

# Migrating from other KL serial parts to KL28

## Contents

### 1. Introduction

This document describes the migration from other KL serials part to KL28. As there's some new IP exists on KL28, in this AN, the new features in these IPs are introduced. This application note gives an overview description here, and for some details, we have provided other separate ANs for reference which would be mentioned in the corresponding chapter.

### 2. New module list

The new module exist on KL28 includes:

- SCG/PCC
- LPSPI
- LPI2C
- FLEX IO with enhanced feature
- TRGMUX
- True random number generator
- Time stamp timer
- Smart card interface module
- Low power periodic interrupt timer
- Memory-mapped divide and square root

1.	Introduction .....	1
2.	New module list .....	1
3.	Common feature in smart peripherals.....	2
4.	New module introduction .....	2
4.1.	SCG/PCC .....	2
4.2.	LPSPI .....	7
4.3.	LPI2C .....	9
4.4.	FLEX IO with enhanced feature.....	13
4.5.	TRGMUX for peripheral triggering .....	15
4.6.	True random number generator .....	16
4.7.	Time stamp timer .....	16
4.8.	Smart card interface module.....	17
4.9.	Low power periodic interrupt timer.....	18
4.10.	Memory-mapped divide and square root.....	19
5.	Large memory configuration .....	20
5.1.	Improvement .....	20
6.	Revision history.....	21

### 3. Common feature in smart peripherals

- Functional clock is independent of Bus clock
  - Allows dynamic frequency scaling without reconfiguring timer/baud rate
- Module optionally remains fully functional in Stop modes
  - Provided functional clock remains enabled
- Support DMA Driven operation
- Data match capability to optionally discard unwanted receive data
- Trigger capability to synchronize operation with other modules

### 4. New module introduction

#### 4.1. SCG/PCC

##### 4.1.1. Block diagram

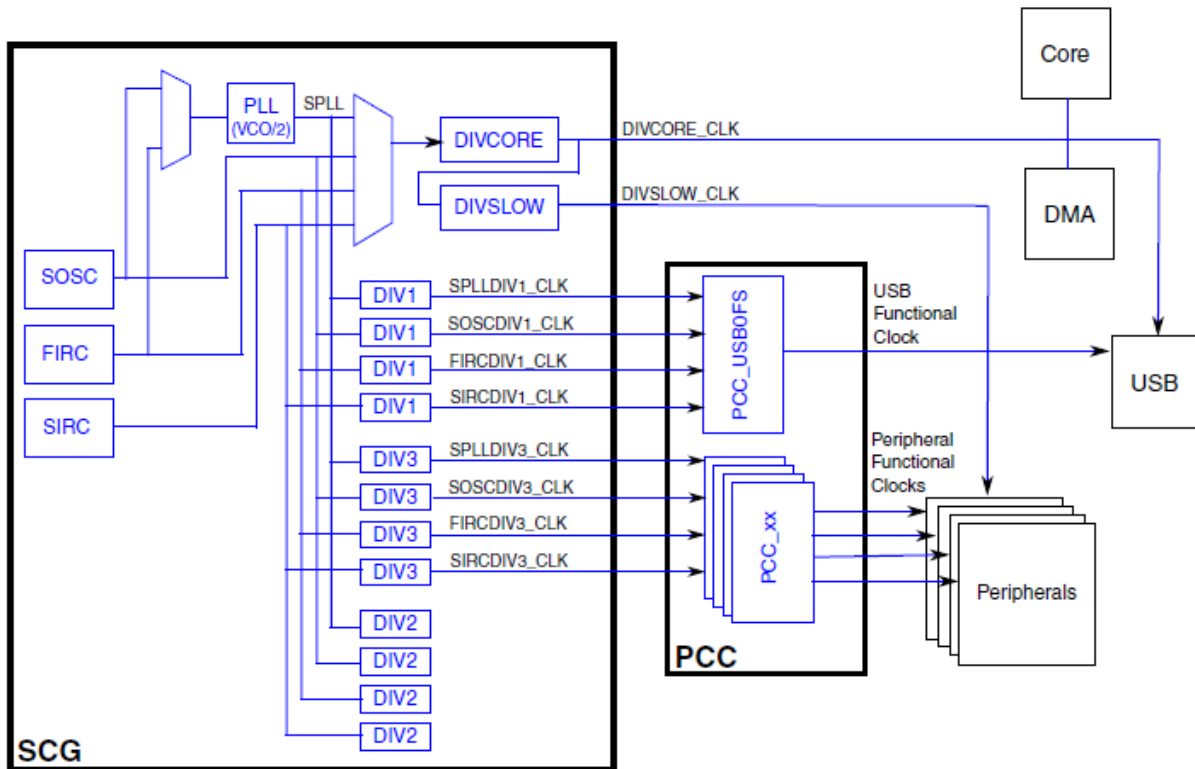


Figure 1. Block diagram of new module introduction

SCG/PCC brings a new clock architecture compared with other KL parts's MCG(MCG-Lite) module. It makes the clock source configuration more flexible.

## 4.1.2. Clock source

For the SCG clock source, please refer to the following table:

**Table 1. SCG clock source**

Clock Source	Description	Additional Comment
SOSC	Output of the external oscillator, a crystal or externally applied clock input	32–40 kHz, or 3–32 MHz crystal oscillator, can be used as the clock source for PLL, RTC or System/Peripheral
SIRC	Output of the slow (2/8M Hz) internal RC oscillator	2/8M Hz is set by SCG_SIRCCFG[RANGE]
FIRC	Output of the fast ( 48/52/56/60M Hz) internal RC oscillator	48/52/56/60M Hz is set by SCG_FIRCCFG[RANGE]

## 4.1.3. SPLL

### 4.1.3.1. Feature list

- Typical output frequency is 72MHz for RUN mode or 96MHz for HSRUN mode.
- PLL output frequency is half of VCO output frequency.
- Voltage-controlled oscillator (VCO)
- Selectable Internal or External reference clock is used as the PLL source
- Modulo VCO frequency divider
- Phase/Frequency detector
- Integrated loop filter
- Can be selected as the clock source for the MCU system clocks
- 2 programmable post-dividers clock outputs, which can be used as clock sources for other on-chip peripherals

## 4.1.4. PCC

The PCC (Peripheral Clock Control module) provides peripheral clock control and configuration functions. The main function is clock gate configuration for peripherals, which is controlled by PCC\_XXX[CGC]. It is a replacement of some of the functions in SIM module on other part.

### 4.1.4.1. Block diagram

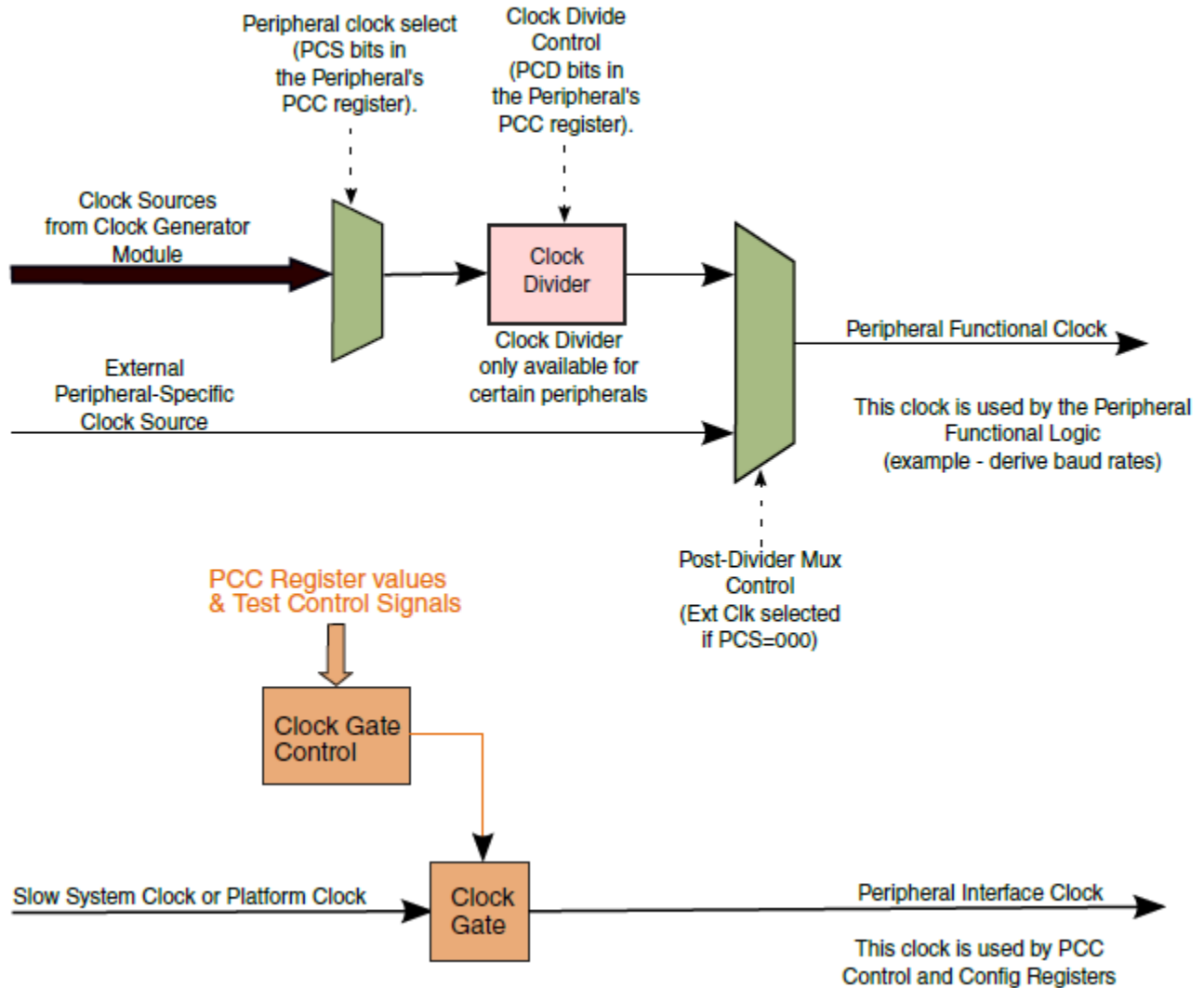


Figure 2. Block diagram

### 4.1.4.2. Feature list

- Clock gating
- Clock source selection
- Clock divide values

### 4.1.4.3. Improvement

**Table 2. PCC improvement**

Feature	KL28Z	Old KL device	Benefit
Peripheral interface clock and functional clock can be gated separately	Yes	No. Clock gating is only controlled by SIM.	Helps reduce further power consumption when peripherals automatically transfer data without CPU intervention
Unified 2 <sup>nd</sup> level of peripheral clock source selection with divider	Yes	Peripheral clock selection and divider are scattered in SIM and inside Peripheral module	Easy to select peripheral clock source, user friendly way.
Unified peripheral clock control register names	Yes	No	Software oriented design makes code compatible

### 4.1.5. Clock configuration for different power mode

One key improvement on KL28 is, for different power mode like HSRUN, RUN and VLPR, SCG proved separate configuration registers, which brings an advantage that when switch from one power mode to another power mode, we do not need to configure the clock again and again. Instead, we can switch power mode directly.

The register list and corresponding power mode are listed in the table below:

**Table 3. SCG register list and corresponding power mode**

Register	Corresponding power mode
SCG_RCCR	RUN
SCG_VCCR	VLPR
SCG_HCCR	HSRUN

### 4.1.6. How to check if current clock setting is valid

After we finished configuring the clock, we need to double confirm if the clock settings are working correctly. This can be implemented by routing the clock signal out to a dedicate clock output pin PTC3, then we can watch it in a scope. For example, we can watch the SLOWCLK or PLLCLK in scope to evaluate if current clock setting is working correctly.

To implement this, the following code snippet can be referred to:

The code snippet:

## New module introduction

```
#define CLK_SRC_SLOWCLK 0
#define CLK_SRC_OSCCLK 1
#define CLK_SRC_SIRC 2
#define CLK_SRC_FIRC 3
#define CLK_SRC_PLL 6

void clk_out(void)
{
    PCC_PORTC |= PCC_CLKCFG_CGC_MASK;
    PORTC_PCR3 = PORT_PCR_MUX(5) | PORT_PCR_DSE_MASK;
    SCG_CLKOUTCNFG = SCG_CLKOUTCNFG_CLKOUTSEL(CLK_SRC_PLL);
}
```

Also there are other ways to check if clock is set correctly. Like use a timer with clock source from bus clock and count its value or use sys-tick.

For more information about SCG/PCC please refer to AN5231 Clock management and distribution in KL28.

## 4.2. LPSPI

### 4.2.1. Block diagram

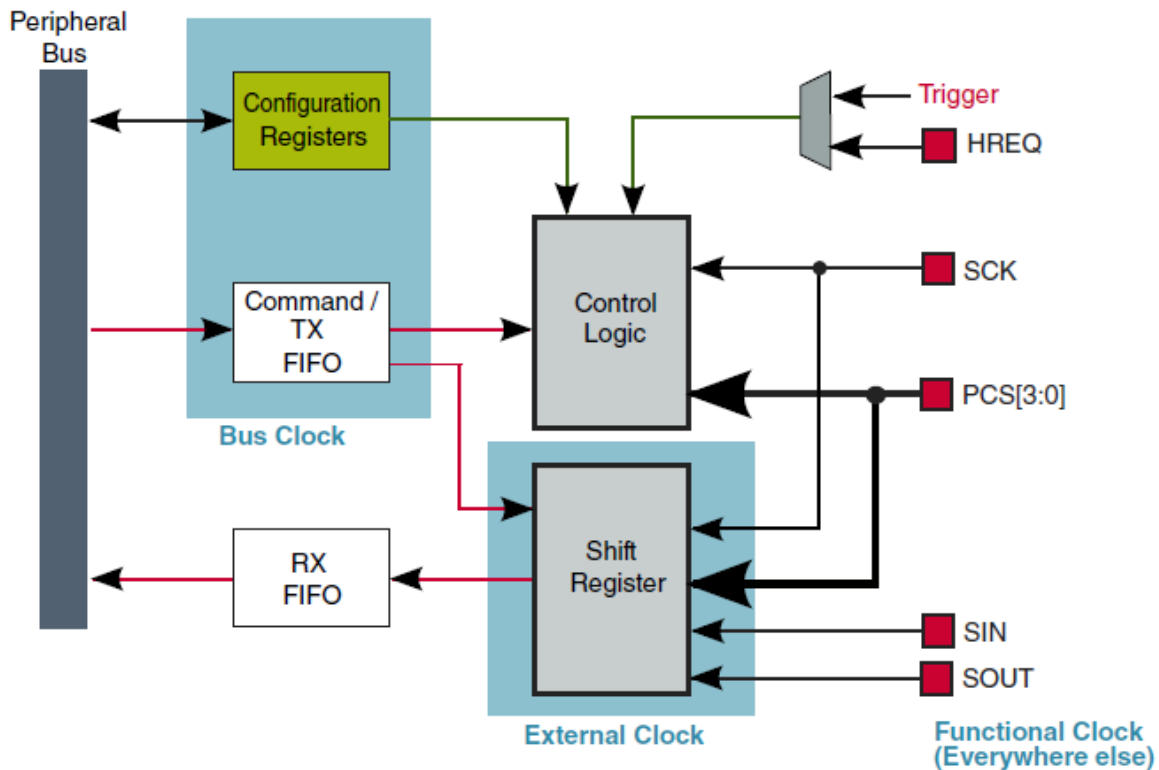


Figure 3. LPSPI block diagram

### 4.2.2. Feature list

LPSPI key feature includes:

- FIFO of 4 words
- SOUT and SIN can be switched
- Support 1, 2, or 4 wires mode
- Direct frame size up to 512 bytes
- Work with DMA in low power mode to keep communication

### 4.2.3. Highlighted feature

The highlighted feature is, LPSPI can work with DMA in lower mode. VLPS mode is usually used here. This means that LPSPI communication can be maintained and go into low power mode to reduce power consumption. This feature is available for both master and slave mode.

### 4.2.4. Improvement

**Table 4. LPSPI improvement**

Feature	LPSPI	SPI/DSPI	Benefit
Separated command FIFO and data FIFO	Yes	No. Both SPI command and DATA share PUSHR register	Software oriented design, easy to use with DMA. Reduce SW intervention and increase data throughput.
Operational in VLPS in slave mode	Yes	No	LPSPI can still receive data in VLPS mode. Give more flexible option of low power mode selection.
Host request input can be used to control the start time of an SPI bus transfer	Yes	No	External pin can direct trigger SPI to send/receive data, help to reduce CPU overhead.

For more information about LPSPI, see [AN5320: Using LPSPI on KL28](#).



## 4.3. LPI2C

### 4.3.1. Block diagram

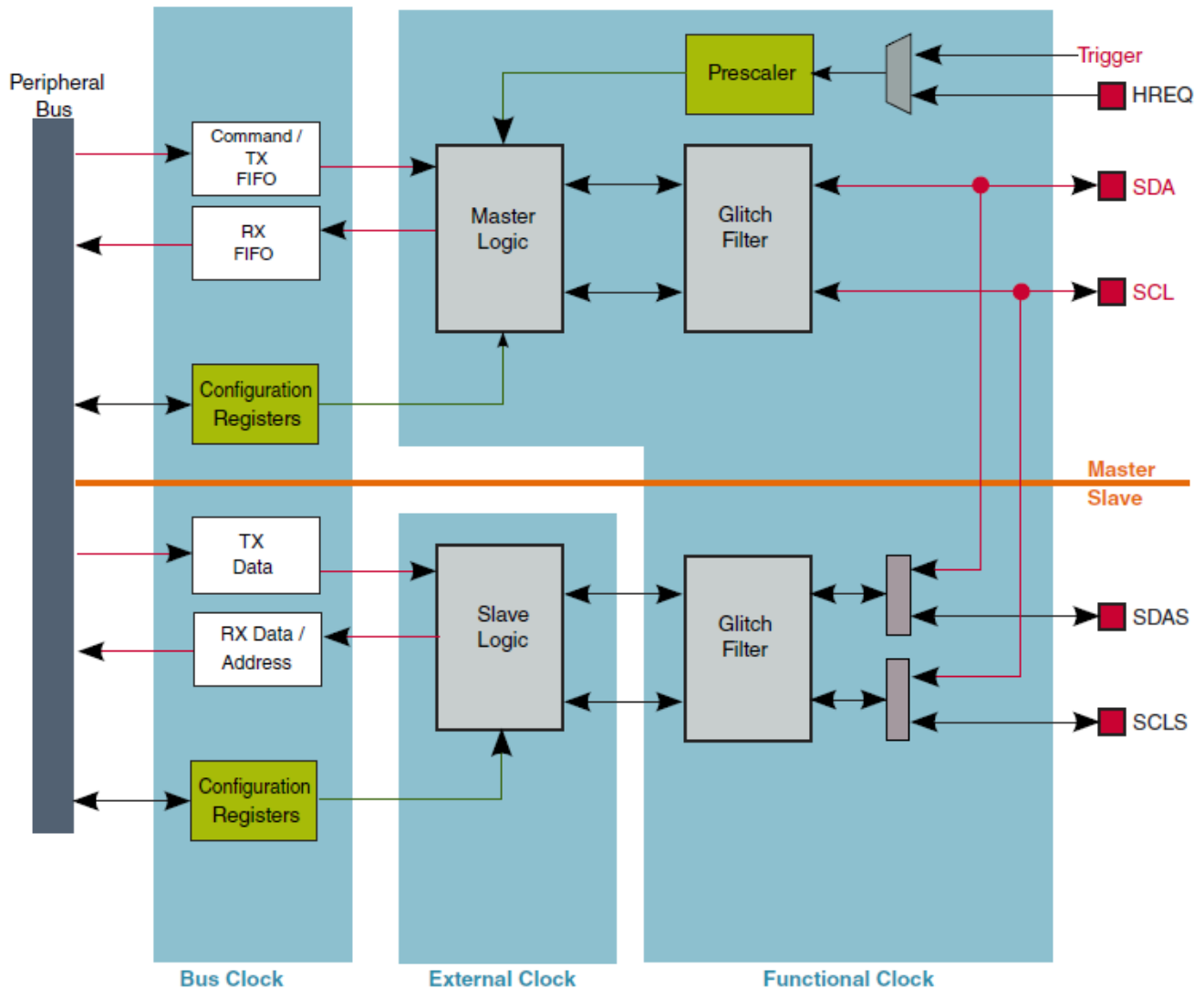


Figure 4. LPI2C Block diagram

### 4.3.2. Feature list

#### 4.3.2.1. General features

- Standard, Fast, Fast+ and Ultra-Fast modes are supported.
- HS-mode supported in slave mode.
- HS-mode supported for master mode, provided SCL pin implements current source pull-up

(device specific).

- Multi-master support including synchronization and arbitration.
- Clock stretching.
- General call, 7-bit and 10-bit addressing.
- Software reset, START byte and Device ID require software support.

#### 4.3.2.2. Master features

- Command/transmit FIFO of 4 words.
- Receive FIFO of 4 words.
- Command FIFO will wait for idle I2C bus before initiating transfer
- Command FIFO can initiate (repeated) START and STOP conditions and one or more master-receiver transfers.
- STOP condition can be generated from command FIFO or automatically when the transmit FIFO is empty.
- Host request input can be used to control the start time of an I2C bus transfer.
- Flexible receive data match can generate interrupt on data match and/or discard unwanted data.
- Flag and optional interrupt to signal Repeated START condition, STOP condition, loss of arbitration, unexpected NACK and command word errors.
- Supports configurable bus idle timeout and pin stuck low timeout.

#### 4.3.2.3. Slave features

- Separate I2C slave registers to minimize software overhead due to master/slave switching.
- Support for 7-bit or 10-bit addressing, address range, SMBus alert and general call address.
- Transmit data register supporting interrupt or DMA requests.
- Receive data register supporting interrupt or DMA requests.
- Software controllable ACK or NACK, with optional clock stretching on ACK/NACK bit.
- Configurable clock stretching to avoid transmit FIFO underrun and receive FIFO overrun.
- Flag and optional interrupt at end of packet, STOP condition or bit error detection.

#### 4.3.3. Highlighted feature

The highlighted feature is, LPI2C can work with DMA in lower mode. VLPS mode is usually used here. This means that I2C communication can be maintained and go into low power mode to reduce power consumption. This feature is available for both master and slave mode.

## 4.3.4. Improvement

**Table 5. LPI2C improvement**

Feature	LPI2C	I2C	Benefit
Function as both master and slave in the same time	Yes	No	Support I2C bridge to transfer data. Enable the fastest switch from master to slave w/ low SW intervention.
Configurable pin mode (4-wire, 2-wire or separately 2-wire for master and slave, as well as drive type)	Yes, there are 8 selectable pin type to get different functions	No, just support 2-wire	Flexible to interface with external custom line drivers Provide higher bus drive capability Reduced total system BOM
Support standard, Fast, Fast+, Ultra-Fast and Hs-mode	Yes	Just support standard and fast mode	Support all I2C bus protocol logics for different applications with baud rate up to 5M bps
Both transmit and receive FIFO w/DMA supported	Yes, support 4 words command/transmit and 4 words FIFO Receive FIFO	No, just some parts support dual buffer, and no command FIFO supported	Command FIFO based state machine provides the best performance with low CPU overhead
Unified peripheral clock control register names	Yes	No	Software oriented design makes code compatible
Software reset master or slave	Yes	No	Conveniently reset master or slave when bus lockup occurs
Configurable host trigger source	Yes, available to configure host request select from LPI2C_HREQ or input trigger	No	Automatic trigger I2C master to initiate a START condition when bus is idle, reduces SW handshake overhead
Configurable timing parameters of SCL and SDA	Support dedicated register to configure setup host time and valid delay of data, as well as high period and low period of clock	determined by I2C baud rate setting,	More flexible to configure I2C timing
Separately address matching flag	Support separately address match flag for address0/1 and general call match event	Just one address matching flag to indicate address matching event occur	Reduce the software intervention to differ matching address in multi-master system

New module introduction

<p>More status indication and interrupt control</p> <ul style="list-style-type: none"> <li>- Bit error detection by slave</li> <li>- End of packet detection by master</li> </ul>	<p>Support stop, start, FIFO, address match and bit error status indication and other status flag, support interrupt control for these flags</p>	<ul style="list-style-type: none"> <li>- Combine some status to one interrupt flag(CCIF),</li> <li>- Can't control stop and start interrupt separately,</li> <li>- No bit error detection</li> </ul>	<p>Able to get more status information and flexible to enable expected interrupt, avoid to generate unwanted interrupt.</p> <p>Able to do fault detection and recover in noise environment.</p> <p>Reduce SW overhead and increase the bus throughput.</p>
<p>Flexible master receive data match can generate interrupt on data match or discard unwanted data</p>	<p>Yes</p>	<p>No</p>	<p>Reduces SW overhead to do pattern search</p>
<p>Automatic STOP generation</p>	<p>Yes</p>	<p>No</p>	<p>Reduce SW overhead for STOP generation</p>

For more information about LPI2C, see [AN5301: Using LPI2C on KL28](#).

## 4.4. FLEX IO with enhanced feature

### 4.4.1. Block diagram

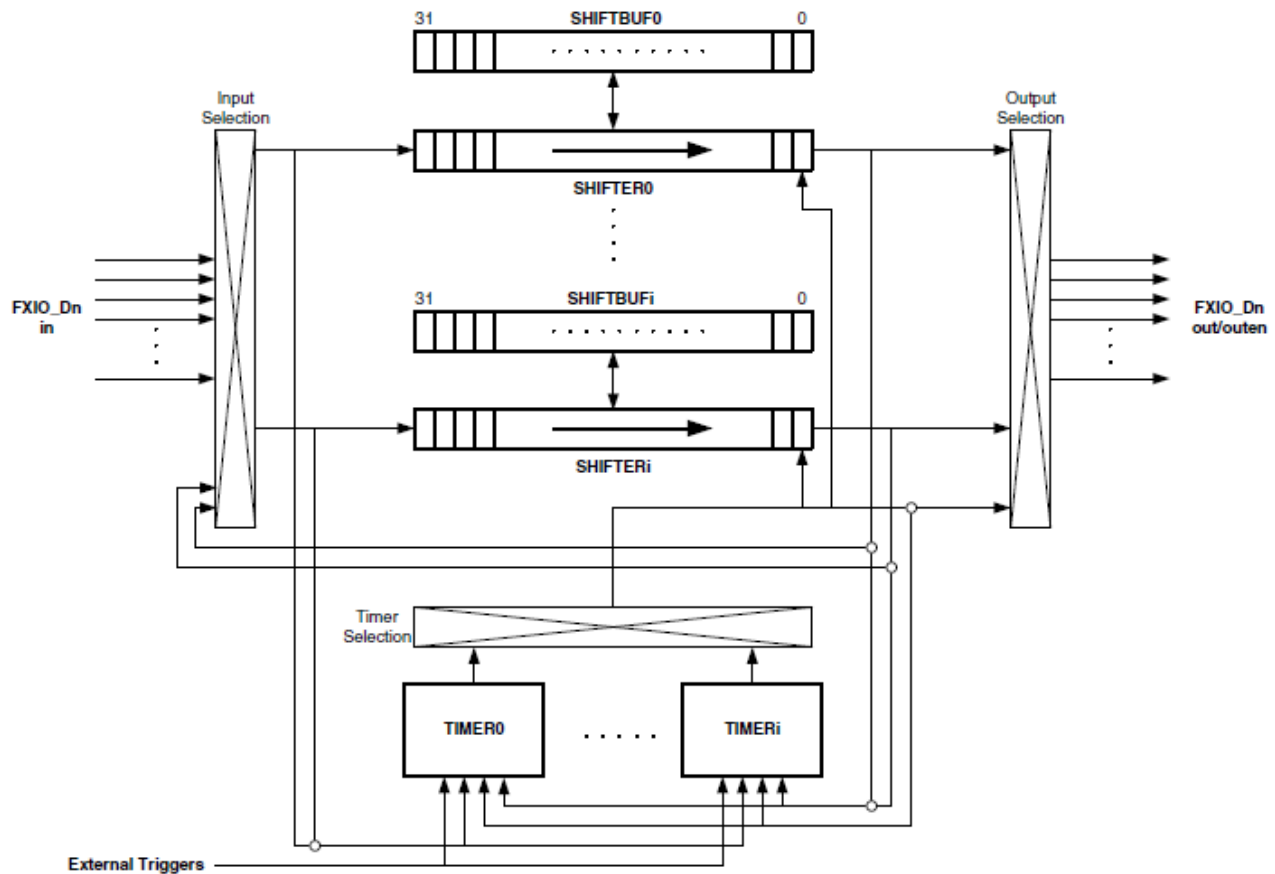


Figure 5. FlexIO block diagram

### 4.4.2. Feature list

FLEX IO feature includes:

- Configurable peripheral:
  - UART
  - I2C
  - SPI
  - I2S
  - PWM/Waveform generator
  - Camera interface
  - 68K/Intel8080 bus
- Low software/CPU loading

- Double buffered shifter operation for continuous data transfer
- Programmable baud rates
- Functional in Stop/Wait/VLPS/LLS/VLLS3 mode
- DMA support

#### 4.4.3. Enhanced features on KL28

For KL28Z, the enhanced features include:

- 8 Shifters
- 8 Timers
- Up to 32 Pins
- Up to 32 External Triggers

#### 4.4.4. Improvement

For the improvement, please refer to the following table:

**Table 6. FLEXIO improvement**

Feature	KL28Z	Old FlexIO (KL43/KL33/ KL27/KL17)	Benefit
Parallel data transfer	Yes	No	Allows emulation of camera interface, LCD interface, 68K/Intel8080 bus, etc.
State Machine	Yes	No	Allows basic system control to be offloading from CPU
Digital logic	Yes	No	SHIFTBUF register can be used to implement a 5-input, 32-bit programmable logic look-up table. Allows digital logic to be integrated on-chip.
Serial data transfer	Yes	Yes	Allows easy emulation of serial interface protocols such as UART, I2C, SPI, I2S, etc.
Generic timer function	Yes	Yes	Can be used as additional timers to generate clock or select output or PWM waveforms
Flexible timer sync	Yes	Yes	Allows multiple timers to be enabled/disabled/started/stopped/reset In synchronization with a trigger, easy to sync multiple PWMs used in motor control and power supply applications.
Flexible pin selection, polarity control, and direction control	Yes	Yes	No external logic needed for such function, reduce BOM cost

For more information about FLEX IO on KL28, see [AN5313: Using FlexIO to Drive 8080 Bus Interface LCD Module](#).

## 4.5. TRGMUX for peripheral triggering

The TRGMUX introduces an extremely flexible way for connecting various trigger sources to multiple pins/peripherals. It is a replacement to SIM module in other KL part.

### 4.5.1. Block diagram

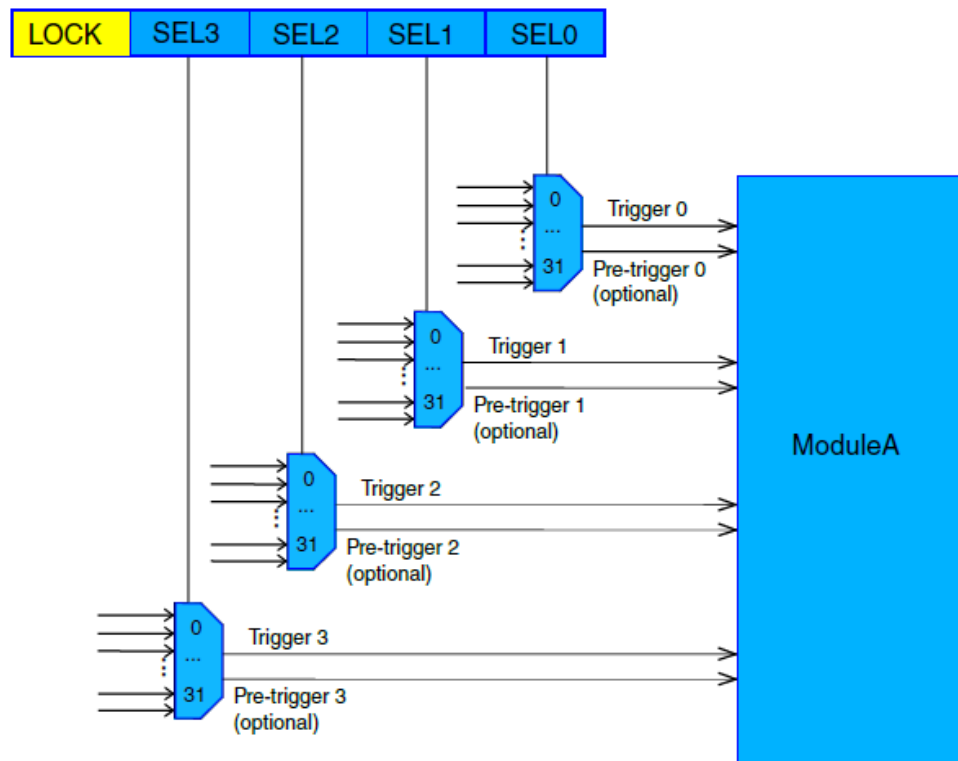


Figure 6. TRGMUX structure

### 4.5.2. Feature list

- Allows software to configure the trigger inputs for various peripherals.
- One 32-bit register per peripheral
- Software lock bit
- 1 to 4 Trigger mux per peripheral
- Each mux supports 32/64/128 inputs (parameterized)
- Inputs are common for every TRGMUX instance

For more information about TRGMUX, see [AN5399: Using TRGMUX on KL28 Based on SDK2.0](#).

## 4.6. True random number generator

### 4.6.1. Block diagram

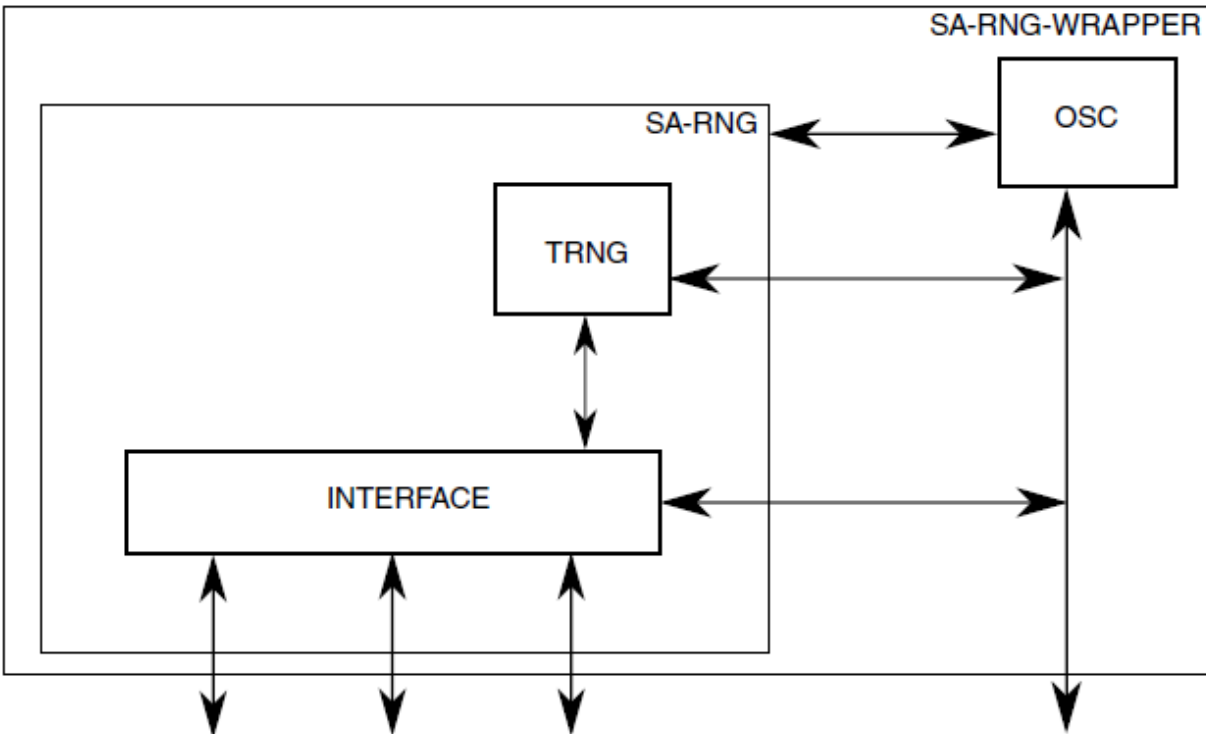


Figure 7. True random number generator block diagram

### 4.6.2. Feature list

- 512-bit entropy
- Based on collecting bits from a random noise source (temperature variations, voltage variations, cross-talk and other random noise)

## 4.7. Time stamp timer

### 4.7.1. Feature list

- Global 56-bit counter for software time stamping
- Resets on POR/LVD/VLLS
- Increments at 1 MHz clock whenever IRC8M is enabled
- Readable by software (reading lower 32-bits capture upper 24-bits)
- Memory mapped to SIM (and RPM\_SIM for Core1)
- Counter bus is output in gray-code format, usable by other modules

Migrating from other KL serial parts to KL28, Application Note, Rev. 0, 01/2017



## 4.8. Smart card interface module

### 4.8.1. Block diagram

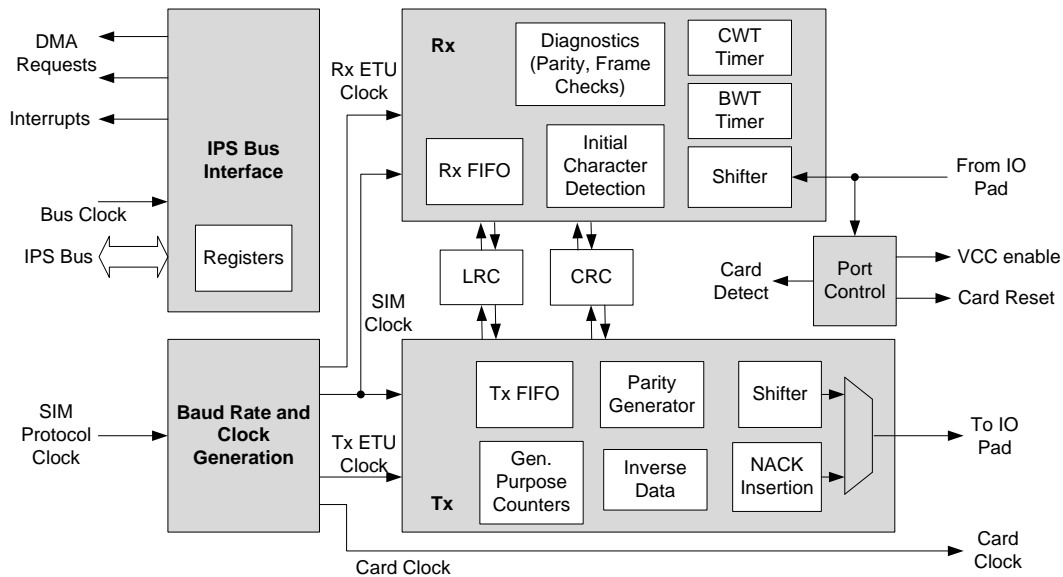


Figure 8. Smart card interface module block diagram

### 4.8.2. Feature list

- Supports SIM cards based on the EMV Standard v4.3 and ISO 7816-3 standard
- Independent clock for SIM logic (transmitter + receiver) and independent clock for register read-write interface
- 16 byte deep FIFO for transmitter and receiver
- Automatic NACK generation on parity error and receiver FIFO overflow error
- Support for both Inverse and Direction conventions
- Re-transmission of byte upon SIM card NACK request with programmable threshold of re-transmissions
- Auto detection of Initial Character in receiver and setting of data format (inverse or direct)
- NACK detection in receiver
- Independent timers to measure character wait time, block wait time and block guard time
- Two general purpose counters available for use by software application with programmable clock selection for the counters
- DMA support available to transfer data to/from FIFOs. Programmable option available to select interrupt or DMA feature
- Programmable Prescaler to generate the desired frequency for SIM card clock and Baud Rate Divisor to generate the internal ETU clocks for transmitter and receiver for any F/D ratio
- Internal oversampling by 16x in receiver

- Deep sleep wake-up via SIM card presence detect interrupt
- Manual control of all SIM card interface signals
- Automatic power down of port logic on SIM card presence detect
- Support for 8-bit LRC and 16-bit CRC generation for transmitter and checking for receiver

## 4.9. Low power periodic interrupt timer

### 4.9.1. Block diagram

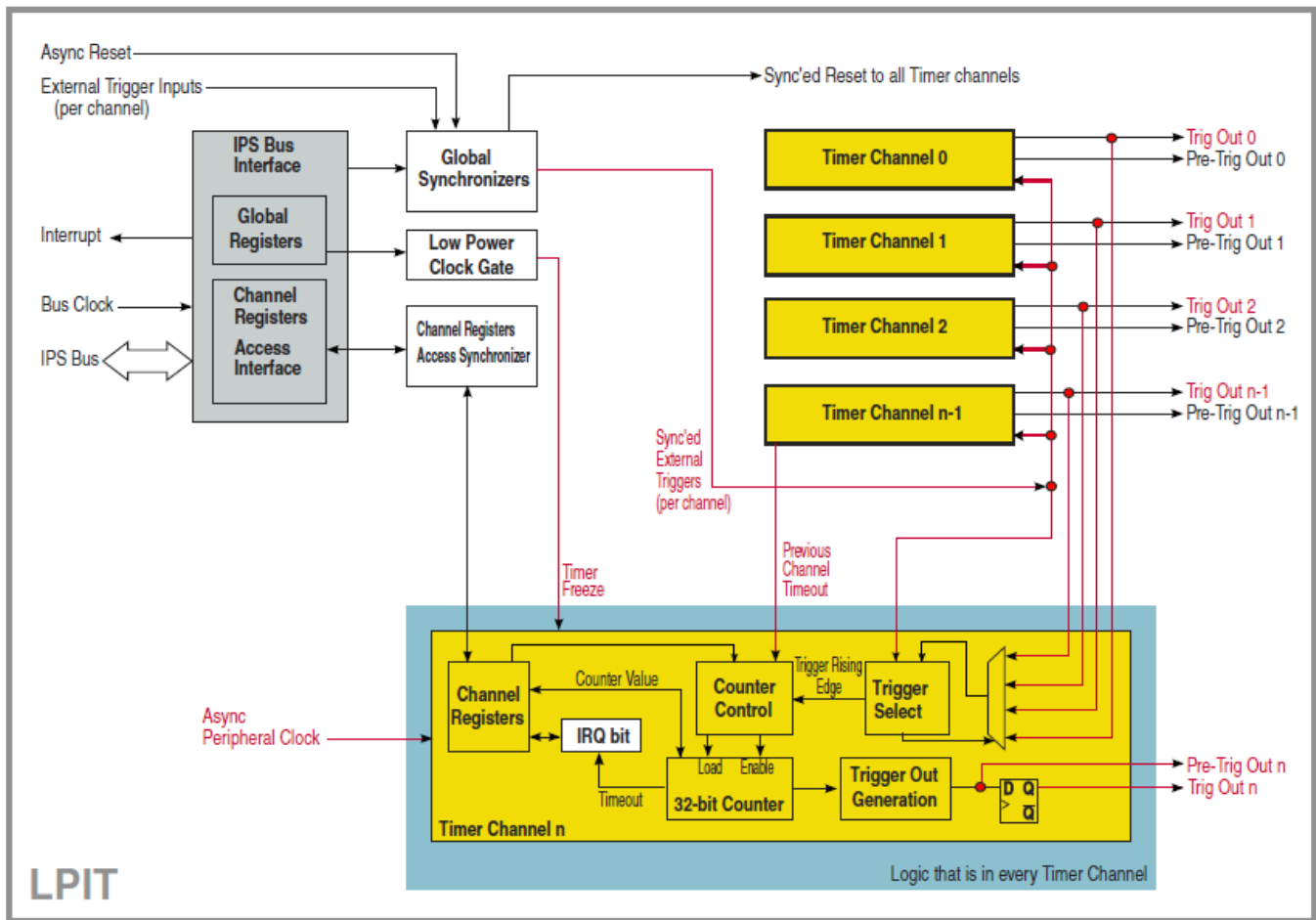


Figure 9. LPIT block diagram

### 4.9.2. Feature list

- Four 32-bit counters, functional in Stop modes
- Supports trigger generation (single or pulse width) for analog components (ADC, DAC, CMP)
- Supports trigger input capture and trigger counting

### 4.9.3. Improvement

Table 7. LPIT improvement

Feature	LPIT	PIT	Benefit
Different clock source	Configured in PCC by software	Bus clock	LPIT can work in low power mode

## 4.10. Memory-mapped divide and square root

### 4.10.1. Block diagram

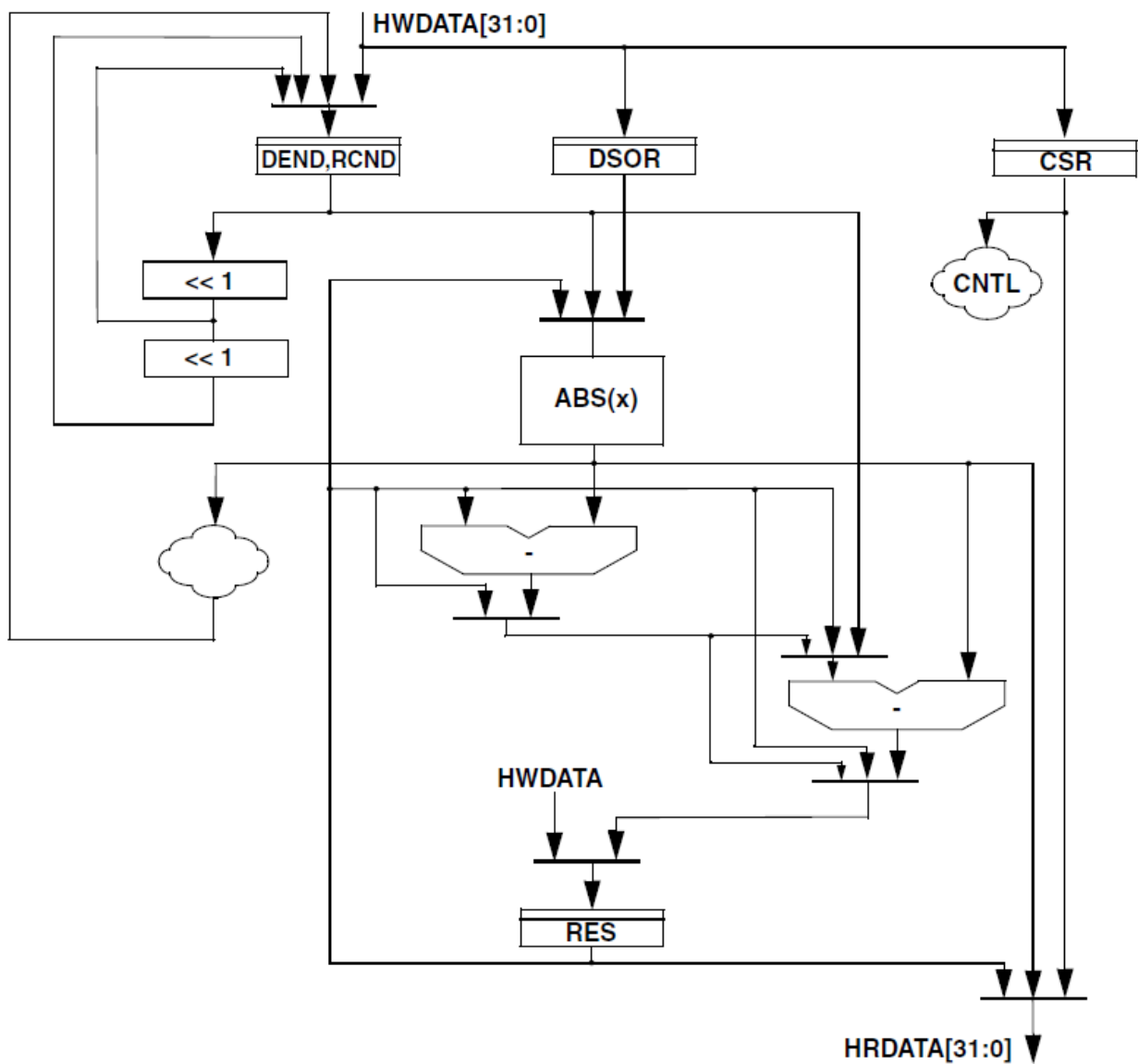


Figure 10. Memory-mapped divide and square root block diagram  
 Migrating from other KL serial parts to KL28, Application Note, Rev. 0, 01/2017

## 4.10.2. Feature list

- Lightweight implementation of 32-bit integer divide and square root arithmetic operations
  - Supports 32/32 signed and unsigned divide (or remainder) calculations
  - Supports 32-bit unsigned square root calculations
- Simple programming model includes input data and result registers plus a control/status register
- Programming model interface optimized for activation from inline code or software library call
- "Fast Start" configuration minimizes the memory-mapped register write overhead
- Supports two methods to determine when result is valid, including software polling
- Configurable divide-by-zero response
- Pipelined design processes 2 bits per cycle with early termination exit for minimum execution time

## 4.10.3. Improvement

**Table 8. MMDVSQ improvement**

Test items	SW implementation	MMDVSQ on SDK 2.0	MMDVSQ by optimized code based SDK2.0
sqrt(0x7000ffee)	13.80 us	0.91us	0.75 us
0x7000ffee/ 31415	27.50 us	1.14 us	0.65 us

The result above is got when core clock is 48MHz CPU clock.

## 5. Large memory configuration

KL28 has up to 512 KB Flash and 128 KB SRAM, which facilitate complex application which need a lot of Flash and RAM.

### 5.1. Improvement

**Table 9. Memory configuration improvement**

Items	KL28	KL36	KL46	KL8x
Maximum FLASH	512	256	256	128
Maximum RAM	128	32	32	96

## 6. Revision history

Table 10. Revision history

Revision number	Date	Substantive changes
0	01/2017	Initial release

---

**How to Reach Us:**

**Home Page:**

[nxp.com](http://nxp.com)

**Web Support:**

[nxp.com/support](http://nxp.com/support)

Information in this document is provided solely to enable system and software implementers to use NXP products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits based on the information in this document. NXP reserves the right to make changes without further notice to any products herein.

NXP makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does NXP assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. "Typical" parameters that may be provided in NXP data sheets and/or specifications can and do vary in different applications, and actual performance may vary over time. All operating parameters, including "typicals," must be validated for each customer application by customer's technical experts. NXP does not convey any license under its patent rights nor the rights of others. NXP sells products pursuant to standard terms and conditions of sale, which can be found at the following address: [nxp.com/SalesTermsandConditions](http://nxp.com/SalesTermsandConditions).

NXP, the NXP logo, NXP SECURE CONNECTIONS FOR A SMARTER WORLD, Freescale, the Freescale logo, and Kinetis are trademarks of NXP B.V. All other product or service names are the property of their respective owners.

ARM, the ARM logo, and Cortex are registered trademarks of ARM Limited (or its subsidiaries) in the EU and/or elsewhere. All rights reserved.

© 2017 NXP B.V.

Document Number: AN5390

Rev. 0

01/2017

