AN4405
Application note

Transition from SPC560P50x to SPC570S50x

Introduction

This application note describes common properties of, and the differences between, the SPC560P50x family and SPC570S50x family.
This document introduces only the topics that impact the porting of user application code from the SPC560P50x to the SPC570S50x. For any further details refer to the SPC560P50x and SPC570S50x specific documentations (see *A.1: Reference document*).
The SPC560P50x and SPC570S50x MCUs are both part of the microcontroller family targeted for automotive chassis applications and designed to reach a high level of compatibility.
The SPC570S50x family devices are targeted for ASIL-D applications and respect to the SPC560P50x family devices implement additional safety related features.

Both devices implement an application-independent architecture to reach its functional safety target. This application independence allows the SPC570S50x to reach a high level of compatibility with other MCUs within the chassis family, such as the SPC560P50x (or SPC56ELx).

Contents

List of tables

List of figures

1 Block diagrams

Figure 1 and *Figure 2* show a simplified version of the two architectures, highlighting similarity and differences.

They show the core architecture and other replicated IP modules from a hardware perspective (see *A.1: Reference document* for further details).

Figure 1. SPC560P50x block diagram

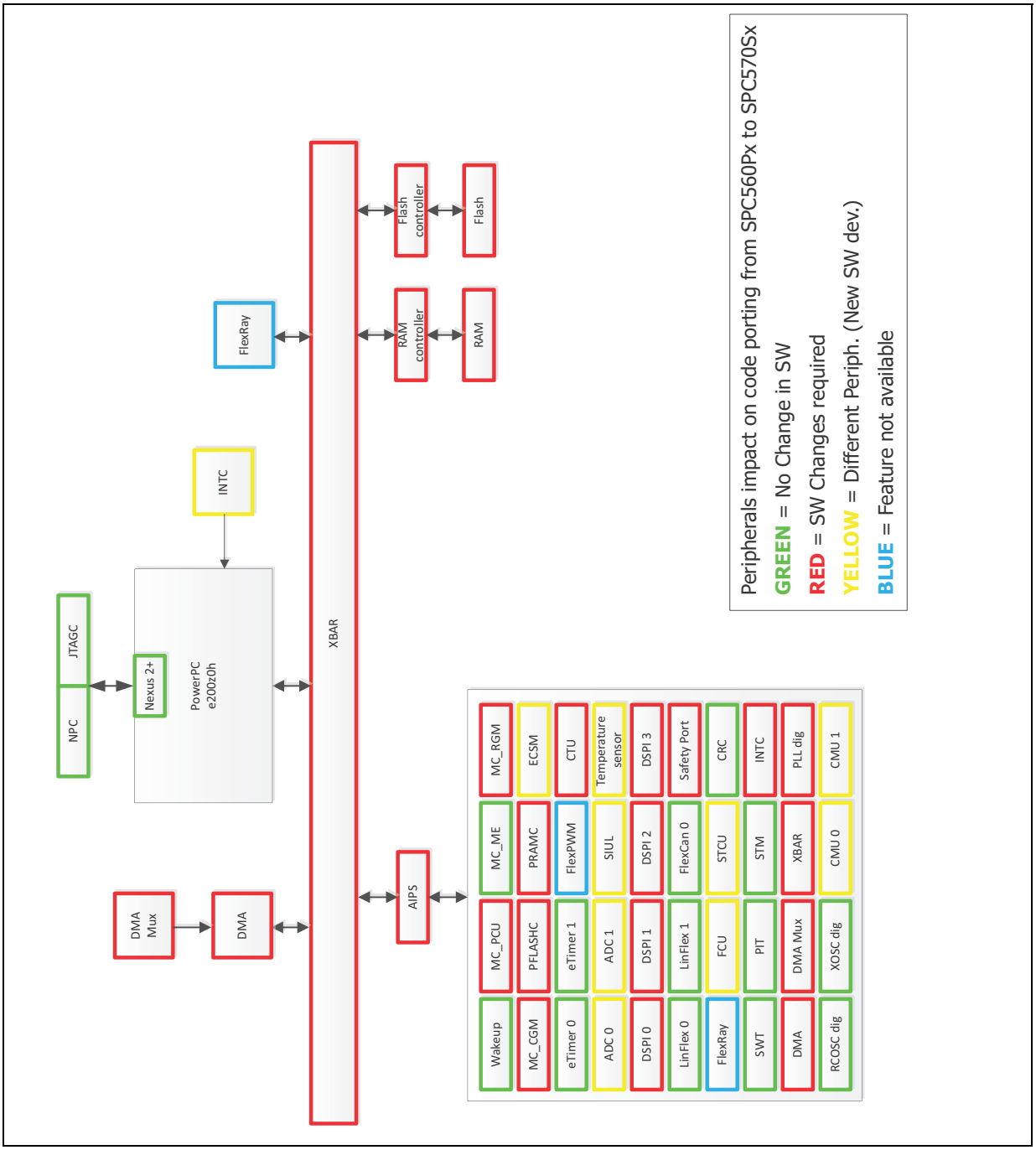
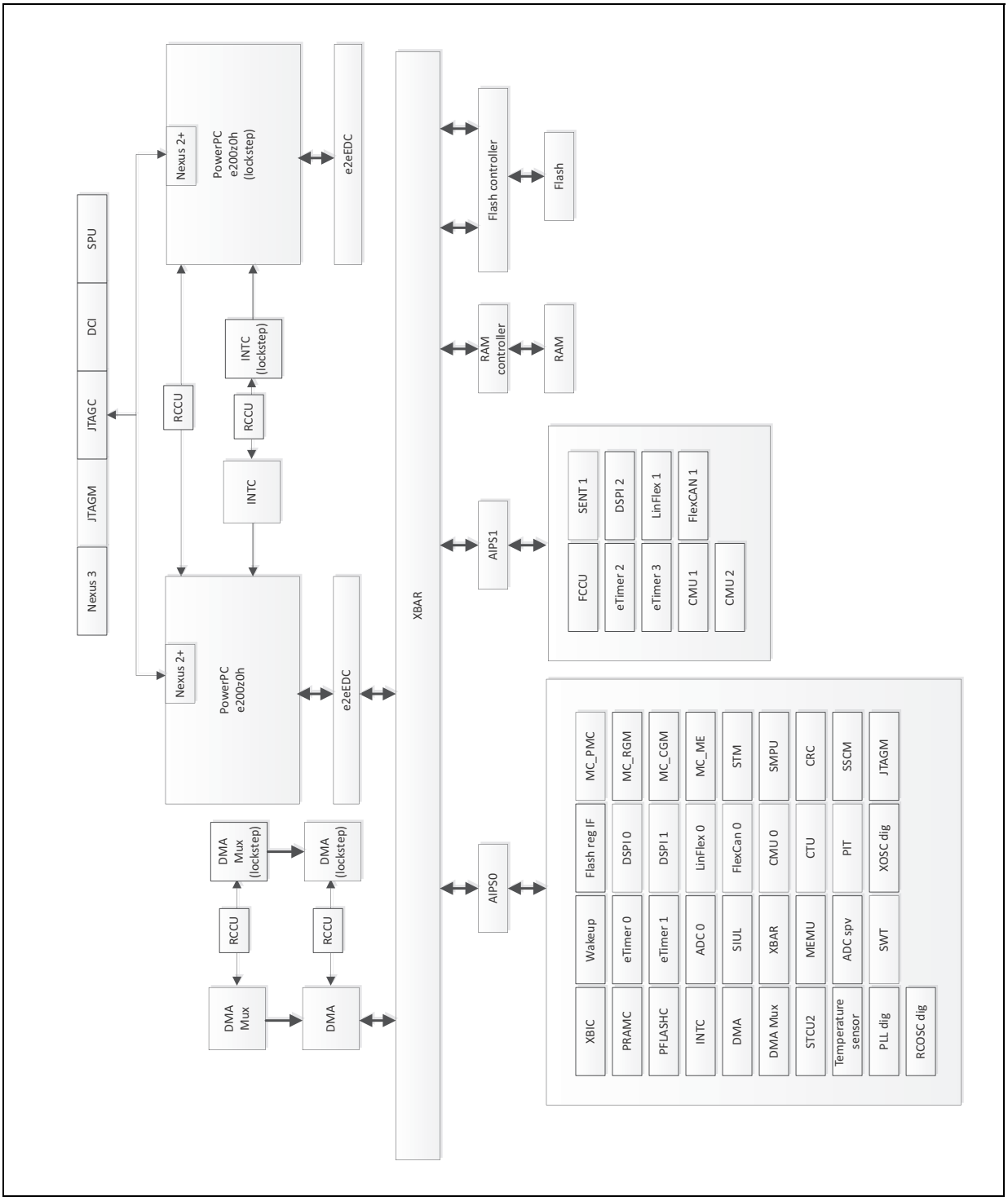


Figure 2. SPC570S50x block diagram



2 Overview

2.1 Features

Table 1 shows an overview of the features set of the two products.
A more detailed description of the differences between the two devices and of the migration from SPC560P50x to SPC570S50x is given in the next sections.

Table 1. SPC560P50x and SPC570S50x feature overview

Features	SPC560P50x	SPC570S50x
Platform - Core		
Core Type	e200 z0hn2	e200z0Hn2p (plus e200z0Hn2p in delayed lock-step)
Architecture	Harvard	Harvard
Execution speed	0–64 MHz (+4% FM)	0–80 MHz (+4% FM)
Instruction set PPC	Yes	Yes
Instruction set VLE	Yes	Yes
Platform		
System MPU	No	8 region descriptors
Crossbar (XBAR)	AHB, 32-bit address, 32-bit data 4 × 3 (master x slave ports)	AHB, 32-bit address, 32-bit data 3 x 5 (master x slave ports)
AIPS	1 (32-bit address, 32-bit data)	2 (32-bit address, 32-bit data)
Interrupt Controller (INTC)	16 interrupt levels	32 interrupt levels, replicated module
eDMA	16 channels,	16 channels, replicated module
DMA Mux	1	1, replicated modules
Platform - Memory		
Flash Memory controller	Yes	Yes
Flash	512KB code + 64KB data, ECC, RWW between code and data Flash modules	512KB code + 32 KB data, ECC, RWW
SRAM Controller	Yes	Yes
Static RAM (SRAM)	40 KB (ECC located in the ECSM)	48 KB, ECC
Reset and Boot		
SSCM	Yes	Yes
Boot Assist	BAM (Boot Assist Module)	BAF (Boot Assist Flash)
Mode handling		
MC_RGM	1	1

Table 1. SPC560P50x and SPC570S50x feature overview (continued)

Features	SPC560P50x	SPC570S50x
MC_ME	1	1
MC_CGM	1	1
MC_PMC	No	Yes
MC_PCU	Yes	No
Integration		
SIUL (System Integration Unit Lite)	1 (ver. 1)	1 (ver. 2)
CTU	1	2
Wakeup	1	1
Clocking		
Frequency-modulated phase-locked loop (FMPLL)	2	2
Internal RC oscillator	16 MHz	16 MHz
External crystal oscillator	4 – 40 MHz	8 – 40 MHz
Clock out	Yes	Yes
Timer		
Periodic Interrupt Timer (PIT)	1 × 4 channels	1 x 8 channels
System Timer Module (STM)	1 × 4 channels	1 × 4 channels,
Software Watchdog Timer (SWT)	Yes	Yes
eTimer	2 × 6 channels	4 x 6 channels each
FlexPWM	1 Module 4 × (2 + 1) channels	No
Communication		
FlexRay	1 × 32 message buffers, dual channel(1)	No
FlexCAN	1 x 32 message buffers + 1 safety port	2 × 32 message buffers
LINFlex (UART and LIN)	2	2 (with DMA support)
DSPI	4 (as many as 8 chip selects)	3 (as many as 8 chip selects)
ADC		
ADC (SAR)	2 × 10-bit ADC	1 × 12-bit ADC + 1 × 12-bit ADC supervisor
Safety		
Fault Collection (and Control) Unit	FCU	FCCU
FOSU (FCCU Output Supervision Unit)	No	Yes
MEMU	No	Yes
ECSM	Yes	No

Table 1. SPC560P50x and SPC570S50x feature overview (continued)

Features	SPC560P50x	SPC570S50x
AMBA e2eEDC	No	2 (DMA)
ISP e2eEDC	No	2 (AIPS0, AIPS1)
Cyclic Redundancy Checker (CRC) unit	Yes	Yes
RegProt (Reg. Protection)	Yes	Yes
TSENS (Temperature sens.)	No	2
Security		
PASS	No	Yes
Debug and Trace		
JTAG Master	No	Yes
JTAG Data Communication Module	No	Yes
Sequence Processing Unit	No	Yes
Debug and Calibration Interface	No	Yes
Nexus Port Controller	Yes (Level 2+)	Yes (Level 3+)
Nexus Multimaster Trace Client	No	Yes (3 modules)

1. Only for SPC560P50Lx.

2.2 Supply and packages

Table 2. SPC560P50x and SPC570S50x supply and package comparison

Features	SPC560P50x	SPC570S50x
Supply		
Digital power supply	3.3 V or 5 V single supply (with external ballast transistor)	3.3 V or 5 V single supply (with internal ballast transistor)
Analog reference voltage	3.3 V +/-10% or 5 V +/-10%	3.3 V +/--10% or 5 V +/-10%
Package		
LQFP	100 pins 144 pins	No
eTQFP	No	64 pins 100 pins
Temperature		
Temperature range (junction)	-40 to 150 °C	-40 to 150 °C

3 Platform

3.1 Core Complex

The SPC560P50x embeds an e200z0hn2 core while the SPC570S50x device embeds an e200z0Hn2p core (plus e200z0Hn2p in delayed lock-step)[a].

The two cores are 32-bit implementation of the Power Architecture Book E as defined in Power Architecture Book E Specification v 2.0. This architecture specification includes a recognition that different processor implementations may require clarifications, extensions or deviations from the architectural descriptions.

e200z0hn2 and e200z0Hn2p are unique in that they support only the VLE instruction set encodings. The VLE APU is described in *PowerPC VLE, version 1.01*

Since in the SPC570S50x family the core is in Lock step, the user has to initialize the core registers before any use. In Lock Step mode (LSM), at power on, the two cores will contain different random data and if for example there is a store to the memory (e.g. stacked) it will cause a Lock Step error.

3.2 CrossBar switch (XBAR)

The XBAR multi-port crossbar switch supports simultaneous connections between master ports and slave ports. The crossbar supports a 32-bit address bus.

The user has to reconfigure the XBAR port configuration registers in order to cover the different port assignments on the SPC570S50x.

Table 3. XBAR master port allocation for both devices

Master port	Master
M0	e200z0x CPU IBUS
M1	e200z0x CPU DBUS
M2	DMA
M3	FlexRay (Only for SPC560P50x)

Table 4. XBAR slave port allocation

Slave port	SPC560P50x	SPC570S50x
S0	Flash Controller Port 0	Flash Controller Port 0
S1	-	Flash Controller Port 1
S2	RAM Controller	RAM Controller
S3	-	AIPS0
S4	-	AIPS1

a. The Lockstep mode (LSM) is selectable by the user through a proper DCF configuration (see *Section 6.1: Boot assist*).

Table 4. XBAR slave port allocation (continued)

Slave port	SPC560P50x	SPC570S50x
S5	-	-
S6	-	-
S7	AIPS	-

3.3 AIPS

While in the SPC560P50x device all the on-chip peripherals are linked to the XBAR by a unique AIPS, in the SPC570S50x device, due to safety reasons, there are two different sets of on-chip peripherals connected to two different AIPSs.

Having two AIPS (AIPS0 and AIPS1) the user has to properly reconfigure the Master Protection Registers and the Peripheral Access Control Registers for both AIPS (see *Figure 2*).

3.4 INTC

The SPC560P50x has 16 priority levels and 147 interrupt sources versus the SPC570S50x that has 32 priority levels and 176 interrupt sources.

The SPC570S50x INTC has only one handshaking mode with the processor: the Software vector mode.

The user has to rewrite the interrupt handling taking into account the register interface, the limitation to only Software vector mode (if any), the interrupt table differences (linked to the different peripherals set and also to the different source mapping) between the two devices.

3.5 DMA and DMAMux

SPC560P50x has 16 channels eDMA with up to 21 DMA sources (slots) while SPC570S50x has up to 16 channels with up to 59 DMA sources.

The user has to reprogram the DMA Mux sources to take into account the increased number of sources and the new mapping.

4 Memory

4.1 SRAM controller

The SRAM controller acts as an interface between the AHB system bus and the integrated system RAM.

From the application point of view the main difference between the SRAM controllers implemented into the two devices is:

- The error reporting: SPC560P50x reports to the ECSM while SPC570S50x reports to the MEMU

4.2 SRAM

The SPC560P50x has 40 KB of general-purpose static SRAM while the SPC570S50x has 48 KB of general-purpose Static SRAM: this allows the user to have available more memory space.

4.3 Flash controller

The flash controller for the two devices behaves in different way.
In the SPC570S50x device a mini-cache has been introduced (instead of the line buffers implemented on SPC560P50x) to improve the performances allowing single-cycle (zero AHB wait-states) read data responses on buffer hits.

The user has to reprogram the Flash controller accordingly with the new features introduced into SPC570S50x.

Below a comparison of the features:

Table 5. Flash controller comparison between SPC560P50x and SPC570S50x

Parameter/feature	SPC560P50x	SPC570S50x
Number of Flash controller ports connected to the XBAR slave ports	1	2
AHB data bus width	32	32
Flash write data bus width	64	64
Flash read data bus width	128	128
Buffer	Four line read buffers and a prefetch controller for each port.	Mini-cache size (per Flash controller port) 2 ways, each way containing 4 x 128 bit buffers

4.4 Flash

In the SPC560P50x the flash memory module provides the following features:

- 576 KB flash memory:
  - 8 blocks (32 KB + 2×16 KB + 32 KB + 32 KB + 3×128 KB) code flash
  - 4 blocks (16 KB + 16 KB + 16 KB + 16 KB) data flash
- Full Read While Write (RWW) capability between code and data flash

In the SPC570S50x there is 560 KB Flash Memory module containing two array partitions:

- 10 blocks (4x16 KB, 2x32 KB, 2x64 KB, 2x128 KB) code flash plus 8 KB UTest and 8 KB BAF
- 4 blocks (8 KB + 8 KB + 8 KB + 8 KB) data flash
- Read-While-Write (RWW) operations are only possible between partitions, meaning fetching/reading from one (or more) partition(s) while a program/erase operation is active on one other partition. Each module counts as an RWW partition.

The user has to take into account that there is a smaller data flash in the SPC570S50x.

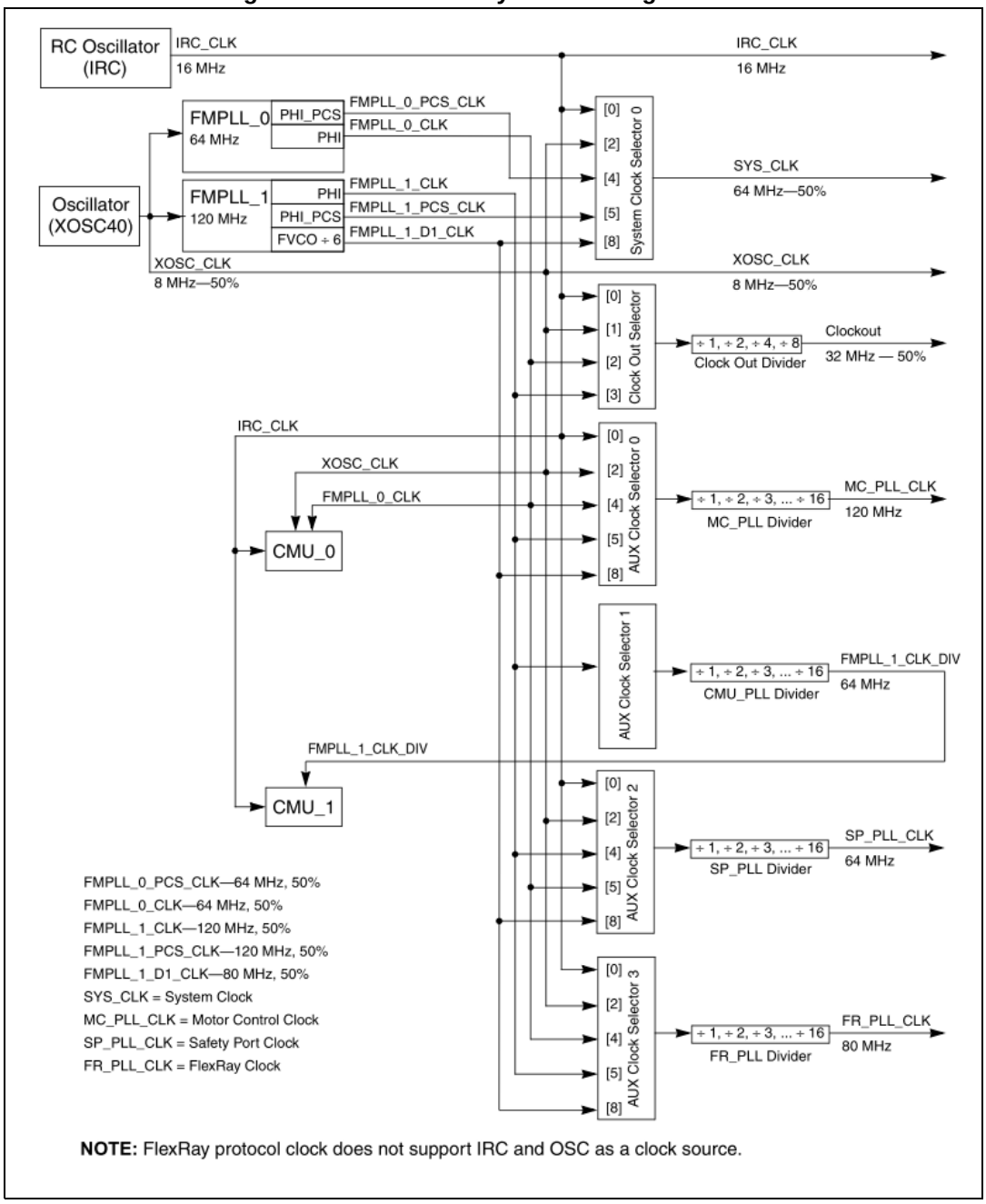
5 Clocking

The Clock architecture is different for the two devices.
The main difference is that in the SPC560P50x the peripherals clock (with the exception of Motor Control peripherals, FlexCAN and FlexRay) is the same as the platform clock.

In the SPC570S50x there is a 1:2 ratio between the platform and peripheral domain clocks.

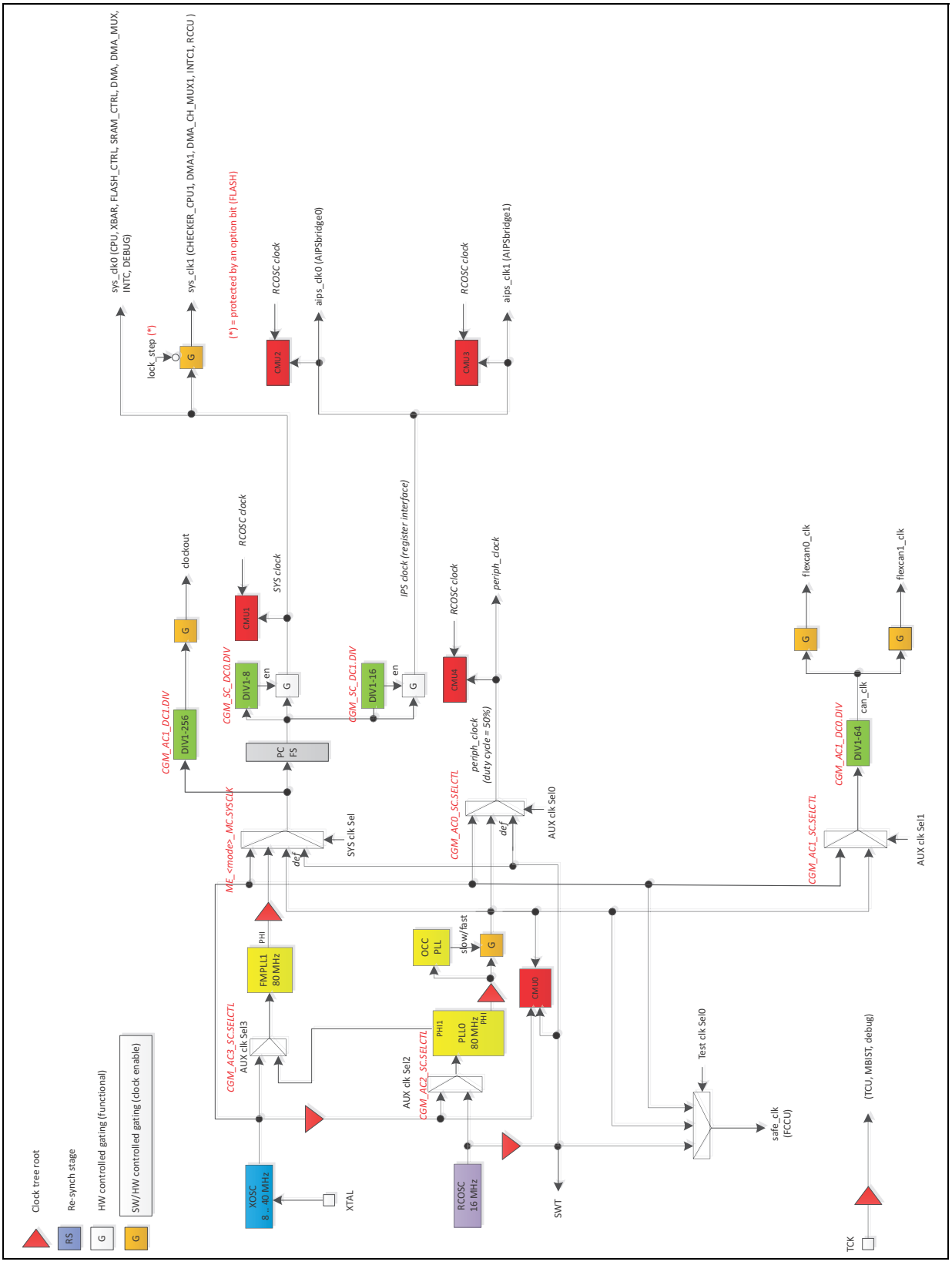
The user has to reprogram the MC_CGM configuration accordingly with changed clock architecture.

Figure 3. SPC560P50x system clock generation



NOTE: FlexRay protocol clock does not support IRC and OSC as a clock source.

Figure 4. SPC570S50x System clock generation



5.1 PLL

The SPC560P50x and SPC570S50x devices implement different versions of this peripheral. Even if there are no functional differences between the two devices, the user shall consider the different configurations of FMPLL modules.

5.2 XOSC

The minimum acceptable value for both devices is 4 MHz.

In the SPC570S50x there is a limitation that the user has to consider: the minimum frequency for the XOSC is 8 MHZ if it drives the PLL0 clock input.

6 Reset and boot

The reset and boot phases are handled in different ways in the two devices.

Figure 5 shows the various possible system boots for both devices.

In the SPC570S50x the application entry point can be retrieved both by HW (by the SSCM) or in SW (by the BAF) while in the SPC560P50x it is retrieved only by HW (by the SSCM).

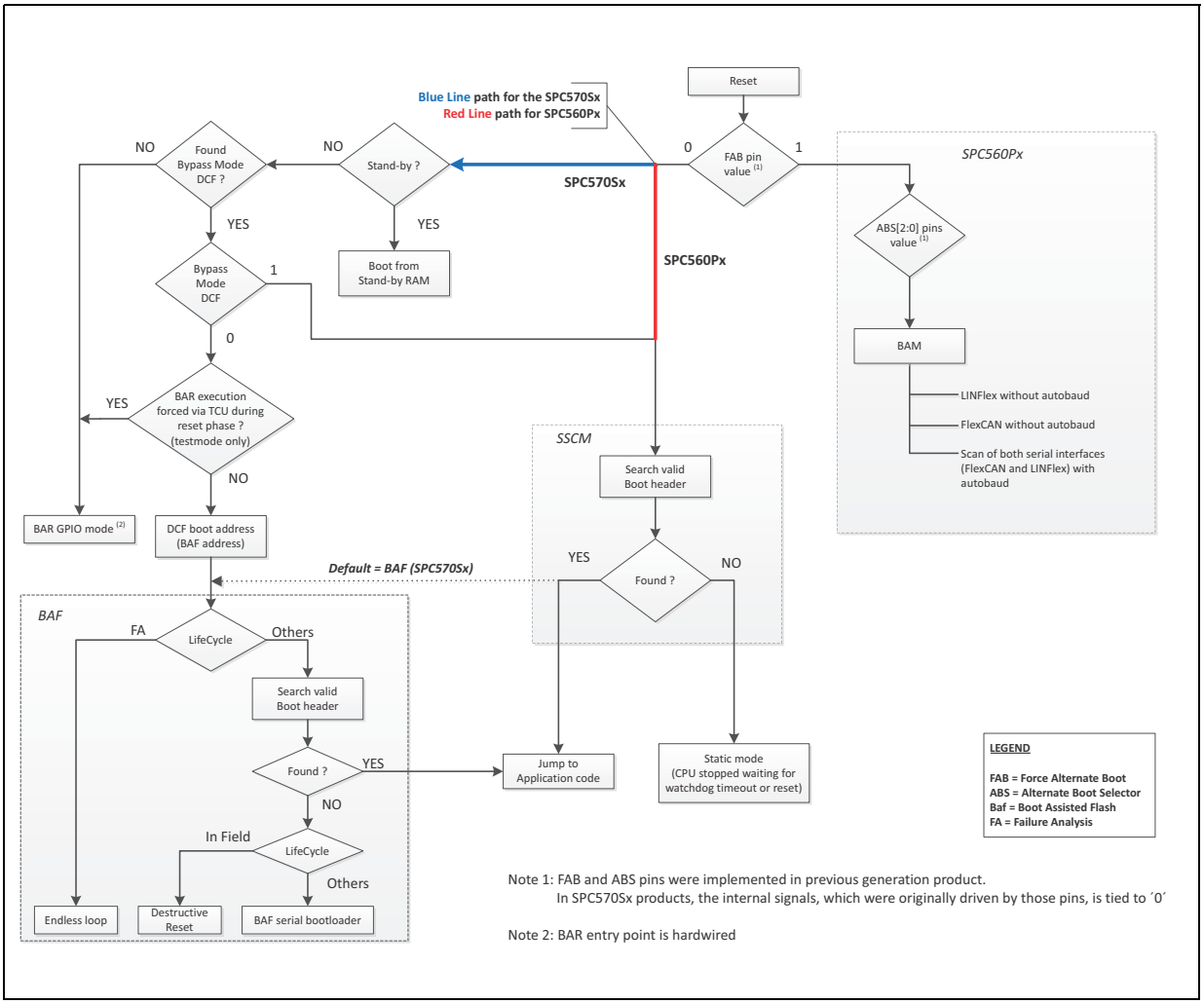
In both cases, the application entry point is retrieved by searching for a valid Boot Record, which shall be programmed by the customer in one of the pre-determined Flash memory locations, different between the devices (see *A.1: Reference document* for further details).

If the application entry point is retrieved by the SSCM the user (apart the different bootable sector where he place his own code) has to change nothing in the application code. Vice-versa if the application entry point is retrieved in SW through the BAF the user has to change the reset vector format.

Table 6. SPC570S50x boot record structure

Offset	Boot strategy	Field description	
0x0	SW (BAF)	BAF Boot Record Tag (A5h)	Reserved
	HW (SSCM)	SSCM Boot Record Tag (5Ah)	Reserved
0x4	SW (BAF)	Application Start Address	
	HW (SSCM)		
0x8	-	Reserved	
0xC	-	Reserved	

Figure 5. System boot sequence for SPC570S50x and SPC560P50x



6.1 Boot assist

The SPC560P50x device implements the BAM (Boot Assist Module) that is a block of read-only memory containing VLE code which is executed according to the boot mode of the device. The code stored in the BAM is not executed when booting in Single Chip mode (see red path in the *Figure 5*), except when entering the "Static mode" in case no valid bootable sector has been found.

The BAM downloads code into internal SRAM through the following serial protocols and executes it afterwards:

- FlexCAN
- LINFlex-UART

Similar mechanism is present in the SPC570S50x device that uses the BAF (Boot Assist Flash) to boot the device.

The MCU is booted through a collaboration of several blocks, hardware and firmware.

The first boot phases are performed by a state machine inside the System Status and Configuration (SSCM).

Once completed, the SSCM sends a reset vector to the core on pointing into the Boot Assist Flash (BAF).

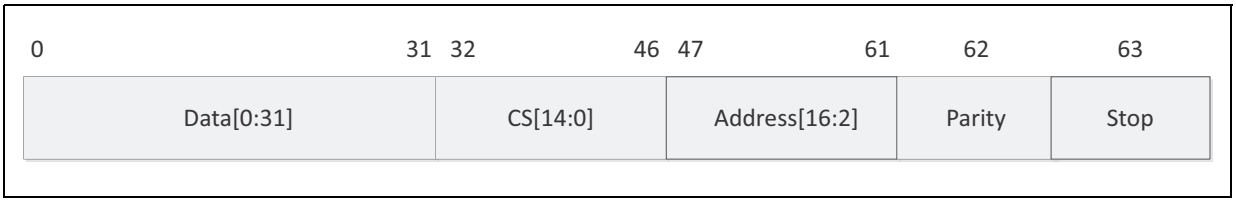
The BAF code does the following main tasks:

1. Search for a valid boot record, where the application entry point is specified
2. Parse and decode the SW-DCF records
3. Execute the BAF serial boot if no valid application entry point is found

Moreover SPC570S50x has certain memory regions, called DCF, that contain system configuration information, used to configure the device during the reset phase. Each DCF is structured as shown in *Figure 6.*
The part of the system (called “DCF client”) receiving the DCF data is selected by the “CS” field.

The “address” field is used to distinguish different types of information for a DCF client. The configuration data is stored within the “Data” field. The “Parity” field is used to check the DCF validity vs. a parity check. The “Stop” field is used to terminate the DCF reading operation.

Figure 6. DCF structure



6.2 SSCM

The primary purpose of the SSCM is to provide information about the current state and configuration of the system that may be useful for configuring application software and for debug of the system.

This module behaves in different way for the two devices (see *A.1: Reference document* for further details).

7 Mode handling

7.1 MC_ME

No functional differences between the two devices.

7.2 MC_RGM

The peripheral implemented on the two devices is different and the user has to program it accordingly.

The RGM implemented in SPC570S50x‘s doesn't handle:

- Bidirectional reset behavior configuration
- Boot mode capture on RESET_B deassertion

but the following features have been introduced:

- Configurable escalation of recurring ‘functional’ resets to ‘destructive’ reset
- Configurable escalation of recurring ‘destructive’ resets to keep chip in reset state until next power-on reset

7.3 MC_CGM

The peripheral implemented in the SPC570S50x is different from the one implemented in the SPC560P50x.

The user has to reprogram it according to the SPC570S50x clock architecture (see *Chapter 5: Clocking*).

7.4 MC_PCU

In the SPC560P50x the power management is handled by the MC_PCU (Power Control Unit) and the PMU (Power Management Unit): the first peripheral is digital, the second is analog. In the SPC570S50x there is the Power Management Controller (MC_PMC) which consists of an analog block and a digital block.

Since the concept of the power management as well as the peripherals involved in the two devices are different, the user has to reprogram/remapping the functionalities.

8 Integration

8.1 SIUL

The devices (SPC560P50x, SPC570S50x) implement different versions of the peripheral and the user has to reprogram it taking into account, either the SPC570S50x's pin mapping or the new peripheral features.

Figure 7. SPC560P50x SIUL block diagram

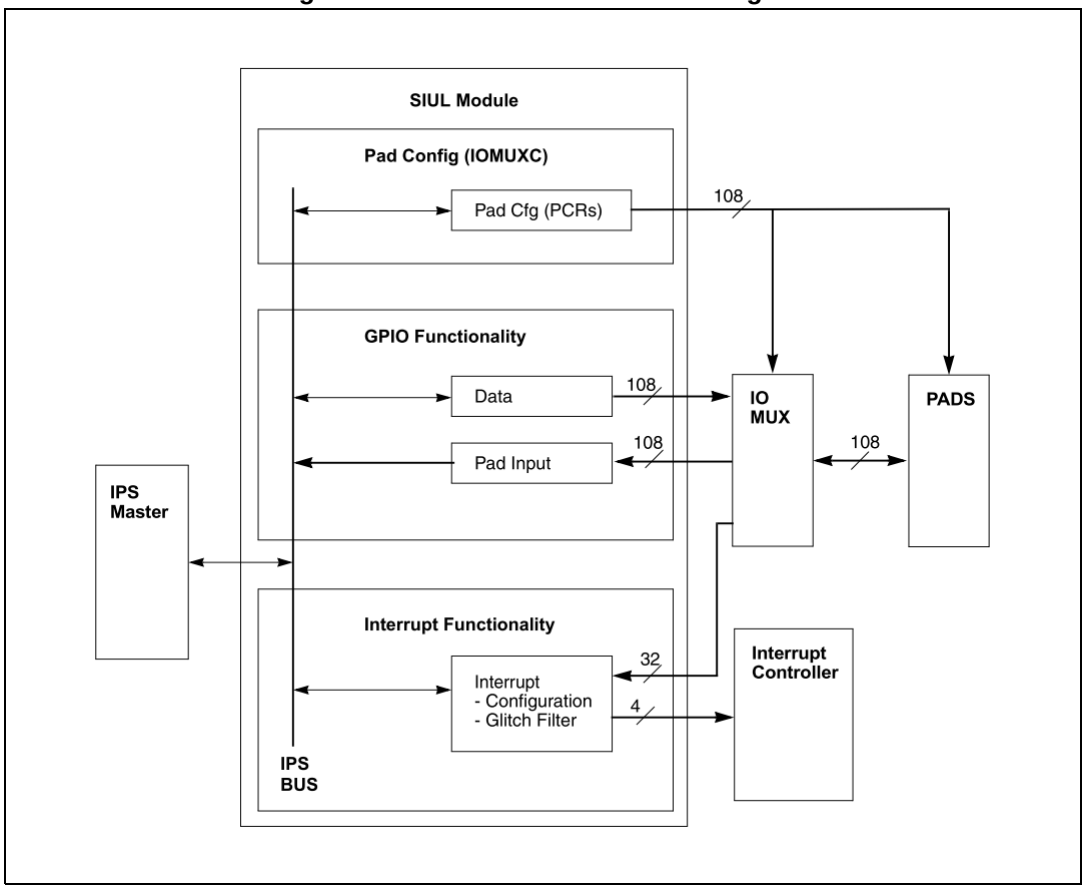


Figure 8. SPC570S50x SIUL output pin driver

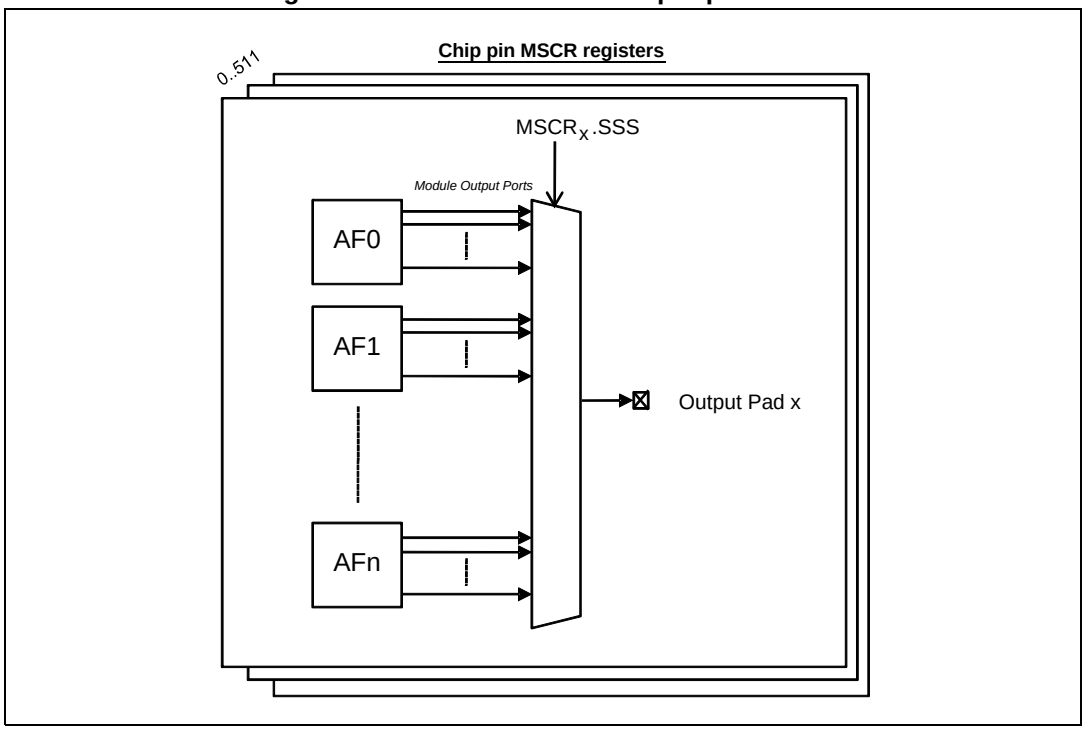
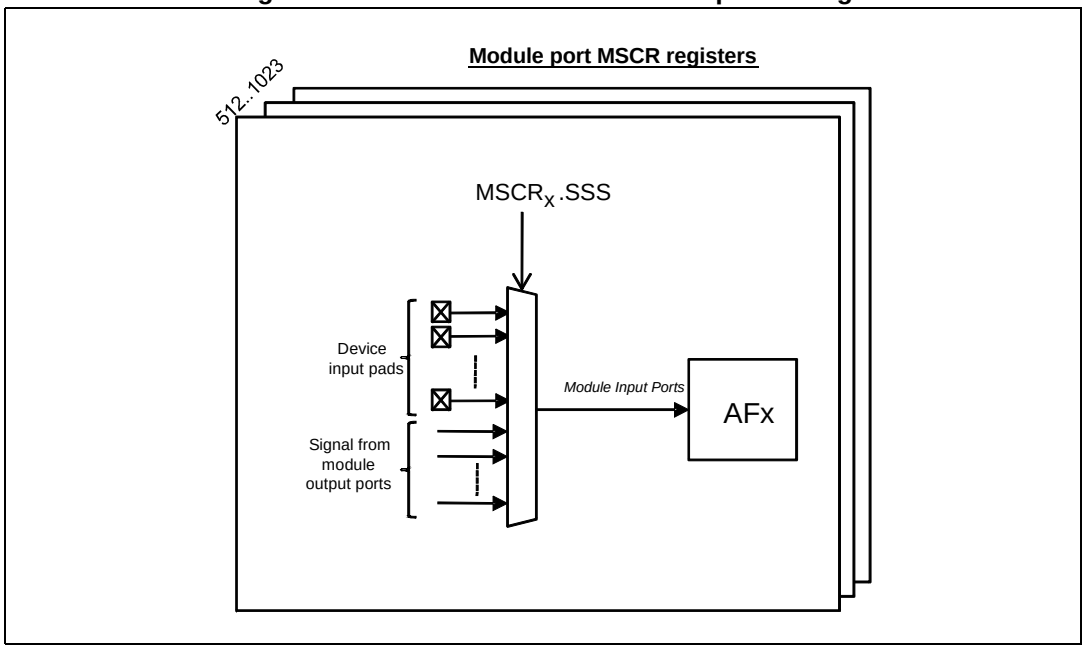


Figure 9. SPC570S50x SIUL module input driving



8.2 CTU

The CTU (Cross Trigger Unit) is used to provide commands to the ADCs, trigger to other IP, and generate an external trigger.

There are no functional differences between the two devices but since in the SPC570S50x there are no FlexPWM modules the user has to reconfigure its own code to take in account the peripheral interface changes.

Table 7. CTU input comparison between SPC560P50x and SPC570S50x

Peripheral Interface	SPC560P50x	SPC570S50x
FlexPWM inputs	13	0
eTimer inputs	2	12
GPIO (external input signal)	1	1

The block diagram of the SPC560P50x's CTU is shown in *Figure 10* while the block diagram related to the SPC570S50x is shown in *Figure 11*.

Figure 10. SPC560P50x cross triggering unit diagram

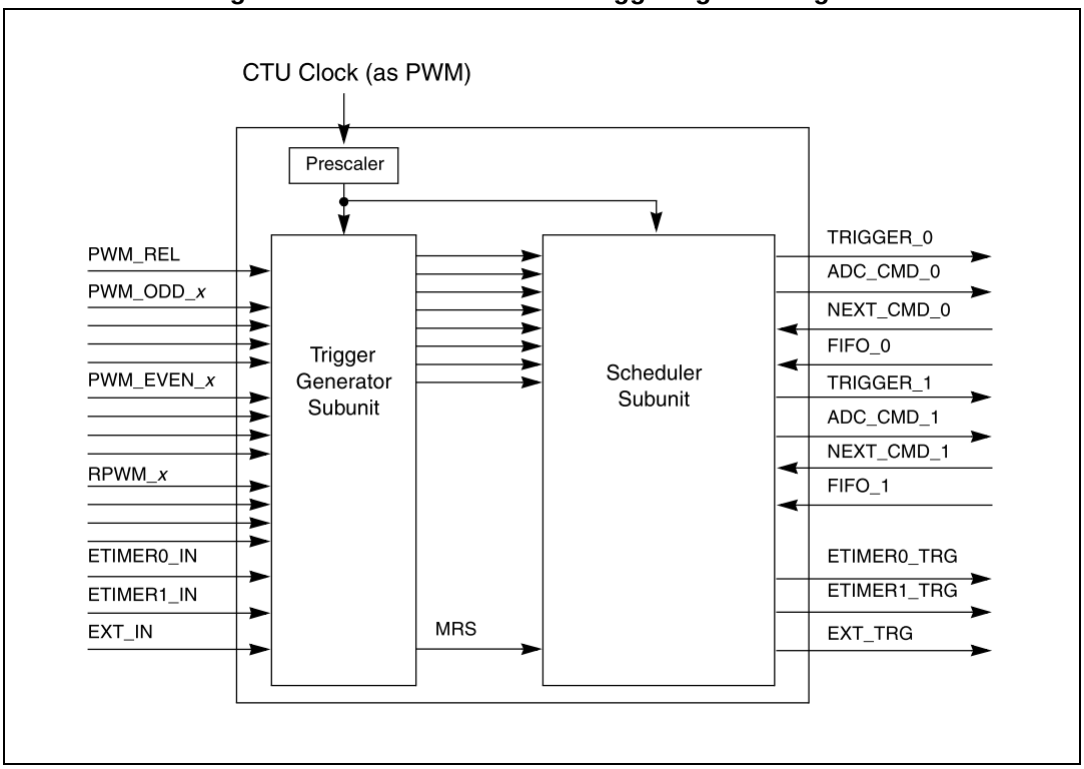
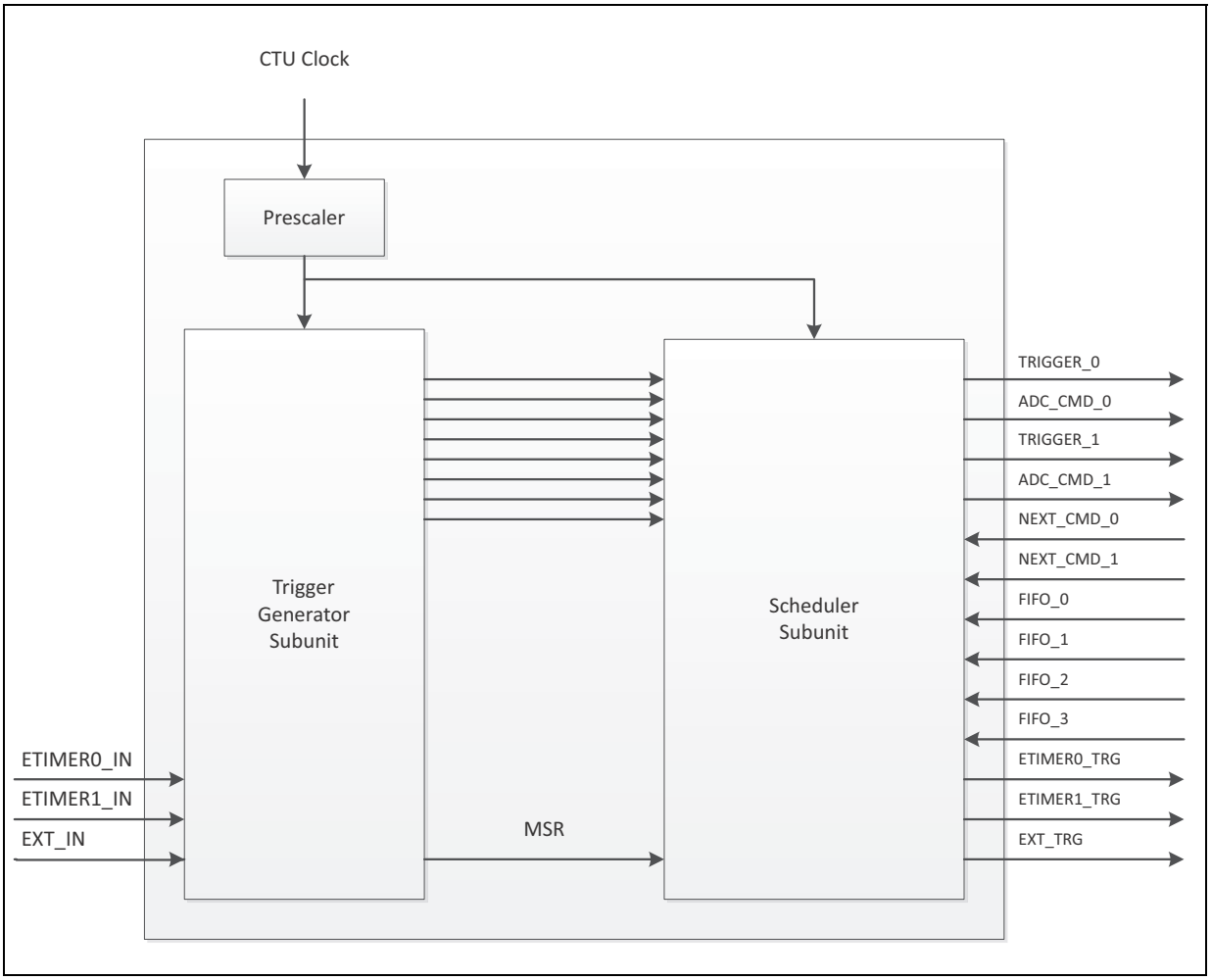


Figure 11. SPC570S50x cross triggering unit diagram



8.3 Wakeup

From the functional point of view the peripheral behaves in the same manner as both SPC560P50x and SPC570S50x.

9 Timer

9.1 STM

STM embbeded on both devices are basically the same. The user has to take into account only the different operating frequency of the peripheral between the two devices (see *Chapter 5: Clocking*).

9.2 PIT

The user has to take into account the different operating frequency of the peripheral between the two devices (see *Chapter 5: Clocking*).

9.3 eTimerVelvety

The SPC560P50x has only two modules while the SPC570S50x has four (6 channels) modules.

No functional differences between the two devices.

9.4 SWT

No functional differences between the two devices.

9.5 FlexPWM

The SPC570S50x doesn't have any FlexPWM module so that the user has to rewrite its own code using the eTimer modules instead of FlewPWM modules for PWM signal generation or emulating the PWM functionalities through the GPIOs.

10 Communication

10.1 FlexCAN

The SPC560P50x has two FlexCAN modules (32 message buffers) and the second module can act as a Safety Port with a bit rate as high as 7.5 Mbit/s.

The SPC570S50x has two identical FlexCAN modules so that the user has to take into account the lower bit rate if the Safety port is used in his own code.

10.2 LinFlex

The LINFlex modules implemented on both devices are the same from the functional point of view even if the SPC570S50x also implements the DMA support and an enhanced UART mode (compatible with SPC560P50x) that also handle 12 bit + parity frame reception to support MicroSecond Channel (MSC) communications.

10.3 DSPI

The DSPI for the two devices are quite similar. Below a short comparison in which the main differences are highlighted.

Table 8. DSPI comparison between SPC560P50x and SPC570S50x

Description	SPC560P50x	SPC570S50x
Number of modules	4	3
Transmission Speed	16 MHz (max speed)	14 MHz (shall guarantee minimum 2.5 Mb/s)
Frame size	Programmable serial frame size of 4 to 16 bits, expandable by software control	Programmable (from 4-bits up to 32-bits in extended mode)
Number of chip select	8	8
Number of transfer descriptors	4	8
Extension to 32-bit frames and multi-frame transmission within the same TX command	0	1
Command FIFO depth	5	4
TX FIFO depth	5	4
TX FIFO depth	5	4

The user has to take into account in his own code the different FIFO depth present on both devices and the reduced number of modules present on SPC570S50x (three instead of four).

10.4 FlexRay

This module has not been implemented on the SPC570S50x so that, the user has to rewrite its own code taking into account this limitation.

11 ADC

Since the peripherals are different for the two devices the user has to rewrite the code for his application and has to take into account the reduced channel availability.
The ADC units for both devices contain advanced features for normal or injected conversion. The conversions can be triggered by software or hardware (CTU).
Below a short comparison where the main differences are highlighted.

Table 9. ADC comparison between SPC560P50x and SPC570S50x

Description	SPC560P50x	SPC570S50x
Modules	2	1 + supervisor
Conversion time	<1 µsec (min)	1.5 µsec (each channel)
TUE (without current injection)	2.5 LSB	2 LSB
Maximum operating frequency	120 MHz	12 MHz
Resolution	10 bit	12 bit
Channels	15 (11 + 4 shared with other ADC unit) Only for ADC_0 unit channel 15 is dedicated for the internal 1.2 V rail	16 (fully shared between the 2 ADC units)
DMA Triggers	Yes	Yes
Watchdog feature	Yes	Yes

12 Safety

12.1 FCU

The SPC570S50x doesn't implement the FCU but implements a more complex peripheral (Fault Collection and Control Unit) that apart the fault collection it is also able to control (program) the fault reaction handling.

The FCCU shall implement the following features:

- 1 to 128 critical faults (static parameter) management
- 1 to 128 non-critical faults (static parameter) management
- HW or SW fault recovery management
- Fault detection collection
- Fault injection (fake faults)
- External reaction (fault state): EOUT signaling
- Internal (SOC) reactions (alarm state): interrupt request
- Internal (SOC) reactions (fault state): long functional reset, short functional reset
- NMI, safe mode request
- Bi-Stable, Dual-Rail and Time Switching output protocols on EOUT
- Watchdog timer for the re-configuration phase
- Configuration lock
- NVM configuration loading
- Self-checking capabilities: FSM redundancy and parity check for the configuration registers

There is also the presence of a monitor (FOSU) that supervises FCCU behavior.
The user has to rewrite his code to take into account this new peripheral and all its features considering the constraints introduced by the safety architecture implemented on the SPC570S50x.

12.2 MEMU/ECSM

The two devices have different peripherals to handle the memory error reporting.

12.2.1 SPC560P50x: ECSM

The Error Correction Status Module (ECSM) provides miscellaneous control functions for the device platform including program-visible information about the platform configuration and revision levels, a reset status register, and wakeup control for exiting sleep modes, and optional features such as information on memory errors reported by error-correcting codes.

12.2.2 SPC570S50x: MEMU

The MEMU is responsible for collection and reporting of error events associated with ECC (Error Correction Code) logic used on SRAM, Peripheral System RAM, and Flash memory. The MEMU receives an error signal which causes an event to be recorded and corresponding error flags to be set and reported to FCCU.

13 Debug and trace

13.1 JTAG controller

No functional differences between the two devices.

13.2 Nexus development interface

No functional differences between the two devices.

Appendix A Further information

A.1 Reference document

1. *32-bit MCU family built on the Power Architecture® embedded category for automotive chassis and safety electronics applications* (RM0022, DocID14891)
2. *32-bit Power Architecture® based MCU with 576 KB Flash memory and 40 KB SRAM for automotive chassis and safety applications* (SPC560P44L3, SPC560P44L5, SPC560P50L3, SPC560P50L5, DocID14723)
3. *32-bit Power Architecture® microcontroller for automotive ASILD Chassis & Safety applications* (RM0349, Doc ID024507)
4. *32-bit Power Architecture® microcontroller for automotive ASILD Chassis & Safety applications* (SPC570S50L1, SPC570S50L3, DocID024492)
5. *Safety Manual for SPC570S50L1/SPC570S50L3* (AN4247, DocID024209)
6. *e200z0h Power Architecture™ Core Reference Manuals*

A.2 Acronyms and abbreviations

Table 10. Acronyms and abbreviations

Terms	Meaning
BAF	Boot Assist Flash
BAM	Boot Assist Module
BTB	Branch Target Buffer
CMPU	Core Memory Protection Unit
CPU	Central Processing Unit
DCache	Data Cache
DCF	Device Configuration Format
DMA	Direct Memory Access
DMEM	Data Memory (internal to the core)
DPM	Decoupled Parallel Mode
ECC	Error Correcting Code
EDC	Error Detection and Correction
FCCU	Fault Collection and Control Unit
GPIO	General purpose input/output
ICache	Instruction Cache
IMEM	Instruction Memory (internal to the core)
LSM	Lock Step Mode
MC_CGM	Clock Generation Module
MC_ME	Mode Entry Module
MC_RGM	Reset Generation Module

Table 10. Acronyms and abbreviations (continued)

Terms	Meaning
MPU	Memory Protection Unit
PMC	Power Management Controller
PLL	Phase Locked Loop
SIPI	Serial Interprocessor Interface
SoC	System on Chip
SoR	Sphere of Replication
SMPU	System Memory Protection Unit
SSCM	System status and configuration module
STCU2	Self-Test Control Unit ver.2
SWT	Software Watchdog Timer
TCM	Tightly-Coupled Memory
XBAR	Crossbar Switch

Revision history

Table 11. Document revision history

Date	Revision	Changes
03-Jan-2014	1	Initial release.
05-Apr-2016	2	Removed “ST Restricted” watermark throughout document. Updated Disclaimer.