Nanjing Linko Semiconductor Co Ltd

LKS08x/LKS05x/LKS07x (07x Change List)

Table of contents

Table of contents

Picture directory

1 Overview

This document mainly compares the similarities and differences of LKS08 series, LKS05 series and LKS07 series, which is convenient for engineers who are familiar with the preorder series MCU to get started with LKS07 series quickly. In summary, LKS05 is a Cost down version of LKS08. Some of the peripheral modules in LKS08 no longer exist in LKS05, and some of the resources are more sophisticated and cost-effective. LKS07 is an upgraded version of LKS08, with increased resource specifications.

The products are listed in the order of LKS08, LKS05 and LKS07. Unless otherwise specified, the new product series will inherit the functions and features of the previous series.

2 Flash

Table STYLE REF 1 \ s **错误!未指定样式名。**SEQ \ * ARABIC \ s 1 **错误!未指定顺序。**Flash specification comparison

	MAIN area	NVR
LKS08x	32kB/64kB	1kB
LKS05x	32kB	1kB
LKS07x	64kB/128kB	1.5kB

3 SRAM

Table STYLE REF 1 \ s **错误!未指定样式名。**SEQ \ * ARABIC \ s 1 **错误!未指定顺序。**SRAM specification comparison

	SRAM capacity
LKS08x	8kB
LKS05x	2.56kB
LKS07x	12kB

4 Clock

Table STYLE REF 1 \ s **错误!未指定样式名。**SEQ \ * ARABIC \ s 1 **错误!未指定顺序。**Clock specification comparison

	HRC/MHz	XTAL/MHz	PLL/MHz
LKS08x	4	4	96
LKS05x	4	None	96
LKS07x	8	8	96

5 ADC

5.1 LKS08x

Support DMA transfer and synchronous double sampling function. The ADC0 _ GAIN0 and ADC0 _ GAIN1 registers control the gain selection for the different channels, respectively. When right-justified, the saturation limit is between 0 x8000 and 0x7FFF. The fastest conversion rate of the ADC is 3 MHz.

The ADC of LKS08x completes signal sampling synchronously for the signal of the 1/2 signal channel set immediately after the sampling is triggered.

5.2 LKS05x

Does not support DMA handling, does not support the simultaneous double sampling function, and does not support the external input power supply as the ADC REF.

Hardware trigger error interrupt enable bit and interrupt flag bit are added, which are ADC _ IE [5] and ADC _ IF [5] respectively.

The total number of ADC channels is reduced from 20 to 16, and the channel select signal is reduced to 4-bit width. Channel select register changed. For example:

ADC _ CHN0 Register

Location	Explain
15:12	ADC 3rd Sample Signal Select
11:08	ADC 2nd Sample Signal Select
07:04	ADC 1st Sample Signal Select
03:00	ADC zeroth sample signal selection

The ADC gain select register is controlled by the SYS _ AFE _ REG0 [6]. The default value is 2/3 times, corresponding to the input signal range of +/-3.6 V, which is consistent with LKS08x. AMC default is 0x200 For right alignment, the saturation limit value range is 0 xF800 to 0 x07FF; for left alignment, the value range is 0x8000 to 0 x7FF0, and the lower 4 bits are always 0. That is, whether left or right justified, the value is limited to a 12-bit representation.

The LKS05x operates at a maximum ADC clock frequency of 48 MHz, and the recommended SYS _ AFE _ REG7 [13:8] (SAMP _ TIME) is 0x08, which corresponds to a maximum ADC conversion rate of 2 MHz.

The LKS05x ADCs do not have simultaneous double sampling. After the ADC of LKS05x is triggered, the sampling of the first signal channel needs to wait for the set sampling time (set by the SAMP _ TIME register in the SYS _ AFE _ REG7), and the time node for completing the sampling is after the end of the time. The second signal channel does not start sampling until the first signal channel has finished sampling and converting. The sampling is also completed after the end of the sampling time. And so on for subsequent sampled signal channels.

5.3 LKS07x

DMA handling is supported.

The number of ADC modules is increased from one to two, that is, simultaneous double sampling is possible. Single-segment and two-segment sampling triggers are supported on each ADC module, and 4-segment sampling triggers are no longer supported on a single ADC module. Using two separate ADC modules for 2-segment sampling triggers is more flexible.

14 analog IO input signals/4 OPA outputs/2 DAC outputs/temperature sensor output/analog ground can be used as the sampled signal of ADC. The regular sampling channels are set up using ADCx _ CHN0/1/2/3.

The trigger source is more abundant, and can be linked with the MCPWM/Timer/CL unit for sampling trigger, wherein the CLU module is similar to the CPLD in the chip, and the key signals of multiple modules of the chip are interconnected and intercommunicated, so that the module linkage is facilitated. Configure using ADCx _ TRIG0/1.

The ADC DC offset automatically adjusts in hardware according to the setting of the left and right alignment of the ADC _ CFG. DATA _ ALIGN. No software intervention is required.

Add idle sampling function: ADC trigger is divided into two types: Normal Trigger (NT) and Idle Trigger (IT). A regular trigger has a higher priority than an idle trigger. If an idle trigger occurs while the ADC is in the middle of a regular sample conversion, the ADC continues to complete the normally triggered channel sample conversion before proceeding to the idle trigger sample conversion; If the ADC is in the middle of an idle sample conversion and a regular trigger occurs, the regular trigger interrupts the idle sample conversion and waits for the regular trigger sample to complete before the idle trigger occurs. The idle trigger is initiated using a software trigger. ADCx _ SWT = 0 xF00F initiates an idle trigger and ADCx _ SWT = 0x5AA5 initiates a software normal trigger. Idle trigger supports up to two channel settings, which are set through the ADCx _ ICHN.

An analog watchdog is added to simultaneously detect the upper and lower thresholds of the ADC sampling signal.

Oversampling is supported to achieve a higher signal-to-noise ratio by averaging multiple samples of data. Set using the ADC _ CFG.

The supports the use of an external supply as the ADC reference, which is configured using the SYS _ AFE _ REG2.REF2VDD.

6 OPA

6.1 LKS08x

Quad OPA (0/1/2/3) without multiplexing.

The SYS _ AFE _ REG2 [2:0] is the OPA output configuration:
000: No output
001: OPA0 output to P2.7
010: OPA1 output to P2.7
011: OPA2 output to P2.7
100: OPA3 output to P2.7
101 ~ 111: Illegal configuration

P3.5 & P3.7 corresponding to OPA0 inputs OPA0 _ IP and OPA0 _ IN
P3.0 & P3.1 correspond to OPA1 inputs OPA2 _ IP and OPA2 _ IN
P3.10 & P3.11 corresponding to OPA2 inputs OPA2 _ IP and OPA2 _ IN
P3.15 & P3.14 corresponding to OPA3 inputs OPA3 _ IP and OPA3 _ IN

On-chip feedback resistor ratio:
00: 200k:10.4k
01: 190k:20.4k
10: 180k:30.4k
11: 170k:40.4k

6.2 LKS05x

Two channels of OPA (0/1), supporting four channels of OPA input (not standard, see Datasheet for details), time division multiplexing. OPA0 corresponds to OPA input signal channel 0 and signal channel 2, and OPA1 corresponds to OPA input signal channel 1 and signal channel 3.

P1.14 & P1.15 Corresponding to input signal channel 0 of OPA0 module: OPA0 _ IP and OPA0 _ IN
P1.1 & P1.2 correspond to input signal channel 2 of OPA0 module: OPA2 _ IP and OPA2 _ IN
P0.9 & P0.10 correspond to input signal channel 1 of OPA1 module: OPA1 _ IP and OPA1 _ IN
P2.14 & P2.15 correspond to input signal channel 3 of OPA1 module: OPA3 _ IP and OPA3 _ IN

OPA has two application cases, one is without time division multiplexing, OPA0 only corresponds to signal channel 0, and OPA1 only corresponds to signal channel 1. The other is multiplexing mode, where OPA0 corresponds to signal channel 0 and signal channel 2, and OPA1 corresponds to signal channel 1 and signal channel 3.

When the OPA is multiplexed, the switching of the OPA input channel needs to insert the stable waiting time. ADC sampling OPA, SYS _ AFE _ REG7 [13:8] (SAMP _ TIME), needs to be configured as 0x20 (can be larger), corresponding to 36 ADC clock cycles, ADC output rate is 1MHz.

When the OPA is not multiplexed, the SYS _ AFE _ REG7 [13:8] (SAMP _ TIME) can still be configured as 0x08, corresponding to 12 ADC clock cycles, and the ADC output rate is 2MHz.

If it is a chip model with OPA multiplexing function enabled, when ADC _ CH8 is set for sampling, the output of OPA2 _ IP/IN signal after being amplified by OPA0 is actually sampled internally; when ADC _ CH9 is set for sampling, the output of OPA3 _ IP/IN signal after being amplified by OPA1 is actually sampled. The ADC _ CH8 and ADC _ CH9 functions on the GPIO pins are not active at this time.

If the OPA multiplexing function is not enabled, when ADC _ CH8 is set for sampling, the actual internal sampling is still ADC _ CH8; when ADC _ CH9 is set for sampling, the actual internal sampling is ADC _ CH9.

The SYS _ AFE _ REG0 [5:4] is the OPA output configuration:
00: No output
01: OPA0 output to P2.7
10: OPA1 output to P2.7
11: Illegal configuration

On-chip feedback resistor ratio:
00: 200k:10.6k
01: 190k:20.6k
10: 180k:30.6k
11: 170k:40.6k

6.3 LKS07x

Same specifications as LKS08x, quad OPA (0/1/2/3), no multiplexing.
On-chip feedback resistor ratio:
00: 320k:10k
01: 160k:10k
10: 80k:10k
11: 40k:10k

In addition, the change of the output common-mode voltage of the 07x operational amplifier with temperature is smaller than that of the 08x/05x operational amplifier, and the temperature correction of the common-mode voltage can not be performed.

7 DAC

7.1 LKS08x

One 12 bit DAC. Three ranges of 1.2 V, 3.6 V, and 4.8 V. The toggle control bits are SYS _ AFE _ REG1 [7:6]

7.2 LKS05x

One 12 bit DAC. Two ranges, 1.2 V and 4.8 V. The toggle control bit is the SYS _ AFE _ REG3 [15].

7.3 LKS07x

Two 12-bit DACs. Two ranges, 1.2 V and 4.85 V. The DAC supports the use of TIMER for trigger increment or decrement, for ramp generation, and SYS _ AFE _ DAC _ CTRL for control. Can be used in digital power applications.

8 GPIO

8.1 LKS08x

Four sets of GPIOs with GPIO lock protection; no internal pull-up for I2C counterpart GPIOs; no bit operation register GPIOx _ BSRR and GPIOx _ BRR; no filter register GPIOx _ PFLT.

8.2 LKS05x

Three sets of GPIOs, no GPIO lock protection, internal pull-up for I2C corresponding GPIOs, bit operation register GPIOx _ BSRR and GPIOx _ BRR, and filter register GPIOx _ PFLT. Some GPIOs add filtering using a 32kHz LRC clock on the chip. The filtering time is 4 LRC clock cycles, which can be turned on or off using the GPIOx _ PFLT [15:0]. It is off by default! It is important to note that the GPIO wake-up and interrupt signals are not filtered.

8.3 LKS07x

Four sets of GPIOs, no GPIO lock protection, internal pull-up for I2C corresponding GPIOs, bit operation register GPIOx _ BSRR and GPIOx _ BRR, and filter register GPIOx _ PFLT.

The external wake-up is increased to 8 IOs and the external interrupt pins are increased to 16.

9 SWD multiplexing

9.1 LKS08x

The multiplexing of SWD IO in LKS08x series is realized at the package level, that is, SWCLK/SWDAT and other IO with the same pin (such as P0.0) are different IO in the chip, but they are packaged to the same pin. GPIO operation still requires operation P0.0. And care should be taken that that operation of the GPIO doe not cause the SWD to malfunction.

Simultaneous multiplexing of SWCLK and SWDAT as GPIOs is not recommended.

9.2 LKS05x

IO multiplexing of LKS05x is implemented at the chip level, that is, SWCLK and P2.13 are the same IO, and SWDAT and P2.0 are the same IO.

Multiplexing to GPIO or SWD is controlled by the SYS _ RST _ CFG [6].

SWCLK (P2.13) and SWDAT (P2.0) can be multiplexed as GPIO at the same time to prevent locking. SWD is forced within 30 ms after power-on, that is, the software is configured as GPIO, which also takes effect 30 ms after power-on (if 30 ms is not safe enough, it is recommended to add additional waiting time. It is recommended to use the original offline downloader for erasure)

9.3 LKS07x

IO multiplexing of LKS07x is implemented at the chip level, that is, SWCLK and P2.14 are the same IO, and SWDAT and P2.15 are the same IO.

The SYS _ IO _ CFG. SWDMUX controls whether the multiplexing is GPIO or SWD. The default is 0, which is the SWD function.

SWCLK (P2.14) and SWDAT (P2.15) can be multiplexed as GPIO at the same time to prevent locking. SWD is forced within 30 ms after power-on, that is, the software is configured as GPIO, which also takes effect 30 ms after power-on (if 30 ms is not safe enough, it is recommended to add additional waiting time. It is recommended to use the original offline downloader for erasure)

10 DMA

10.1 LKS08x

There is a DMA module

As shown in the figure below, DMA transfers are supported only for peripherals whose ports have an Arbiter. DMA transfer of flash is not supported.

Single DMA engine, 4 channels.

Multiple rounds × multiple transmissions are supported.

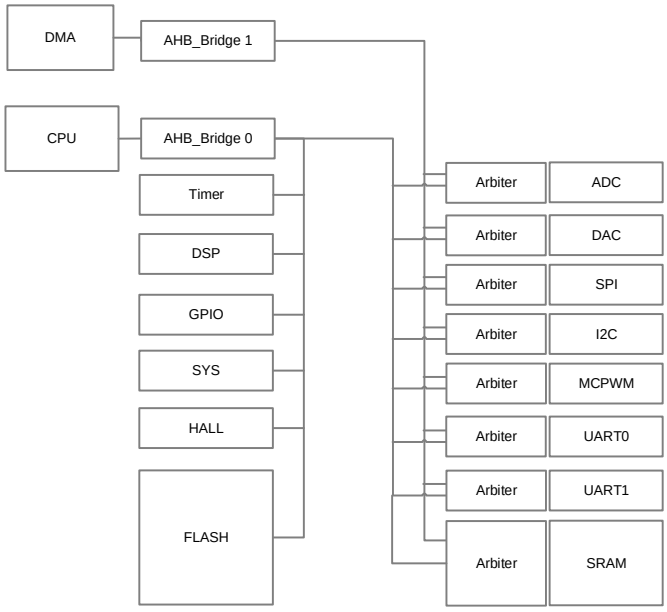

Figure STYLE REF 1 \ s **错误!未指定样式名。**SEQ Figure \ * ARABIC \ s 1 **错误!未指定顺序。**LKS08x DMA Bus Architecture

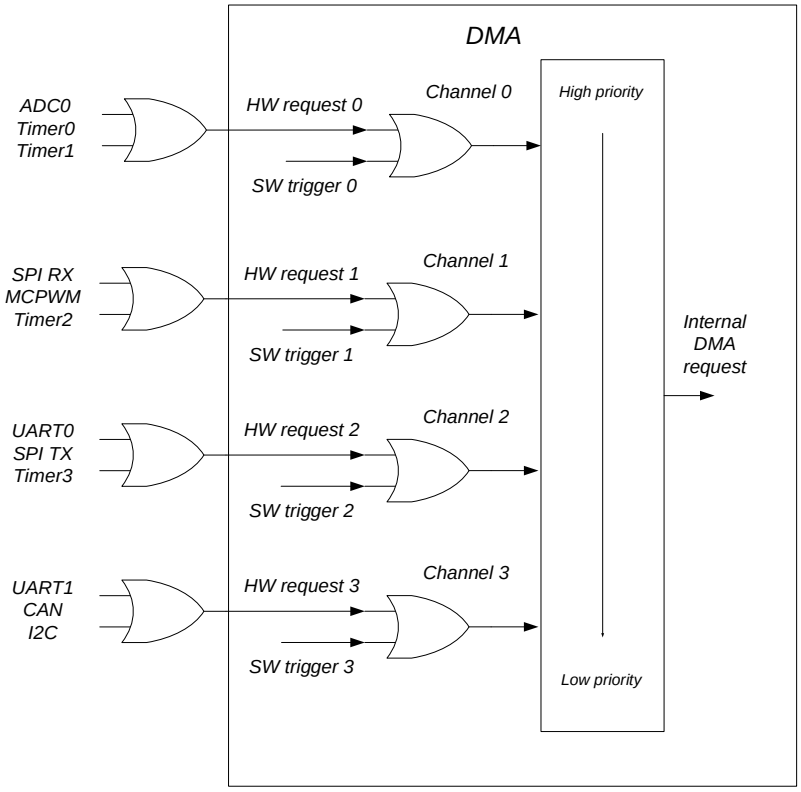

Figure STYLE REF 1 \ s **错误!未指定样式名。**SEQ Figure \ * ARABIC \ s 1 **错误!未指定顺序。**LKS08x Peripheral DMA Request

10.2 LKS05x

No DMA module

10.3 LKS07x

There is a DMA module, and most of the peripherals can be accessed by DMA. Most device events can be the source of a DMA request. Typically, compared with LKS08x, TX/RX requests of all serial interfaces UART/I2C/SPI/CAN are split into different DMA requests, such as TX/RX of UART0/1, so that two DMA channels can be used to receive and transmit UART at the same time.

Multi-round × multiple transmission is not supported, but multi-round transmission is supported, and data is transmitted once in each round, such as UART receiving and transmitting; or data is transmitted multiple times in one round, such as continuous transmission of data from some peripherals to memory.

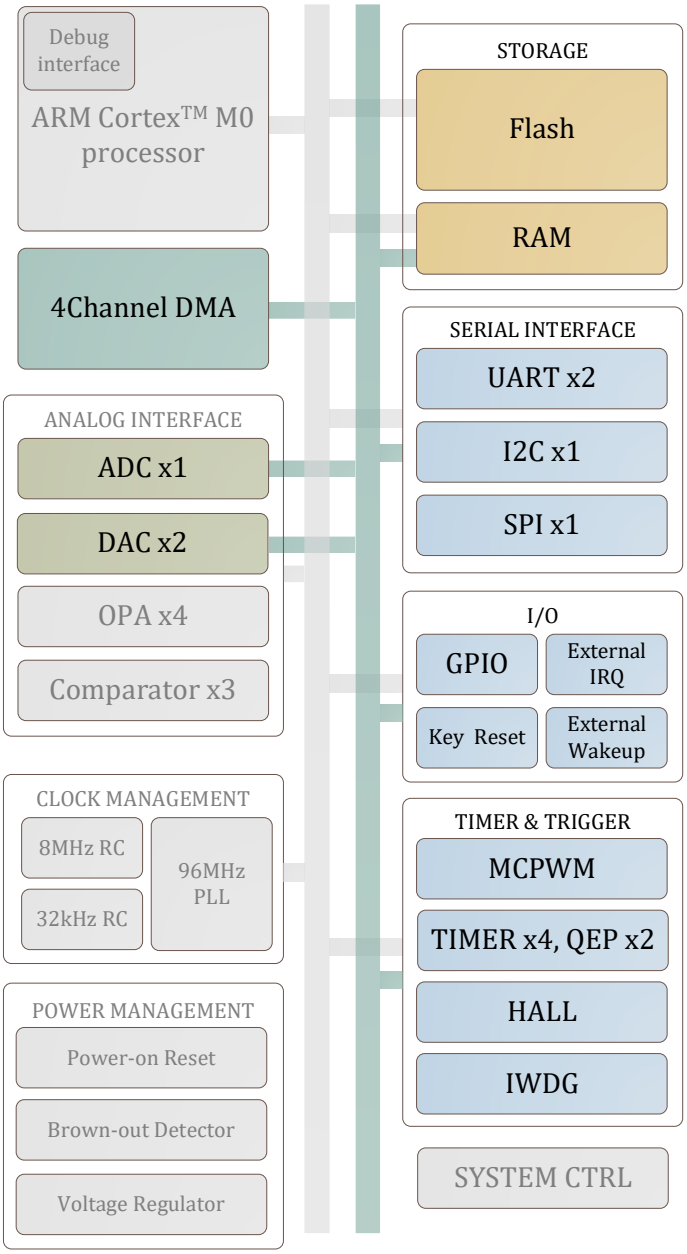

Figure STYLE REF 1 \ s **错误!未指定样式名。**SEQ Figure \ * ARABIC \ s 1 **错误!未指定顺序。**LKS07x DMA Bus Architecture

11 UART

11.1 LKS08x

UART has DMA mode

Support 7/8 bit length data

11.2 LKS05x

UART No DMA Mode

Support 7/8 bit length data

Multi-drop Slave/Master mode supporting one master and multiple slaves

11.3 LKS07x

UART has DMA mode

Support 8/9 bit length data

Support LIN mode break character transceiving

Idle frame detection is supported

Better support the multi-drop Slave/Master mode with one master and multiple slaves

12 MCPWM

12.1 LKS08x

4 pairs of PWM channels.

Read and write access is made to the address where the shadow register exists, the write value is written to the preset value register, and the read back value is the value of the preset register.

The MCPWM _ CNT does not have a shadow register, and the software writes an immediate update.

It should be noted that the counting range of-MCPWM _ TH ~ MCPWM _ TH is actually shifted inside the MCPWM, and the actual counting range of MCPWM _ CNT is 0x8000-MCPWM _ TH ~ 0x8000 + MCPWM _ TH. Therefore, the range of the MCPWM _ CNT read by the software is 0x8000-MCPWM _ TH-0x8000 + MCPWM _ TH, which shall be written in the following way. After writing the MCPWM _ CNT, execute a dummy write operation to complete the update:

```
MCPWM_CNT = 0x8000+SET_VALUE;
MCPWM_PRT = 0xDEAD;
```

T0 of LKS32MC08x is MCPWM _ CNT = = 0x8000-MCPWM _ TH.

The PWM channel level is changed only when the MCPWM _ CNT = = MCPWM _ THxx. If the software changes the MCPWM _ CNT from < MCPWM _ THxx directly to > MCPWM _ THxx, the channel level does not change.

When a FAIL event occurs, all PWM channels are switched to the default level set by software.

12.2 LKS05x

4 pairs of PWM channels.

Unlike the LKS32MC08x, read and write access to the address where the shadow register exists, the write value is written to the preset value register, and the read back value is the value of the shadow register. A preset value is load into that shadow register at an appropriate time. Only when the update is completed, the two are consistent.

In the LKS32MC05x, the MCPWM _ CNT register has a corresponding shadow register, and the software write does not take effect immediately.

The MCPWM _ CNT can be updated manually by writing a 1 to the MCPWM _ UPDATE [13] or automatically using hardware. If you want to enable the automatic hardware update of the MCPWM _ CNT, you need to set MCPWM _ AUEG [13] = 1.

MCPWM _ CNT are not protected by MCPWM _ PRT.

The T0 of LKS32MC05x is modified to MCPWM _ CNT = = 0x8000 + MCPWM _ TH. The goal of that modification is that if a hardware update of the MCPWM _ TH is require at time T0, the next cycle is immediately counted use the new MCPWM _ TH as a PWM cycle. Otherwise, on the LKS32MC08x, it will appear that the PWM has started counting with the old MCPWM _ TH at T0, resulting in this PWM cycle

being counted from -THold to THnew. Through this change, the MCPWM _ TH can be updated to support the zero point update and the current cycle will take effect immediately.

A new MCPWM _ AUEN register is added to select which MCPWM registers are automatically updated. A new MCPWM _ UPDATE [13] is added to control the manual update of the MCPWM _ CNT. The MCPWM internal shadow counter can be loaded with the MCPWM _ CNT preload value by first assigning the MCPWM _ CNT preload register and then writing a 1 to the MCPWM _ UPDATE [13].

When a FAIL event occurs, all PWM channels are switched to the default level set by software.

12.3 LKS07x

6 pairs of PWM channels.

The MCPWM _ AUEN follows the MCPWM design of LKS05x.

The MCPWM time base increases from 1 to 2, the corresponding MCPWM _ CNT increases to 2 MCPWM _ CNT0/1, and the MCPWM _ TH increases to 2 MCPWM _ TH0/1. Channels 0/1/2 of the MCPWM operate on timebase 0, using MCPWM _ TH0/MCPWM _ CNT0, and channels 3/4/5 operate on timebase 1, using MCPWM-TH1/MCPWM-CNT1. The periods of the two time bases are different, the initial phase can be set independently, and a certain moment of one time base can be set to be used as a trigger for starting the other time base, so that the two time bases are orthogonal when the double motors are driven.

MCPWM _ DTH00/01 controls the deadband setting for channel 0/1/2 and MCPWM _ DTH00/01 controls the deadband setting for channel 3/4/5.

The PWM channel level changes when the MCPWM _ CNT software rewrite update is supported.

Use the MCPWM _ CH _ DEF to set the default value for 6 pairs of PWM channels, which are no longer placed in the MCPWM _FAIL register.

Use the MCPWM _ CH _ MSK to set whether one of the six pairs of 12 channels is switched to the default level when a FAIL event occurs, that is, whether the 12 channels are protected by FAIL can be set independently.

The PWM adds functions related to the digital power supply, and the related registers are MCPWM _ TCLK, MCPWM _ STT _ HYST, and MCPWM _ ZCS _ DELAY.

13 CAN

13.1 LKS08x

There is a CAN module,

13.2 LKS05x

No CAN module

13.3 LKS07x

With CAN module, CAN ID filtering is increased from 2 groups to 4 groups. The receiving buffer can receive 10 frames of data, and the sending buffer can store 2 frames.

14 SIF

Table STYLE REF 1 \ s **错误!未指定样式名。**SEQ \ * ARABIC \ s 1 **错误!未指定顺序。**Comparison of SIF resources

	Equipped or not
LKS08x	Have
LKS05x	None
LKS07x	Have

15 SPI/IIC

15.1 LKS08x

IIC has DMA mode; SPI has DMA mode, and the data transfer length unit is not configurable (fixed 8-bit)

15.2 LKS05x

No DMA mode for IIC, no DMA mode for SPI, configurable data transfer length unit (8-Bit to 16-Bit)

15.3 LKS07x

IIC has DMA mode; SPI has DMA mode, and data transfer length unit is configurable (8-Bit to 16-Bit)

16 CMP

16.1 LKS08x

2-way comparator.

16.2 LKS05x

2-way comparator.

The path from comparator output to timer is added. 4 pairs of timers can select gpio as input, or select the output of comparator 0/1 as timer input for capture. The capture filtering of the timer is also effective for the signal of the comparator and is optional. Note that the comparator signal input to timer is the raw output from the analog comparator, not the digital CMP block.

Add a CMP _ DATA register that contains the raw value of the output from the analog comparator block and the filtered value from the digital CMP block.

The signal channel selection of the comparator has been changed compared to the LKS08x. See the manual for details.

16.3 LKS07x

3-way comparator. The signal filtering width of the 3-channel comparator can be set independently, and the filtering width range is wider.

DAC0 _ OUT can be sent to the positive input of comparator 0.

17 Encoder

Table STYLE REF 1 \ s **错误!未指定样式名。**SEQ \ * ARABIC \ s 1 **错误!未指定顺序。**Encoder resource comparison

	Equipped or not
LKS08x	Yes (non-standard, see Datasheet for details), the encoder UTIMER _ ECDx _ CNT cannot be modified by software
LKS05x	None
LKS07x	Yes, add Z signal input (clear function), and the QEP _ CNT register can be modified by software

18 UTimer

18.1 LKS08x

4-channel Timer, Timer0/1 is 16bit, and Timer2/3 is 32bit.

TIMER changes channel levels in compare mode only when UTIMER _ UNT0 _ CNT = UTIMER _ UNT0 _ CMP0/1. If the software changes the UTIMER _ UNT0 _ CNT from UTIMER _ UNT0 _ CNT < UTIMER-UNT0 _ CMP0 to UTIMER-UNT0-CNT > UTIMer _ UNT0-CMP0, the channel output level does not follow the change.

18.2 LKS05x

TIMER changes channel levels in compare mode only when UTIMER _ UNT0 _ CNT = UTIMER _ UNT0 _ CMP0/1. If the software changes the UTIMER _ UNT0 _ CNT from UTIMER _ UNT0 _ CNT < UTIMER-UNT0 _ CMP0 to UTIMER-UNT0-CNT > UTIMer _ UNT0-CMP0, the channel output level does not follow the change.

UTimer adds a software one-shot function, see UTIMER _ UNTx _ CFG [14] for details.

Table18-1UTIMER _ UNT0 _ CFG Timer 0 Configuration Register

15	14	13	12	11	10	9 8	7	6	5	4	3	2	1	0
	ONE_TRIG	SRC1	SRC0	ETON		CLK_DIV	CH1_POL	CH1_MODE	CH1_FE_CAP_EN	CH1_RE_CAP_EN	CH0_POL	CH0_MODE	CH0_FE_CAP_EN	CH0_RE_CAP_EN
	RW	RW	RW	RW		RW	RW	RW	RW	RW	RW	RW	RW	RW
	0	0	0	0		0	0	0	0	0	0	0	0	0

Location	Bit name	Explain
[31:15]		Not used
[14]	ONE_TRIG	In the compare mode, when UTIMER _ CFG [4] is 0, writing a 1 triggers Timer0 to send a pulse with a specific duty cycle for one cycle. This bit is 1 during the pulse transmission period and is automatically cleared after one Timer cycle.

Add path from comparator output to timer input

Used to source the comparator output as the signal for the Utimer capture mode

See UTINER _ UNTx _ CFG [13:12]

[13]	SRC1	Timer0 capture mode channel 1 signal source. The default is 0. 0: Chip GPIO (see Data sheet and application configuration) 1: Output of comparator 1
[12]	SRC0	Timer0 capture mode channel 0 signal source. The default is 0.

		0: Chip GPIO (see Data sheet and application configuration) 1: Output of comparator 0

Input signal filtering time in UTimer capture mode

In the UTimer block of LKS08, the signal needs 8 × n 96 MHz system clock cycles to be stable before passing through the filter. Wherein n can be 0 to 15, and when n is 0, filtering is not performed.
The clock period of this filtering is independent of the 1/2/4/8 division factor UTIMER _ UNT0/1/2/3 _ CFG [9:8] of the Timer, and the system clock is always used!

In LKS05, the running clock is modified to the filter clock Timer, that is, the clock divided by 1/2/4/8, so that the Timer filter time constant range is larger.

18.3 LKS07x

4-way Timer split independent, IE/IF and other registers independent.

Clear CNT on one-shot write.

Support two channels to capture the signal of one Timer channel at the same time, so as to facilitate the capture of the rising and falling edges of the input signal, one for the rising edge and the other for the falling edge, so as to facilitate the calculation of the duty ratio

Supports edge-aligned PWM output, complementary PWM output with dead-time, and external clock count

Timer supports rising edge and falling edge clearing

The filter coefficients of the two channels of Timer are the same and are not configured separately.

TIMER _ CMP0 = 0 of Timer is channel output all 1, CMP0 = TH + 1 channel output all 0.

Timer can be used as a capture signal source after exclusive or of two channels.

When the Timer modifies the TIMER _ CNT, the channel level immediately follows. If the software changes the UTIMER _ UNT0 _ CNT from UTIMER _ UNT0 _ CNT < UTIMER-UNT0 _ CMP0 to UTIMER-UNT0-CNT > UTIMer-UNT0-CMP0, the channel output level changes accordingly.

TIMER0 uses the center count mode to generate the complementary PWM outputs. TH/CMP0/CMP1 of TIMER0 are incremented by the shadow registers and updated at zero crossings. Whether the shadow register is enabled is software configurable. Add FAIL protection mechanism.

19 DSP

19.1 LKS08x

It has DSP module, independent program space and data space, and can run DSP program independently;

The divide instruction requires 10 bus cycles (96 MHz) to complete.

The root instruction requires eight bus cycles (96 MHz) to complete.

A trigonometric function instruction requires eight bus cycles (96 MHz) to complete.

19.2 LKS05x

Without DSP module, it is simplified to co-processing module, and DSP no longer has the function of running DSP program independently. Only CORDIC and SQRT functions are realized without division function, and trigonometric function operation is completed once in 16 system cycles.

The root instruction requires eight bus cycles (96 MHz) to complete.

A trigonometric function instruction requires 16 bus cycles (96 MHz) to complete.

19.3 LKS07x

It has DSP module, independent program space and data space, and can run DSP program independently.

Fix the problem of CORDIC arctan pattern vector length overflow.

Fix DSP DIV division limit operand problem

The divide instruction requires 12 bus cycles (96 MHz) to complete.

The root instruction requires eight bus cycles (96 MHz) to complete.

A trigonometric function instruction requires 20 bus cycles (96 MHz) to complete.

SQRT result should be a 16-bit unsigned number

When the DSP is suspended, the software can directly write to the PC to jump, which is convenient for the ARM host software to directly call the DSP function.

DSP multiplication changed from 16-bit to 32-bit

The DSP interacts with the CPU, DMA, GPIO, and CLU modules, allowing the user to use the DSP to emulate UART transceivers.

20 CLU

CLU is a newly added module of LKS07x, which is used to increase the interconnection of each module in the chip and increase the linkage mechanism. For details, please refer to UM.

21 Sleep wake-up

21.1 LKS08x

Enter hibernation through the following process

```
SYS_CLK_SLP=0xDEAD;
__WFI();
```

External GPIO wakeups can use P0.0/P0.1/P1.0/P1.1. This is set by the EXTI related register of the GPIO.

Timed wake-up supports a time interval of 0.25 to 32 seconds, which is set by the SYS _ RST _ CFG. WK _ INTV. Timed wake-up cannot be turned off. After each wake-up, the software determines whether it is GPIO external wake-up (whether there is EXIT flag) or timed wake-up.

21.2 LKS05x

Enter hibernation through the following process

```
SYS_CLK_SLP=0xDEAD;
__WFI();
```

External GPIO wakeups can use P0.0/P0.1/P1.0/P1.1. This is set by the EXTI related register of the GPIO.

Timed wake-up supports a time interval of 0.25 to 32 seconds, which is set by the SYS _ RST _ CFG. WK _ INTV. Timed wake-up cannot be turned off. After each wake-up, the software determines whether it is GPIO external wake-up (whether there is EXIT flag) or timed wake-up.

21.3 LKS07x

There are 16 external GPIO interrupt sources and 8 GPIO wake-up sources, which are more abundant. For detailed pin functions, please refer to DS. Set by AON _ IO _ WAKE _ EN and AON _ IO _ WAKE _ POL.

Enter hibernation through the following process

```
__WFI();
```

Timed wake-up can be enabled and disabled. Timed wake-up is performed through the IWDG module, and the configurable granularity of wake-up time is smaller.

22 WatchDog

22.1 LKS08x

Support 2S, 4s, 8s, 64s four-gear reset time options.

22.2 LKS05x

The register access interface of the 05 series is increased from one register to four registers.

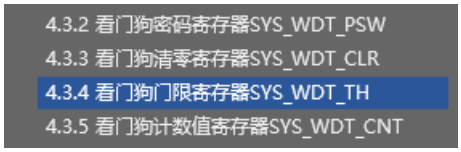

Figure 21-1

A path for reading the value of the counter, namely a SYS _ WDT _ CNT, is added, which is mainly used for judging the frequency ratio of the clock and is convenient for independent clock self-check authentication.

Increase the threshold value register SYS _ WDT _ TH, the lower 12 bits are always 0, [20:12] can be configured, the reset time range is 0.0625 s ~ 32 s, and the configuration step interval is 0.0625 s. Writing all 0s to TH causes TH to be written as 0x 1000, corresponding to a minimum reset interval of 0.0625 s. The watchdog decrements from the value of th and generates a reset when the watchdog counter counts to 0x7 to 0x0, with a reset width of 8 LRC clock cycles.

The watchdog password register SYS _ WDT _ PSW is added. Only when 0xA6B4 is written, the WDT _ CLR/WDT _ TH can be written. Writing to SYS _ WDT _ CLR or SYS _ WDT _ TH will clear the password, so the password needs to be written before writing to the watchdog every time.

Both the write SYS _ WDT _ CLR and the SYS _ WDT _ TH have the function of feeding the dog.

The default TH reset value is 32 seconds after power-up.

22.3 LKS07x

The watchdog module is changed to IWDG. IWDG _ RTH = 0 × 001000 corresponds to a minimum 4096/32kHz ≈ 128ms independent watchdog reset interval.

IWDG _ RTH = 0x1FF000 corresponds to a maximum independent watchdog reset interval of 511 × 4096/32kHz ≈ 64s.

Watchdog support as a source for sleep timed wake-up.

23 Power failure monitoring

23.1 LKS08x

Power-down threshold setting:
00: 4.5V
01: 4.2V
10: 3.9V
11: 3.6V

23.2 LKS05x

None

23.3 LKS07x

Power-down threshold setting:
00: 4.00V
01: 3.75V
10: 3.50V
11: 3.25V

24 Temperature sensor

24.1 LKS08x

After power-on, temperature sensor Gain correction coefficient A is stored in RAM variable A; temperature sensor Offset correction coefficient B is stored in RAM variable B; software code, combined with ADC sampling value, calculates the temperature.

24.2 LKS05x

After being powered on, the correction coefficient a of the temperature sensor Gain is stored in the register SYS _ TMP _ A, and the correction coefficient B of the temperature sensor Offset is stored in the register SYS _ TMP _ B.

Added temperature sensor coefficient A register SYS _ TMP _ A and temperature sensor coefficient B register SYS _ TMP _ B.

24.3 LKS07x

After power-on, temperature sensor Gain correction coefficient A is stored in RAM variable A; temperature sensor Offset correction coefficient B is stored in RAM variable B; software code, combined with ADC sampling value, calculates the temperature.

25 Version history

Table25-1Document version history

Time	Version number	Explain
2023.08.24	1.3	Remove OPAHFLF_EN
2022.12.29	1.2	Similarities and differences of adding LKS07x
2020.09.16	1.1	LKS05x VS LKS08x
2020.04.15	1.0	Initial version