27.2 Overview

Figure 1 provides an overview on the STM module. It shows the options for reading parts of the STM content.

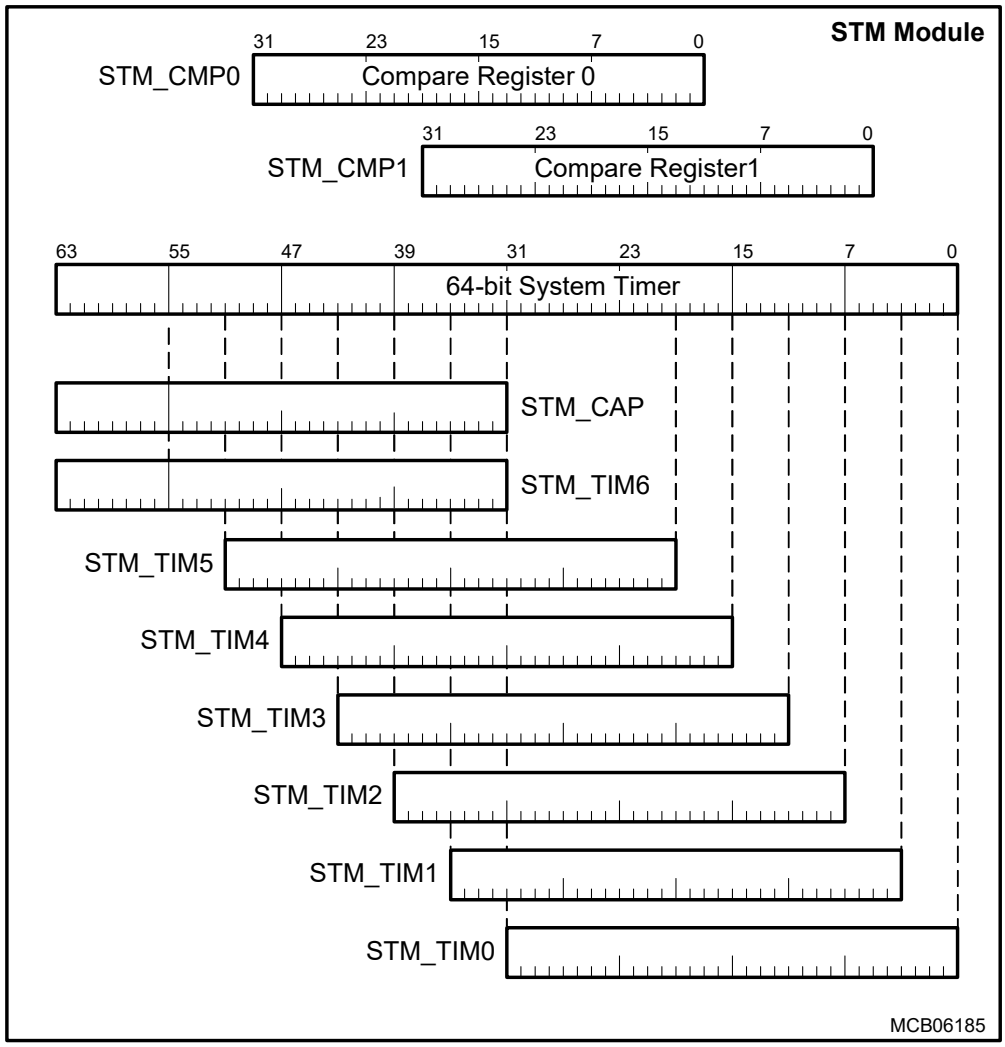

Figure 1 General Block Diagram of the STM Module

27.3 Functional Description

The STM is an upward counter, running at frequency f_{STM}. In case of an Application Reset, the STM is reset if bit SCU_ARSTDIS.STMxDIS is cleared. After reset, the STM is enabled and immediately starts counting up. It is not possible to affect the content of the timer during normal operation. The timer registers can only be read but not written to.

The STM can be optionally disabled for power-saving purposes, or suspended for debugging purposes. In suspend mode, the STM clock is stopped but all registers are still readable.

Due to the 64-bit width of the STM, it is not possible to read its entire content with one instruction. It needs to be read with two load instructions. Since the timer would continue to count between the two load operations, there is a chance that the two values read are not consistent (due to possible overflow from the low part of the timer to the high part between the two read operations). To enable a synchronous and consistent reading of the STM content, a capture register (CAP) is implemented. It latches the content of the high part of the STM each time

when one of the registers TIM0 to TIM5 is read. Thus, CAP holds the upper value of the timer at exactly the same time when the lower part is read. The second read operation would then read the content of the CAP to get the complete timer value.

The STM can also be read in sections from seven registers, TIM0 through TIM6, that select increasingly higher-order 32-bit ranges of the STM. These can be viewed as individual 32-bit timers, each with a different resolution and timing range.

The content of the 64-bit System Timer can be compared against the content of two compare values stored in the CMP0 and CMP1 registers. Service requests can be generated on a compare match of the STM with the CMP0 or CMP1 registers.

27.3.1 Compare Register Operation

The content of the 64-bit STM can be compared against the content of two compare values stored in the CMP0 and CMP1 registers. Service requests can be generated on a compare match of the STM with the CMP0 or CMP1 registers.

Two parameters are programmable for the compare operation:

1. The width of the relevant bits in registers CMP0/CMP1 (compare width MSIZEx) that is taken for the compare operation can be programmed from 0 to 31.
2. The first bit location in the 64-bit STM that is taken for the compare operation can be programmed from 0 to 31.

These programming capabilities make compare functionality very flexible. It even makes it possible to detect bit transitions of a single bit n (n = 0 to 31) within the 64-bit STM by setting MSIZE = 0 and MSTART = n.

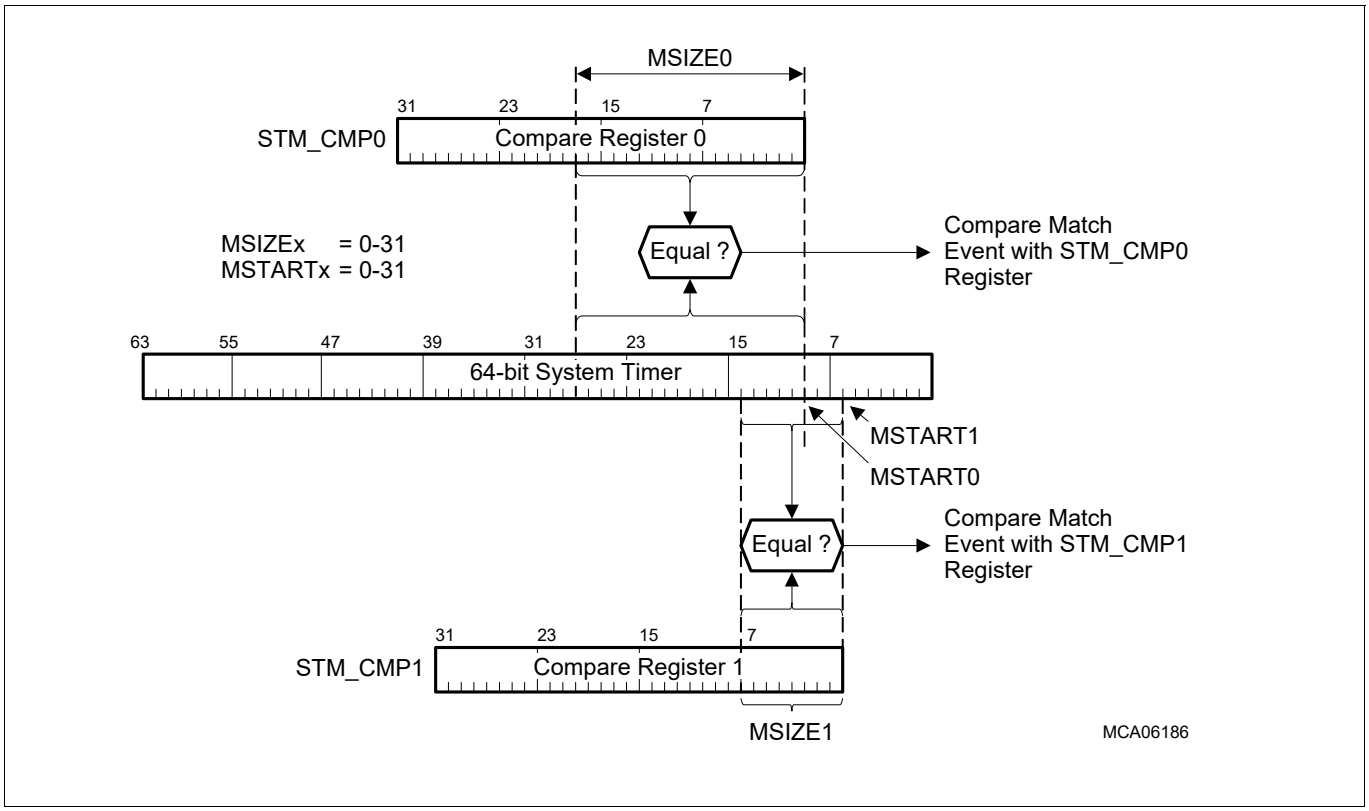

Figure 2 Compare Mode Operation

Figure 2 shows an example of the compare operation. In this example the following parameters are programmed:

- MSIZE0 = $10001_B = 17_D$; MSTART0 = $01010_B = 10_D$
- MSIZE1 = $00111_B = 7_D$; MSTART1 = $00111_B = 7_D$

A compare operation with MSIZE not equal 11111_B always implies that the compared value as stored in the CMP register is right-extended with zeros. This means that in the example of Figure 2, the compare register content CMP0[17:0] plus ten zero bits right-extended is compared with STM[27:0] with STM[9:0] = 000_H. In case of register CMP1, STM[14:0] with STM[6:0] = 00_H are compared with CMP1[9:0] plus seven zero bits right-extended.

27.3.2 Compare Match Interrupt Control

The compare match interrupt control logic is shown in **Figure 3**. Each CMPx register has its compare match interrupt request flag (ICR.CMPxIR) that is set by hardware on a compare match event. The interrupt request flags can be set (ISCR.CMPxIRS) or cleared (ISCR.CMPxIRR) by software. Note that setting ICR.CMPxIR by writing a 1 into ISSR.CMPxIRS does not generate an interrupt at STMIRx. The compare match interrupts from CMP0 and CMP1 can be further directed by ICR.CMPxOS to either output signal STMIR0 or STMIR1.

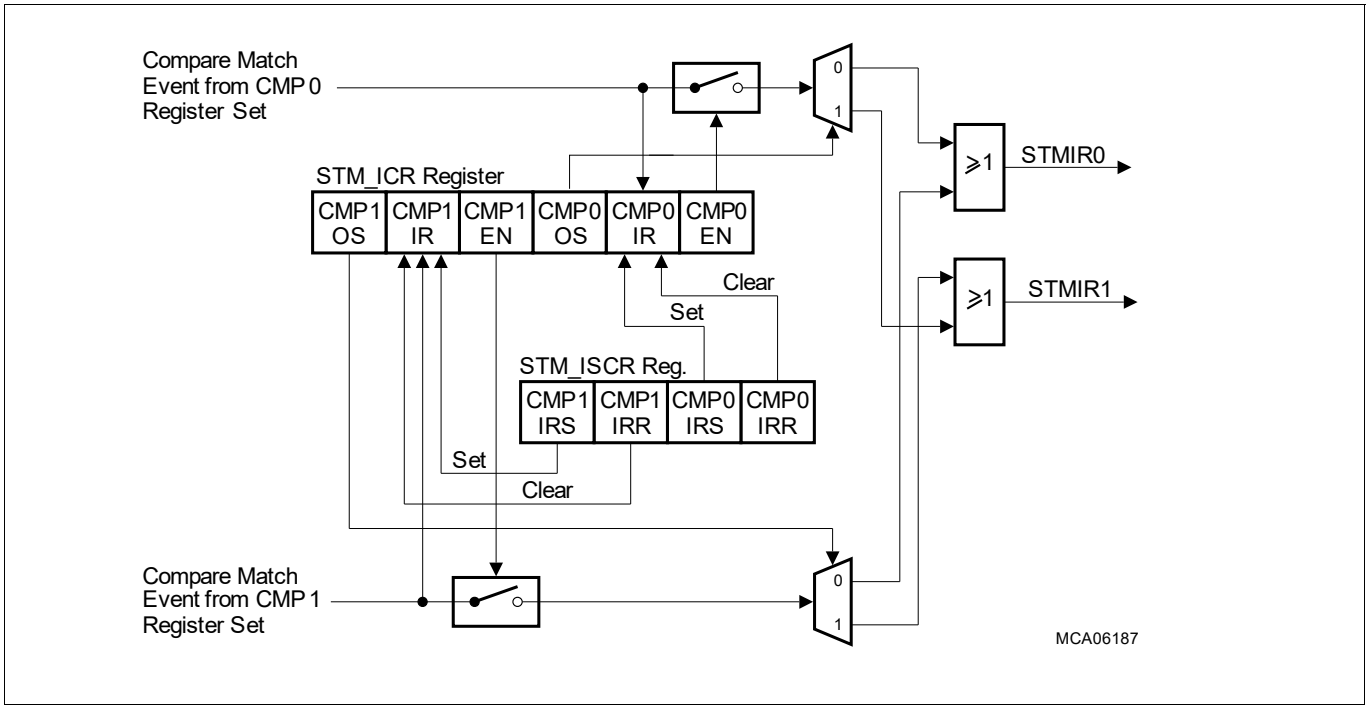

Figure 3 STM Interrupt Control

The compare match interrupt flags ICR.CMPxIR are immediately set after an STM reset operation, caused by a compare match event with the reset values of the STM and the compare registers CMPx. This does not directly generate compare match interrupts because the compare match interrupts are automatically disabled after an STM reset operation (ICR.CMPxEN = 0). Therefore, before enabling a compare match interrupt after an STM Application Reset, the software should configure the STM and modify the reset values of the compare registers. Otherwise, undesired compare match interrupt events are triggered. The CMPxIR flags which are set after an STM reset can be cleared by writing register ISCR with CMPxIRR set.

27.3.3 Using Multiple STMs

For systems that include multiple CPUs there are also multiple STMs available. Each STM is aimed to serve as time base for one individual CPU operating system main scheduler clock trigger. This is done by the usage of one compare register and the associated interrupt generating the trigger.

All STM modules are connected and controlled by f_{STM} and can therefore operate on the same frequency if desired.

27.3.4 STM as Reset Trigger

A compare match triggered by an CMP0 event can generate a reset in the system. The reset has to be enabled for each STM module individually in register SCU_RSTCON.