Offline BIST clock configuration in the SPC58xEx and SPC58xGx devices

Introduction

This technical note describes how to set up the clocks for the execution of the offline BIST.

During the offline BIST execution, the user cannot configure all parameters of the *Clock generation* tree reported in the "Clocking" chapter of the RM.

Clock dividers (like "CGM_SC_DCx" or "CGM_ACx_DCy"), for example, are fixed at their default value, and the user cannot change them.

However, some STCU2 registers allow you to program at what frequency the offline BIST can run and select internal IRC or PLL0 as the clock source.

This technical note focuses on the SPC58xEx and SPC58xGx devices but most of the concepts are also valid for the entire SPC58 family, taking care to evaluate the main differences between the different clock tree architectures.

1 Overview

BISTs are periodic tests that detect latent faults in the device.

The SPC58xEx and SPC58xGx devices include two different types of BIST:

- Memory BIST (MBIST): for volatile memories
- Logic BIST (LBIST): for digital logic

A BIST operation can run during the IDLE[DEST] phase of the reset configures by a set of DCFs (offline mode) or during runtime configured by a set of registers (online mode).

The STCU2 hardware module controls the BIST execution for online and offline modes.

In terms of clock configuration, the user can:

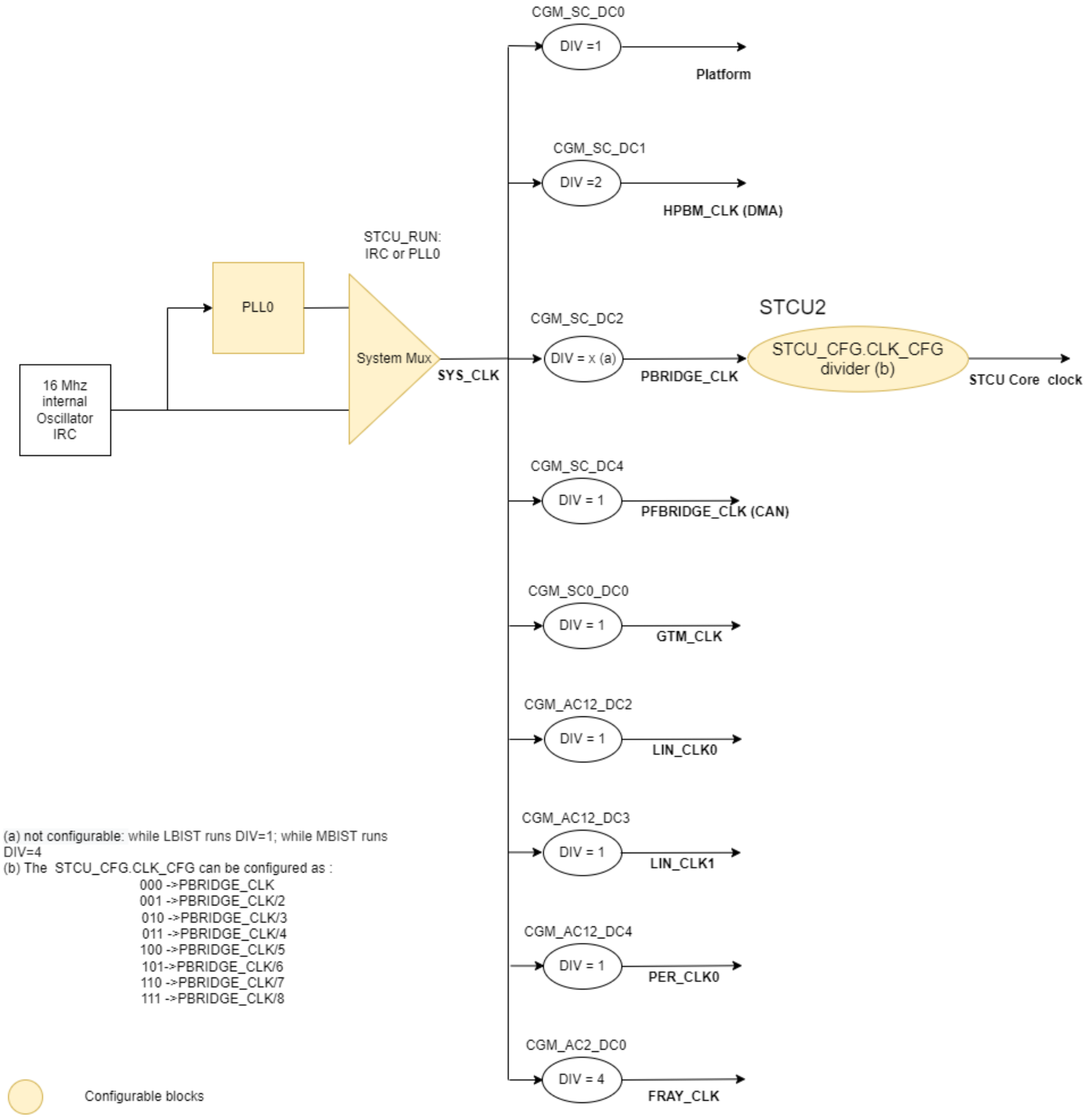
- configure the clock by the STCU2 module in offline mode
- configure the clock by the MC_CGM module in online mode.

All the configurations described in the following sections apply to the execution of the BIST in offline mode.

1.1 Clock tree during offline self-test

During the offline BIST, the user can consider a simplified clock diagram:

Figure 1. **Clock tree for offline BIST**



(a) not configurable: while LBIST runs DIV=1; while MBIST runs DIV=4
(b) The STCU_CFG.CLK_CFG can be configured as :

000	->PBRIDGE_CLK
001	->PBRIDGE_CLK/2
010	->PBRIDGE_CLK/3
011	->PBRIDGE_CLK/4
100	->PBRIDGE_CLK/5
101	->PBRIDGE_CLK/6
110	->PBRIDGE_CLK/7
111	->PBRIDGE_CLK/8

This figure reports the clock frequency of peripherals relevant to the execution of the offline BIST. Two clock sources are available during the execution of the offline BIST:

1. IRC
2. PLL0.

Note: *The XOSC clock source is not active during the execution of the offline BIST.*

1.1.1 STCU2 registers for offline clock configuration

The user writes three STCU2 registers for configuring the clock during offline BIST:

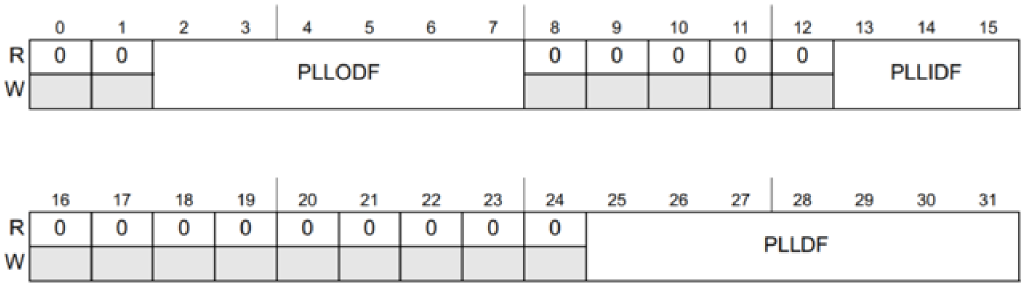
- **STCU_PLL_CFG**, to configure the PLL0
- **STCU_RUN**, to start offline BIST and select the source frequency
- **STCU_CFG**, to configure the STCU2 Core clock.

The STCU_PLL_CFG register contains three fields that the user can configure:

- PLLODF

- PLLIDF
- PLLDF.

Figure 2. Table STCU_PLL_CFG register



The formula below calculates the frequency of the PLL output clock:

$$fPLLout = \frac{fVCO}{PLLODF.2}$$

Where:

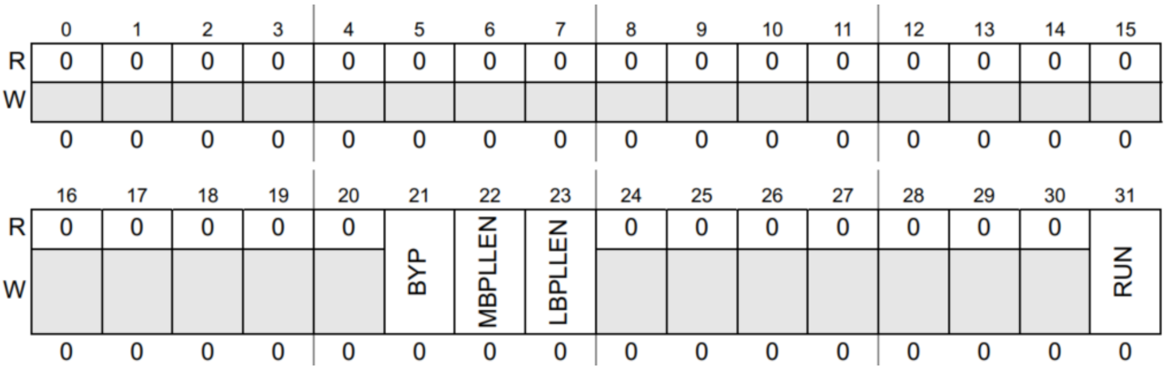
$$fVCO = fPLLin \cdot \left(PLLDF \cdot 2\right) = \left(\frac{fRC}{PLLIDF}\right) \cdot \left(PLLDF \cdot 2\right)$$

fRC is the frequency of IRC (that is, 16 MHz).

The user shall pay attention to the frequency range of the fVCO, which shall remain between 600 MHz and 1400 MHz.

The clock source is selected from the **STCU_RUN** register.

Figure 3. STCU_RUN register



The following table shows the allowed combination for the clock selection.

Table 1. Clock source selection

Possible configuration	LBPLLEN	MBPLLEN	Behavior
LBIST only	0	0	IRC osc drives the LBIST
	1	0 [(1)]	PLL drives the LBIST
MBIST only	0 [(1)]	0	IRC osc drives the MBIST
	0 [(1)]	1	PLL drives the MBIST
LBIST runs first, MBIST runs second	0	0	IRC osc drives both LBIST and MBIST
	0	1	IRC osc drives the LBIST

Possible configuration	LBPLLEN	MBPLLEN	Behavior
LBIST runs first, MBIST runs second			PLL0 drives MBIST
	1	0	Not allowed
	1	1	PLL0 drives both LBIST and MBIST
MBIST runs first, LBIST runs second	0	0	IRC osc drives both LBIST and MBIST
	0	1	Not allowed [2]
	1	0	IRC osc drives the MBIST PLL0 drives LBIST
	1	1	PLL0 drives both LBIST and MBIST

1. '1' is not allowed

2. It is not allowed to switch from PLL to RC

1.1.2 CGM dividers during offline BIST

The value of the CGM dividers is fixed to a default value, and the user cannot change it. Figure 1. Clock tree for offline BIST shows the value of these CGM dividers.

Note: The hardware changes the value of the CGM_SC_DC2 divider according to the BIST execution:

1. CGM_SC_DC2.DIV equal to '0' (DIV = 1) during LBIST execution
2. CGM_SC_DC2.DIV equal to '3' (DIV = 4) during MBIST execution.

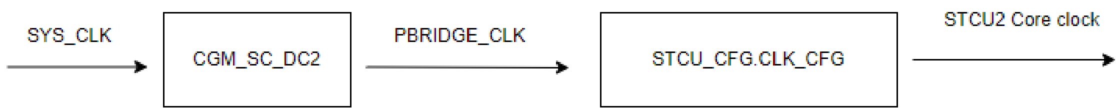
2 STCU2 clock configuration

Besides the frequency limitation of each memory and logic part involved during the offline BIST,

the user shall properly set the STCU2 peripheral that controls the BIST execution for both offline and online modes.

The clock for the STCU2 logic comes from the system clock that goes through two dividers configured by CGM_SC_DC2 and STCU_CFG.CLK_CFG (parameter of the **STCU_CFG** register).

Figure 4. **STCU clock**



The maximum frequency for the STCU2 Core clock is 50 MHz and the STCU_CFG.CLK_CFG field can have the following values (see the following table).

Table 2. **STCU_CFG.CLK_CFG values**

000	STCU2 core clock = PBRIDGE_CLK
001	STCU2 core clock = PBRIDGE_CLK/2
010	STCU2 core clock = PBRIDGE_CLK/3
011	STCU2 core clock = PBRIDGE_CLK/4
100	STCU2 core clock = PBRIDGE_CLK/5
101	STCU2 core clock = PBRIDGE_CLK/6
110	STCU2 core clock = PBRIDGE_CLK/7
111	STCU2 core clock = PBRIDGE_CLK/8

STCU freq is so calculated:

$$STCU\ freq = \frac{1}{STCU2Core\ clock} = \frac{SYS\_CLK}{(CGM\_SC\_DC2.DIV + 1)*(STCU\_CFG.CLK\_CFG\ + 1)} = \frac{PBRIDGE\_CLK}{(STCU\_CFG.CLK\_CFG\ + 1)} \tag{1}$$

The following examples show how to set these dividers to prevent to exceed the max frequency of the STCU (50 MHz).

Example N.1 - Offline LBIST

1. The maximum frequency of the LBIST is 100 MHz. This is the SYS_CLK frequency (see Figure 1. Clock tree for offline BIST).
2. CGM_SC_DC2.DIV = 0 when the offline LBIST runs

As a consequence, the user must configure the STCU_CFG.CLK_CFG = 1.

(STCU_CFG.CLK_CFG shall not be 0).

$$STCU\ freq = \frac{100}{(0 + 1)*(1 + 1)} = \frac{100}{2} = \ 50\ MHz \tag{2}$$

Example N.2 - Offline MBIST

1. The maximum frequency of the MBIST is 180 MHz.This is the SYS_CLK frequency (see Figure 1. Clock tree for offline BIST).
2. CGM_SC_DC2.DIV = 3 when the offline MBIST runs

As a consequence, the user could configure the STCU_CFG.CLK_CFG = 0.

$$STCU_{freq} = \frac{180}{(3+1)*(0+1)} = \frac{45}{(0+1)} = 45\,MHz \tag{3}$$

Example N.3 - Offline LBIST and MBIST

1. The maximum frequency of SYS_CLK is 100 MHz (max frequency of the LBIST)
2. CGM_SC_DC2.DIV= 0 during LBIST and CGM_SC_DC2.DIV = 3 during MBIST

As a consequence,

- During LBIST, the STCU2 frequency = 50 MHz
- During MBIST, the STCU2 frequency = 12.5 MHz.

  As a consequence, the user shall configure the STCU_CFG.CLK_CFG = 1.

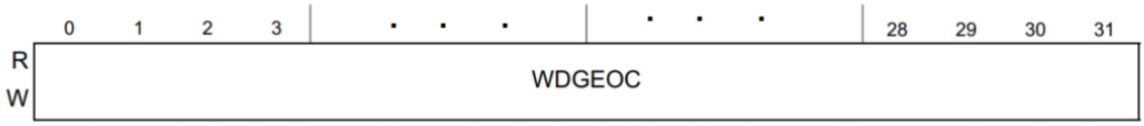
  (STCU_CFG.CLK_CFG = 0 is forbidden to avoid STCU overclocking.)

2.1 WDG configuration

The STCU2 WDG monitors the self-test execution. It is clocked by the STCU2 Core clock.

If the STCU2 WDG timer expires before the STCU2 completes the BIST execution, the STCU2 WDG triggers a fault. For this reason, the user shall configure the WDG counter (WDG_EOC field in the STCU_WDG resister) with a proper value. The following table shows the relationship between the WDG counter and the STCU2 Core clock (additional details in the reference manual see Section 5 Reference document).

Figure 5. **STCU_WDG register**



- 0x0000_0000: 1 * 16 * STCU2 Core clock
- 0x0000_0001: 2 * 16 * STCU2 Core clock

Generalizing, we have:

$$Timeout = (WDG\_EOC + 1)*16*STCU2\ Core\ clock \tag{4}$$

And then:

$$\mathrm{STCUfreq} = \frac{1}{STCU2\ \mathrm{Coreclock}}$$

$$WDG\_EOC = \left(Timeout*\frac{STCUfreq}{16}\right) - 1 \tag{5}$$

To establish the correct value for such a register, the user shall follow these steps:

- Leave the default WDG value and measure the duration of the self-test (for example, using an oscilloscope)
- This is the *timeout* value in standard condition (Eq. (4))
- Extrapolate the WDG_EOC value from the equation above Eq. (5)

2.1.1 How to calculate the WDG counter during LBIST

A numeric example can help to understand better how to fix the STCU_WDG.WDG_EOC register.

The following example refers to LBIST0 run at 100MHz.

- The execution of the offline LBIST configuration with 5000 patterns takes 6.5 ms
- the STCU2 core clock is 50 MHz.

The user may configure a timeout of 6.5 ms.

According to Eq. (5) we have:

WDG_EOC = $(6.5 \cdot 10^{-3} * 50 \cdot 10^{6} /16) -1$

Considering the Eq. (5) ->**WDG_EOC** = 20312 ->**0x4F58.**

The user must consider aging and variation due to the process, voltage, and temperature. A “safe” recommendation is to choose a timeout twice the value obtained using the above equation.

In such a case considering 13 ms (instead of 6.5 ms), we obtain STCU_WDG.WDG_EOC = 0x9EB0.

It is worth mentioning that the value of the STCU2 WDG timeout is application-dependent. It depends on the system's fault-tolerant time interval (FTTI). That is, the maximum time the system has to detect and react to a failure affecting the system, the STCU2 in this case.

The user must confirm that the STCU2 WDG timeout complies with the FTTI.

Note: For a combined offline self-test configuration (MBIST + LBIST), the user shall consider that the STCU2 WDG is clocked by two different values of STCU Core clock according to which BIST is currently running, MBIST or LBIST.

3 Summary

This document describes how to configure the clock during the offline self-test on SPC58xEx and SPC58xGx.

The user shall take care of:

- Frequency limitation of each memory cut or logic part under test
- The working frequency of the STCU2 must not overcome 50 MHz.
  - The internal STCU_CFG.CLK_CFG divider of the STCU2 can be configured to reach this target

All described concepts can be easily reused for the entire SPC58 family.

4 Acronyms and abbreviations

Table 3. Acronyms

Abbreviation	Complete name
STCU2	Self -Test Control Unit
BIST	Built in self - test
MBIST	Memory BIST
LBIST	Logic BIST
MC_CGM	Clock Generation Module
IRC	Internal RC oscillator
STCU2 WDG	Internal watchdog of the STCU2
XOSC	External oscillator

5 Reference document

Table 4. Reference document

Document name	Document title
RM0391	SPC58 E/G Line - 32 bit Power Architecture automotive MCU. Triple z4 cores 180 MHz, 6 MBytes Flash, HSM, ASIL-D

Appendix A

Dedicated technical notes with recommended offline BIST setup exists for each 40 nm device and could be considered.

Here is the list of the document available:

Table 5. **List of technical note**

Reference	Doc name
TN1293	Self-test configuration for SPC584Bx devices
TN1297	Self-test configuration for SPC58ECx devices
TN1317	Self-test configuration for SPC58xNx devices
TN1319	Self-test configuration for SPC58NE8x and SPC58NG8x devices
TN1334	Self-test configuration for SPC58EHx/SPC58NHx devices
TN1400	Self-test configuration for SPC582Bx devices

Revision history

Table 6. **Document revision history**

Date	Version	Changes
04-Aug-2022	1	Initial release.

Contents