheral clock gating register 2 (CLK\_PCKENR2)*. But you have to disable properly the peripheral using the appropriate bit, before stopping the corresponding clock.

To enable a peripheral, you must first enable the corresponding PCKEN bit in the CLK\_PCKENR registers and then set the peripheral enable bit in the peripheral's control registers.

The AWU counter is driven by an internal or external clock (LSI or HSE) independent from  $f_{\text{MASTER}}$ , so that it continues to run even if the register clock to this peripheral is switched off.

## 8.6 Clock security system (CSS)

The Clock Security System (CSS) monitors HSE crystal clock source failures. When  $f_{\text{MASTER}}$  depends on HSE crystal, i.e. when HSE is selected, if the HSE clock fails due to a broken or disconnected resonator or any other reason, the clock controller activates a stall-safe recovery mechanism by automatically switching  $f_{\text{MASTER}}$  to the auxiliary clock source (HSI/8). Once selected the auxiliary clock source remains enabled until the MCU is reset.

You enable the clock security system by setting the CSSEN bit in the *Clock security system register (CLK\_CSSR)*. For safety reason, once CSS is enabled it cannot be disabled until the next reset.

The following conditions must be met so that the CSS can detect HSE quartz crystal failures:

- HSE crystal on: (HSEEN = 1 in the *External clock register (CLK\_ECKR)*)
- HSE oscillator in quartz crystal configuration (EXTCLK option bit is set)
- CSS function enabled: (CSSEN = 1 in the CLK\_CSSR register)

If HSE is the current clock master when a failure is detected, the CSS performs the following actions:

- The CSSD bit is set in the CLK\_CSSR register and an interrupt is generated if the CSSIEN bit is set.
- The *Clock master status register (CLK\_CMSR)*, *Clock master switch register (CLK\_SWR)* register and the HSIDIV[1:0] bits in the *Clock divider register (CLK\_CKDIVR)* are set to their reset values (CKM[7:0]= SWI[7:0]=E1h). HSI/8 becomes the master clock.
- The HSIEN bit in the *Internal clock register (CLK\_ICKR)* register is set (HSI on).
- The HSEEN bit in the *External clock register (CLK\_ECKR)* is cleared (HSE off)
- The AUX bit is set to indicate that the HSI/8 auxiliary clock source is forced.

You can clear the CSSD bit by software but the AUX bit is cleared only by reset.

To select a faster clock speed, you can modify the HSIDIV[1:0] bits in the CLK\_CKDIVR register after the CSSD bit in the CLK\_CSSR register is cleared.

If HSE is not the current clock master when a failure is detected, the master clock is not switched to the auxiliary clock and none of the above actions are performed except:

- The HSEEN bit is cleared in the CLK\_ECKR register, HSE is then switched OFF
- The CSSD bit is set in the CLK\_CSSR register and interrupt is generated if CSSDIE is also set, it can be cleared by software.

If HSE is not the current clock master and the master clock switch to HSE is ongoing, the SWBSY bit in the CLK\_SWCR register must be cleared by software before clearing the CSSD bit.

If HSE is selected by CCOSEL to be in output mode (see *Clock-out capability (CCO)*) when a failure is detected, the selection is automatically changed to force HSI (HSIDIV) instead of HSE.

## 8.7 Clock-out capability (CCO)

The configurable Clock Output (CCO) capability allows you to output a clock on the external CCO pin. You can select one of 6 clock signals as CCO clock:

- $f_{HSE}$
- $f_{HSI}$
- $f_{HSIDIV}$
- $f_{LSI}$
- $f_{MASTER}$
- $f_{CPU}$  (with current prescaling selection)

*Note:* 50% duty cycle is not guaranteed on all possible prescaled values

The selection is controlled by the CCOSEL[3:0] bits in the [Configurable clock output register \(CLK\\_CCOR\)](#).

The user has to select first the desired clock for the dedicated I/O pin (see Pin Description chapter). This I/O must be set at 1 in the corresponding Px\_CR1 register to be set as input with pull-up or push-pull output.

The sequence to really output the chosen clock starts with CCOEN=1 in [Configurable clock output register \(CLK\\_CCOR\)](#).

The CCOBSY is set to indicate that the configurable clock output system is operating. As long as the CCOBSY bit is set, the CCOSEL bits are write protected.

The CCO automatically activates the target oscillator if needed. The CCORDY bit is set when the chosen clock is ready.

To disable the clock output the user has to clear the CCOEN bit. Both CCOBSY and CCORDY remain at 1 till the shut down is completed. The time between the clear of CCOEN and the reset of the two flags can be relatively long, for instance in case the selected clock output is very slow compared to  $f_{CPU}$ .

## 8.8 CLK interrupts

The following interrupts can be generated by the clock controller:

- Master clock source switch event
- Clock Security System event

Both interrupts are individually maskable.

**Table 8. CLK interrupt requests**

Interrupt event	Event flag	Enable control bit	Exit from wait	Exit from halt
CSS event	CSSD	CSSDIE	Yes	No
Master clock switch event	SWIF	SWIEN	Yes	No

## 8.9 CLK register description

### 8.9.1 Internal clock register (CLK\_ICKR)

Address offset: 0x00

Reset value: 0x01

7	6	5	4	3	2	1	0
Reserved		REGAH	LSIRDY	LSIEN	FHW	HSIRDY	HSIEN
		rw	r	rw	rw	r	rw

Bits 7:6 Reserved, must be kept cleared.

Bit 5 **REGAH**: Regulator power off in active halt mode

This bit is set and cleared by software. When it is set, the main voltage regulator is powered off as soon as the MCU enters active halt mode, so the wakeup time is longer.

- 0: MVR regulator ON in active halt mode
- 1: MVR regulator OFF in active halt mode

Bit 4 **LSIRDY**: Low speed internal oscillator ready

This bit is set and cleared by hardware.

- 0: LSI clock not ready
- 1: LSI clock ready

Bit 3 **LSIEN**: Low speed internal RC oscillator enable

This bit is set and cleared by software. It is set by hardware whenever the LSI oscillator is required, for example:

- When switching to the LSI clock (see CLK\_SWR register)
- When LSI is selected as the active CCO source (see CLK\_CCOR register)
- When BEEP is enabled (BEEPEN bit set in the BEEP\_CSR register)
- When LSI measurement is enabled (MSR bit set in the AWU\_CSR register)

It cannot be cleared when LSI is selected as master clock source (CLK\_CMSR register), as active CCO source or as clock source for the AWU peripheral or independent Watchdog.

- 0: Low-speed internal RC off
- 1: Low-speed internal RC on

Bit 2 **FHWU**: Fast wakeup from halt/active halt modes

This bit is set and cleared by software.

- 0: Fast wakeup from halt/active halt modes disabled
- 1: Fast wakeup from halt/active halt modes enabled

Bit 1 **HSIRDY**: High speed internal oscillator ready

This bit is set and cleared by hardware.

- 0: HSI clock not ready
- 1: HSI clock ready

Bit 0 HSIEN: High speed internal RC oscillator enable

This bit is set and cleared by software. It is set by hardware whenever the HSI oscillator is required, for example:

- When activated as safe oscillator by the CSS
- When switching to HSI clock (see CLK\_SWR register)
- When HSI is selected as the active CCO source (see CLK\_CCOR register)

It cannot be cleared when HSI is selected as clock master (CLK\_CMSR register), as active CCO source or if the safe oscillator (AUX) is enabled.

0: High-speed internal RC off

1: High-speed internal RC on

### 8.9.2 External clock register (CLK\_ECKR)

Address offset: 0x01

Reset value: 0x00

	7	6	5	4	3	2	1	0
Reserved							HSERDY	HSEEN
							r	rw

Bits 7:2 Reserved, must be kept cleared.

Bit 1 **HSERDY**: High speed external crystal oscillator ready

This bit is set and cleared by hardware.

0: HSE clock not ready

1: HSE clock ready (HSE clock is stabilized and available)

Bit 0 **HSEEN**: High speed external crystal oscillator enable

This bit is set and cleared by software. It can be used to switch the external crystal oscillator on or off. It is set by hardware in the following cases:

- When switching to HSE clock (see CLK\_SWR register)
- When HSE is selected as the active CCO source (see CLK\_CCOR register)

It cannot be cleared when HSE is selected as clock master (indicated in CLK\_CMSR register) or as the active CCO source.

0: HSE clock off

1: HSE clock on

### 8.9.3 Clock master status register (CLK\_CMSR)

Address offset: 0x03

Reset value: 0xE1

7	6	5	4	3	2	1	0
CKM[7:0]							
r	r	r	r	r	r	r	r

Bits 7:0 **CKM[7:0]**: Clock master status bits

These bits are set and cleared by hardware. They indicate the currently selected master clock source. An invalid value occurring in this register will automatically generate an MCU reset.

0xE1: HSI selected as master clock source (reset value)

0xD2: LSI selected as master clock source (only if LSI\_EN option bit is set)

0xB4: HSE selected as master clock source

### 8.9.4 Clock master switch register (CLK\_SWR)

Address offset: 0x04

Reset value: 0xE1

7	6	5	4	3	2	1	0
SWI[7:0]							
rw	rw	rw	rw	rw	rw	rw	rw

Bits 7:0 **SWI[7:0]**: Clock master selection bits

These bits are written by software to select the master clock source. Its contents are write protected while a clock switch is ongoing (while the SWBSY bit is set). They are set to the reset value (HSI) if the AUX bit is set in the CLK\_CSSR register. If Fast Halt wakeup mode is selected (FHW bit = 1 in CLK\_IICKR register) then these bits are set by hardware to E1h (HSI selected) when resuming from Halt/Active halt mode.

0xE1: HSI selected as master clock source (reset value)

0xD2: LSI selected as master clock source (only if LSI\_EN option bit is set)

0xB4: HSE selected as master clock source

### 8.9.5 Switch control register (CLK\_SWCR)

Address offset: 0x05

Reset value: undefined

7	6	5	4	3	2	1	0
Reserved				SWIF	SWIEN	SWEN	SWBSY
				rc_w0	rw	rw	rw

Bits 7:4 Reserved, must be kept cleared.

Bit 3 **SWIF**: Clock switch interrupt flag

This bit is set by hardware and cleared by software writing 0. Its meaning depends on the status of the SWEN bit. Refer to [Figure 15](#) and [Figure 16](#).

- **In manual switching mode (SWEN = 0):**
  - 0: Target clock source not ready
  - 1: Target clock source ready
- **In automatic switching mode (SWEN = 1):**
  - 0: No clock switch event occurred
  - 1: Clock switch event occurred

Bit 2 **SWIEN**: Clock switch interrupt enable

This bit is set and cleared by software.

- 0: Clock switch interrupt disabled
- 1: Clock switch interrupt enabled

Bit 1 **SWEN**: Switch start/stop

This bit is set and cleared by software. Writing a 1 to this bit enables switching the master clock to the source defined in the CLK\_SWR register.

- 0: Disable clock switch execution
- 1: Enable clock switch execution

Bit 0 **SWBSY**: Switch busy

This bit is set and cleared by hardware. It can be cleared by software to reset the clock switch process.

- 0: No clock switch ongoing
- 1: Clock switch ongoing

### 8.9.6 Clock divider register (CLK\_CKDIVR)

Address offset: 0x06

Reset value: 0x18

7	6	5	4	3	2	1	0
Reserved			HSIDIV[1:0]		CPUDIV[2:0]		
			rw	rw	rw	rw	rw

Bits 7:5 Reserved, must be kept cleared.

Bits 4:3 **HSIDIV[1:0]**: High speed internal clock prescaler

These bits are written by software to define the HSI prescaling factor.

- 00:  $f_{HSI} = f_{HSI\ RC\ output}$
- 01:  $f_{HSI} = f_{HSI\ RC\ output}/2$
- 10:  $f_{HSI} = f_{HSI\ RC\ output}/4$
- 11:  $f_{HSI} = f_{HSI\ RC\ output}/8$

Bits 2:0 **CPUDIV[2:0]**: CPU clock prescaler

These bits are written by software to define the CPU clock prescaling factor.

- 000:  $f_{CPU} = f_{MASTER}$
- 001:  $f_{CPU} = f_{MASTER}/2$
- 010:  $f_{CPU} = f_{MASTER}/4$
- 011:  $f_{CPU} = f_{MASTER}/8$
- 100:  $f_{CPU} = f_{MASTER}/16$
- 101:  $f_{CPU} = f_{MASTER}/32$
- 110:  $f_{CPU} = f_{MASTER}/64$
- 111:  $f_{CPU} = f_{MASTER}/128$

### 8.9.7 Peripheral clock gating register 1 (CLK\_PCKENR1)

Address offset: 0x07

Reset value: 0xFF

7	6	5	4	3	2	1	0
PCKEN1[7:0]							
rw	rw	rw	rw	rw	rw	rw	rw

Bits 7:0 **PCKEN1[7:0]**: Peripheral clock enable

These bits are written by software to enable or disable the  $f_{MASTER}$  clock to the corresponding peripheral (see [Table 9](#)).

0:  $f_{MASTER}$  to peripheral disabled

1:  $f_{MASTER}$  to peripheral enabled

**Table 9. Peripheral clock gating bits**

Control bit	Peripheral
PCKEN17	TIM1
PCKEN16	TIM3
PCKEN15	TIM2/TIM5 (product dependent)
PCKEN14	TIM4/ TIM6 (product dependent)
PCKEN13	UART1/2/3 (product dependent, see datasheet for bit assignment table)
PCKEN12	
PCKEN11	SPI
PCKEN10	I <sup>2</sup> C

### 8.9.8 Peripheral clock gating register 2 (CLK\_PCKENR2)

Address offset: 0x0A

Reset value: 0xFF

7	6	5	4	3	2	1	0
PCKEN2[7:0]							
rw	rw	rw	rw	rw	rw	rw	rw

Bits 7:0 **PCKEN2[7:0]**: Peripheral clock enable

These bits are written by software to enable or disable the  $f_{\text{MASTER}}$  clock to the corresponding peripheral. See [Table 9](#)

0:  $f_{\text{MASTER}}$  to peripheral disabled

1:  $f_{\text{MASTER}}$  to peripheral enabled

**Table 10. Peripheral clock gating bits**

Control bit	Peripheral
PCKEN27	CAN (product dependent, see datasheet)
PCKEN26	Reserved
PCKEN25	Reserved
PCKEN24	Reserved
PCKEN23	ADC
PCKEN22	AWU
PCKEN21	Reserved
PCKEN20	Reserved

### 8.9.9 Clock security system register (CLK\_CSSR)

Address offset: 0x08

Reset value: 0x00

7	6	5	4	3	2	1	0
Reserved				CSSD	CSSDIE	AUX	CSSEN
				rc_w0	rw	r	rwo

Bits 7:4 Reserved, must be kept cleared.

Bit 3 **CSSD**: Clock security system detection

This bit is set by hardware and cleared by software writing 0.

0: CSS is off or no HSE crystal clock disturbance detected.

1: HSE crystal clock disturbance detected.

Bit 2 **CSSDIE**: Clock security system detection interrupt enable

This bit is set and cleared by software.

0: Clock security system interrupt disabled

1: Clock security system interrupt enabled

Bit 1 **AUX**: Auxiliary oscillator connected to master clock

This bit is set and cleared by hardware.

0: Auxiliary oscillator is off.

1: Auxiliary oscillator (HSI/8) is on and selected as current clock master source.

Bit 0 **CSSEN**: Clock security system enable

This bit can be read many times and be written once-only by software.

0: Clock security system off

1: Clock security system on

### 8.9.10 Configurable clock output register (CLK\_CCOR)

Address offset: 0x09

Reset value: 0x00

7	6	5	4	3	2	1	0
Reserved	CCOBSY	CCORDY	CCOSEL[3:0]				CCOEN
	r	r	rw	rw	rw	rw	rw

Bit 7 **Reserved**, must be kept cleared.

Bit 6 **CCOBSY**: Configurable clock output busy

This bit is set and cleared by hardware. It indicates that the selected CCO clock source is being switched-on and stabilized. While CCOBSY is set, the CCOSEL bits are write-protected. CCOBSY remains set until the CCO clock is enabled.

0: CCO clock not busy

1: CCO clock busy

Bit 5 **CCORDY**: Configurable clock output ready

This bit is set and cleared by hardware. It indicates that the CCO clock is being output.

0: CCO clock not available

1: CCO clock available

Bits 4:1 **CCOSEL[3:0]**: Configurable clock output selection.

These bits are written by software to select the source of the output clock available on the CLK\_CCO pin. They are write-protected when CCOBSY is set.

0000:  $f_{HSIDIV}$

0001:  $f_{LSI}$

0010:  $f_{HSE}$

0011: Reserved

0100:  $f_{CPU}$

0101:  $f_{CPU}/2$

0110:  $f_{CPU}/4$

0111:  $f_{CPU}/8$

1000:  $f_{CPU}/16$

1001:  $f_{CPU}/32$

1010:  $f_{CPU}/64$

1011:  $f_{HSI}$

1100:  $f_{MASTER}$

1101:  $f_{CPU}$

1110:  $f_{CPU}$

1111:  $f_{CPU}$

Bit 0 **CCOEN**: Configurable clock output enable

This bit is set and cleared by software.

0: CCO clock output disabled

1: CCO clock output enabled

### 8.9.11 CAN external clock control register (CLK\_CANCCR)

Address offset: 0x0B

Reset value: 0x00



Bits 7:0 **CANDIV[2:0]**: External CAN clock divider

These bits are written by software to define the divider for the external CAN clock. See [Section 23.9: Clock system on page 379](#) for more details.

000: External CAN clock =  $f_{HSE}/1$  (reset value)

001: External CAN clock =  $f_{HSE}/2$

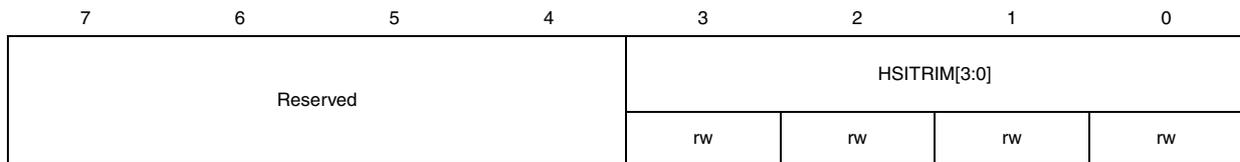
...

111: External CAN clock =  $f_{HSE}/8$

### 8.9.12 HSI clock calibration trimming register (CLK\_HSITRIMR)

Address offset: 0x0C

Reset value: undefined



Bits 7:3 Reserved, must be kept cleared.

Bits 2:0 **HSITRIM[3:0]** HSI trimming value

These bits are written by software to fine tune the HSI calibration.

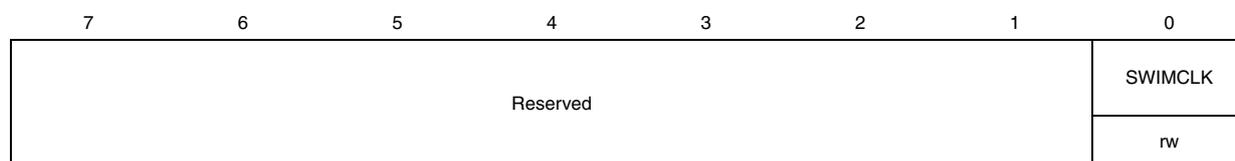
*Note: In high density devices only bits 2:0 are available.*

*In medium and low density devices bits 3:0 or 2:0 are available, depending on the option byte configuration (refer to datasheet).*

### 8.9.13 SWIM clock control register (CLK\_SWIMCCR)

Address offset: 0x0D

Reset value: undefined



Bits 7:1 Reserved, must be kept cleared.

- Bit 0 **SWIMCLK** SWIM clock divider  
 This bit is set and cleared by software.  
 0: SWIM clock divided by 2  
 1: SWIM clock not divided by 2

## 8.10 CLK register map

Table 11. CLK register map and reset values

Address offset	Register name	7	6	5	4	3	2	1	0
0x00	CLK_ICKR Reset value	- 0	- 0	REGAH 0	LSIRDY 0	LSIEN 0	FHWU 0	HSIRDY 0	HSIEN 1
0x01	CLK_ECKR Reset value	- 0	- 0	- 0	- 0	- 0	- 0	HSERDY 0	HSEEN 0
0x02	Reserved area (1 byte)								
0x03	CLK_CMSR Reset value	CKM7 1	CKM6 1	CKM5 1	CKM4 0	CKM3 0	CKM2 0	CKM1 0	CKM0 1
0x04	CLK_SWR Reset value	SWI7 1	SWI6 1	SWI5 1	SWI4 0	SWI3 0	SWI2 0	SWI1 0	SWI0 1
0x05	CLK_SWCR Reset value	- x	- x	- x	- x	SWIF 0	SWIEN 0	SWEN 0	SWBSY 0
0x06	CLK_CKDIVR Reset value	- 0	- 0	- 0	HSIDIV1 1	HSIDIV0 1	CPUDIV2 0	CPUDIV12 0	CPUDIV0 0
0x07	CLK_PCKENR1 Reset value	PCKEN17 1	PCKEN16 1	PCKEN15 1	PCKEN14 1	PCKEN13 1	PCKEN12 1	PCKEN11 1	PCKEN10 1
0x08	CLK_CSSR Reset value	- 0	- 0	- 0	- 0	CSSD 0	CSSDIE 0	AUX 0	CSSEN 0
0x09	CLK_CCOR Reset value	- 0	CCOBSY 0	CCORDY 0	CCOSEL3 0	CCOSEL2 0	CCOSEL1 0	CCOSEL0 0	CCOEN 0
0x0A	CLK_PCKENR2 Reset value	PCKEN27 1	PCKEN26 1	PCKEN25 1	PCKEN24 1	PCKEN23 1	PCKEN22 1	PCKEN21 1	PCKEN20 1
0x0B	CLK_CANCCR Reset value	- x	- x	- x	- x	- x	CANDIV2 0	CANDIV1 0	CANDIV0 0
0x0C	CLK_HSITRIMR Reset value	- x	- x	- x	- x	- x	HSITRIM2 0	HSITRIM1 0	HSITRIM0 0
0x0D	CLK_SWIMCCR Reset value	- x	- x	- x	- x	- x	- 0	- 0	SWIMCLK 0