



南京凌鸥创芯电子有限公司

LKS MCU 08x/05x/07x/09x

异同

© 2025, 版权归凌鸥创芯所有
机密文件，未经许可不得扩散



目 录

1	概述	7
1.1	LKS09x Highlights	7
2	Flash	8
3	SRAM	9
4	时钟	10
5	ADC	11
5.1	LKS08x	11
5.2	LKS05x	11
5.3	LKS07x	11
5.4	LKS09x	12
6	OPA	13
6.1	LKS08x	13
6.2	LKS05x	13
6.3	LKS07x	14
6.4	LKS09x	14
7	DAC	17
7.1	LKS08x	17
7.2	LKS05x	17
7.3	LKS07x	17
7.4	LKS09x	17
8	GPIO	18
8.1	LKS08x	18
8.2	LKS05x	18
8.3	LKS07x	18
8.4	LKS09x	18
9	SWD 复用	19
9.1	LKS08x	19
9.2	LKS05x	19
9.3	LKS07x	19
9.4	LKS09x	19
10	DMA	20
10.1	LKS08x	20
10.2	LKS05x	21
10.3	LKS07x	21
10.4	LKS09x	22
11	UART	23
11.1	LKS08x	23

11.2	LKS05x.....	23
11.3	LKS07x.....	23
11.4	LKS09x.....	23
12	MCPWM.....	24
12.1	LKS08x.....	24
12.2	LKS05x.....	24
12.3	LKS07x.....	25
12.4	LKS09x.....	25
13	CAN.....	26
13.1	LKS08x.....	26
13.2	LKS05x.....	26
13.3	LKS07x.....	26
13.4	LKS09x.....	26
14	SIF.....	27
15	SPI/IIC.....	28
15.1	LKS08x.....	28
15.2	LKS05x.....	28
15.3	LKS07x.....	28
15.4	LKS09x.....	28
16	CMP.....	29
16.1	LKS08x.....	29
16.2	LKS05x.....	29
16.3	LKS07x.....	29
16.4	LKS09x.....	29
17	Encoder.....	30
18	UTimer.....	31
18.1	LKS08x.....	31
18.2	LKS05x.....	31
18.3	LKS07x.....	32
18.4	LKS09x.....	32
19	DSP.....	34
19.1	LKS08x.....	34
19.2	LKS05x.....	34
19.3	LKS07x.....	34
19.4	LKS09x.....	34
20	CLU.....	36
21	EEPROM.....	37
22	休眠唤醒.....	38

22.1	LKS08x.....	38
22.2	LKS05x.....	38
22.3	LKS07x.....	38
22.4	LKS09x.....	38
23	WatchDog.....	39
23.1	LKS08x.....	39
23.2	LKS05x.....	39
23.3	LKS07x.....	39
23.4	LKS09x.....	39
24	掉电监测.....	40
24.1	LKS08x.....	40
24.2	LKS05x.....	40
24.3	LKS07x.....	40
24.4	LKS09x.....	40
25	温度传感器.....	41
25.1	LKS08x.....	41
25.2	LKS05x.....	41
25.3	LKS07x.....	41
25.4	LKS09x.....	41
26	版本历史.....	42

表格目录

表 2-1 Flash 规格对比.....	8
表 3-1 SRAM 规格对比.....	9
表 4-1 时钟规格对比.....	10
表 14-1 SIF 资源对比.....	27
表 17-1 Encoder 资源对比.....	30
表 18-1 UTIMER_UNT0_CFG Timer 0 配置寄存器.....	31
表 26-1 文档版本历史.....	42

图片目录

图 10-1 LKS08x DMA 总线架构	20
图 10-2 LKS08x 外设 DMA 请求	21
图 10-3 LKS07x DMA 总线架构	22

1 概述

本文档主要比较了 LKS08 系列、LKS05 系列、LKS07 系列、LKS09 系列的异同，便于熟悉前序系列 MCU 的工程师快速上手 LKS09 系列。概括地讲，LKS05 是 LKS08 的 Cost down 版本，部分 LKS08 中的外设模块在 LKS05 中不再存在，某些资源更加精进，具有更高性价比。LKS07 是 LKS08 的升级版本，资源规格比 LKS08 有所增加。LKS09 是 LKS07 的 Cost down 版本，部分 LKS07 中的外设模块在 LKS09 中不再存在，部分外设的功能也发生了变化。

产品问市顺序为 LKS08、LKS05、LKS07、LKS09，如无特殊描述，新产品系列继承前序系列的功能特性。

1.1 LKS09x Highlights

09x SRAM 缩减至 8kB

OPA 支持动态零漂测量，集成 5%精度相电压采样分压电阻，阻值可配置。

集成 12bit/8bit DAC 各一个，允许正负过流检测

UART 数量增加至 3 个，UART0 支持硬件波特率自适应

除法器可以保存两组结果，最多支持两个线程同时使用（举例来说，main 函数和某个优先级的中断处理函数）



2 Flash

表 2-1 Flash 规格对比

	MAIN 区域	NVR
LKS08x	32kB/64kB	1kB
LKS05x	32kB	1kB
LKS07x	64kB/128kB	1.5kB
LKS09x	64kB/128kB	1.5kB



3 SRAM

表 3-1 SRAM 规格对比

	SRAM 容量
LKS08x	8kB
LKS05x	2.56kB
LKS07x	12kB
LKS09x	8kB

4 时钟

表 4-1 时钟规格对比

	HRC/MHz	XTAL/MHz	PLL/MHz
LKS08x	4	4	96
LKS05x	4	无	96
LKS07x	8	8	96
LKS09x	8	8	96, 可以降频为 64MHz

5 ADC

5.1 LKS08x

支持 DMA 搬运，支持同步双采样功能。寄存器 ADC0_GAIN0 和 ADC0_GAIN1 分别控制不同通道的增益选择。右对齐时，饱和限制数值在 0x8000~0x7FFF。ADC 的最快转换率为 3MHz。

LKS08x 的 ADC，在采样被触发后，针对所设置第 1/2 个信号通道的信号，立即同步完成信号采样。

5.2 LKS05x

不支持 DMA 搬运，不支持同步双采样功能，不支持外部输入电源作为 ADC REF。

新增硬件触发错误中断使能位及中断标志位，分别为 ADC_IE[5]、ADC_IF[5]，。

ADC 通道总数量由 20 个减少为 16 个，通道选择信号缩减到 4-Bit 宽度。通道选择寄存器发生改变。举例：

ADC_CHN0 寄存器

位置	说明
15:12	ADC 第 3 次采样信号选择
11:08	ADC 第 2 次采样信号选择
07:04	ADC 第 1 次采样信号选择
03:00	ADC 第 0 次采样信号选择

ADC 增益选择寄存器，统一受到 SYS_AFE_REG0[6]控制，默认是 2/3 倍，对应+/-3.6V 的输入信号量程，跟 LKS08x 一致。AMC 默认值为 0x200

右对齐时，饱和限制数值范围在 0xF800~0x07FF；左对齐时，数值范围限制在 0x8000~0x7FF0，且低 4 位恒为 0。即无论是左对齐还是右对齐，数值都限制在 12bit 表示范围内。

LKS05x 的 ADC 工作时钟频率最快为 48MHz，SYS_AFE_REG7[13:8] (SAMP_TIME) 推荐配置为 0x08，对应 ADC 的最快转换率为 2MHz。

LKS05x 的 ADC 没有同步双采样功能。在 LKS05x 的 ADC 被触发后，第一个信号通道的采样需等待所设置的采样时间（由 SYS_AFE_REG7 里的 SAMP_TIME 寄存器所设置），时间结束后才是完成采样的时间节点。第二个信号通道则需要在第一个信号通道的采样、转换结束之后，才会开始进行采样。同样在采样时间结束之后完成采样。之后采样的信号通道以此类推。

5.3 LKS07x

支持 DMA 搬运。

ADC 模块由一个增加为 2 个，即可以进行同步双采样。每个 ADC 模块支持单段、两段采样触发，单个 ADC 模块不再支持 4 段采样触发。使用两个 ADC 模块分别进行 2 段采样触发更为灵活。



14 个模拟 IO 输入信号/4 个 OPA 输出/2 个 DAC 输出/温度传感器输出/模拟地均可作为 ADC 的被采样信号。常规采样通道使用 ADCx_CHN0/1/2/3 进行设置。

触发源更丰富，可以与 MCPWM/Timer/CL 单元联动进行采样触发，其中 CLU 模块类似片内的 CPLD，将芯片多个模块的关键信号进行了互联互通，便于模块联动。使用 ADCx_TRIG0/1 进行配置。

ADC DC offset 根据 ADC_CFG.DATA_ALIGN 左右对齐的设置进行硬件自动调整。无须软件干预。

增加空闲采样功能：ADC 触发分为常规触发(NT: Normal Trigger)和空闲触发(IT: Idle Trigger)两种触发。常规触发比空闲触发有更高优先级。如果 ADC 正在进行常规采样转换，此时发生了空闲触发，则 ADC 会继续完成常规触发的通道采样转换，完成后再进行空闲触发采样转换；如果 ADC 正在进行空闲采样转换，此时发生了常规触发，则常规触发会打断空闲采样转换，等常规触发采样完成后再进行空闲触发。空闲触发使用软件触发发起。ADCx_SWT=0xF00F 发起一次空闲触发，ADCx_SWT=0x5AA5 发起一次软件常规触发。空闲触发最多支持两个通道设置，通过 ADCx_ICHN 进行设置。

增加模拟看门狗，可对一路 ADC 采样信号同时检测上下阈值。

支持过采样，通过过采样对多次采样数据进行平均，可取得更高信噪比。使用 ADC_CFG 进行设置。

支持使用外部电源作为 ADC 基准，使用 SYS_AFE_REG2.REF2VDD 进行配置。

5.4 LKS09x

支持 DMA 搬运。

ADC0 支持 12 路通道选择，ADC1 支持 8 路通道选择。ADC0 支持常规采样和空闲采样，常规采样与空闲采样的总次数不能超过 12 次。但 ADC1 只支持常规采样，且总次数不超过 8 次

工作频率和采样率降低，2Msps 采样率，32MHz 工作频率。

ADC0 的常规采样/空闲采样优先级改为可配置。ADC0 空闲采样的通道数可选 0~6 个，且 ADC0 的空闲采样结果和常规采样结果共用 12 个数据寄存器即 ADC0_DAT0~ADC0_DAT11。空闲采样的结果按采样顺序依次存放至 ADC0_DAT11~ADC0_DAT6，即第 0 次空闲采样的结果存放在 ADC0_DAT11 中，第 1 次空闲采样的结果存放在 ADC0_DAT10 中。

ADC0 的空闲采样的硬件触发事件更加丰富，可以与 MCPWM/Timer/CL 单元联动进行采样触发。

ADC 通道 ADC0_CH9 变为 ADC0_CH9/2.4V ref（比 1.2V ref 驱动能力更强）。通过采样 Golden 2.4V，可以在运行时对 5V AVDD 作为 ADC ref 进行校正。

ADC 支持使用 5V 电源作为 ref。

将 ADC 的校正寄存器修改为只能硬件复位清零，软复位不影响寄存器的值



6 OPA

6.1 LKS08x

四路 OPA (0/1/2/3) , 无复用功能。

SYS_AFE_REG2[2:0]为 OPA 运放输出配置:

000: 不输出

001: OPA0 输出到 P2.7

010: OPA1 输出到 P2.7

011: OPA2 输出到 P2.7

100: OPA3 输出到 P2.7

101~111: 非法配置

P3.5 & P3.7 对应 OPA0 输入 OPA0_IP 和 OPA0_IN

P3.0 & P3.1 对应 OPA1 输入 OPA2_IP 和 OPA2_IN

P3.10 & P3.11 对应 OPA2 输入 OPA2_IP 和 OPA2_IN

P3.15 & P3.14 对应 OPA3 输入 OPA3_IP 和 OPA3_IN

片内反馈电阻比例:

00: 200k:10.4k

01: 190k:20.4k

10: 180k:30.4k

11: 170k:40.4k

6.2 LKS05x

两路 OPA (0/1) , 支持四路 OPA 输入 (非标配, 具体参见 Datasheet) , 时分复用。OPA0 对应 OPA 输入信号通道 0 和信号通道 2, OPA1 对应 OPA 输入信号通道 1 和信号通道 3。

P1.14 & P1.15 对应 OPA0 模块的输入信号通道 0: OPA0_IP 和 OPA0_IN

P1.1 & P1.2 对应 OPA0 模块的输入信号通道 2: OPA2_IP 和 OPA2_IN

P0.9 & P0.10 对应 OPA1 模块的输入信号通道 1: OPA1_IP 和 OPA1_IN

P2.14 & P2.15 对应 OPA1 模块的输入信号通道 3: OPA3_IP 和 OPA3_IN

OPA 有两种应用情形, 一种是不做时分复用, OPA0 只对应信号通道 0, OPA1 只对应信号通道 1。另一种是复用模式, OPA0 对应信号通道 0 和信号通道 2, OPA1 对应信号通道 1 和信号通道 3。

在 OPA 复用的时候, OPA 输入通道切换需插入稳定等待时间。ADC 采样 OPA, SYS_AFE_REG7[13:8] (SAMP_TIME), 需配置为 0x20 (可更大), 对应 36 个 ADC 时钟周期, ADC 输出速率 1MHz。

在 OPA 不复用的时候, SYS_AFE_REG7[13:8] (SAMP_TIME), 仍可配置为 0x08, 对应 12 个



ADC 时钟周期，ADC 输出速率 2MHz。

如果是启用了 OPA 复用功能的芯片型号,设置采样 ADC_CH8 时,内部实际采样的是 OPA2_IP/IN 信号经过 OPA0 放大后的输出;设置采样 ADC_CH9 时,实际采样的是 OPA3_IP/IN 信号经过 OPA1 放大后的输出。此时 GPIO 引脚上的 ADC_CH8 和 ADC_CH9 功能无效。

如果是未开启 OPA 复用功能的芯片型号,设置采样 ADC_CH8 时,内部实际采样的仍是 ADC_CH8;设置采样 ADC_CH9 时,实际采样的是 ADC_CH9。

SYS_AFE_REG0[5:4]为 OPA 运放输出配置:

00: 不输出

01: OPA0 输出到 P2.7

10: OPA1 输出到 P2.7

11: 非法配置

片内反馈电阻比例:

00: 200k:10.6k

01: 190k:20.6k

10: 180k:30.6k

11: 170k:40.6k

6.3 LKS07x

运放规格与 LKS08x 相同,四路 OPA (0/1/2/3),无复用功能。

片内反馈电阻比例:

00: 320k:10k

01: 160k:10k

10: 80k:10k

11: 40k:10k

此外,07x 运放输出共模电压随温度的变化相较 08x/05x 也更小,可不做共模电压的温度校正。

6.4 LKS09x

运放规格与 LKS08x 相同,四路 OPA (0/1/2/3),无复用功能。

片内反馈电阻比例:

00: 660k:20k

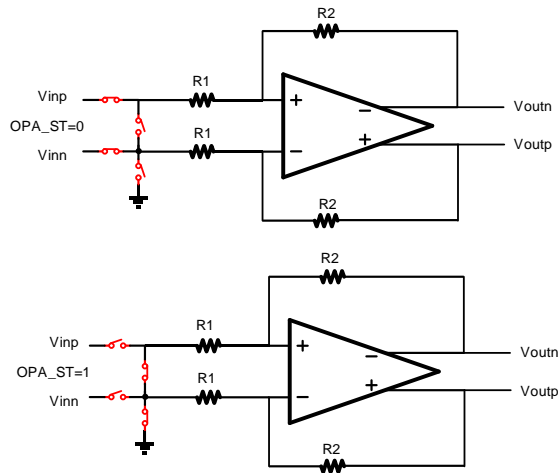
01: 640k:40k

10: 320k:40k

11:160k:40k



增加 OPA 零点偏移动态检测。运放可通过设置 $OPA_ST=1$ 来将运放工作在零漂测试模式，具体电路图如下。允许用户在芯片运行过程中进行零漂的测量和软件记录，以抵消零漂随温度变化引入的误差。

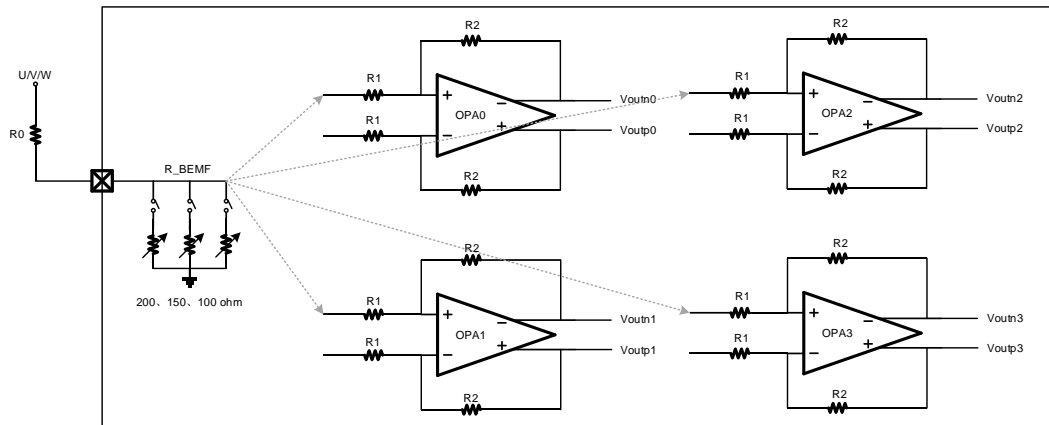


运放输入正负端内置钳位二极管。正端内置 $\pm 5\%$ 精度的反电动势分压电阻 R_BEMF ，可时分复用分别接到 OPA0/1/2/3 的 IP 端。 R_BEMF 只有一路，因此，同一时间只能进行一路反电动势的分压采样。

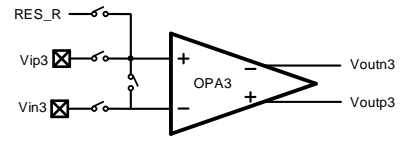
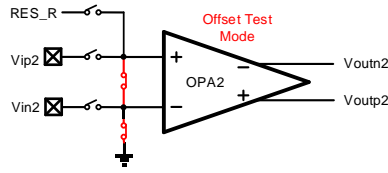
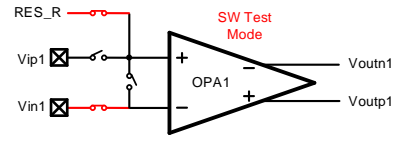
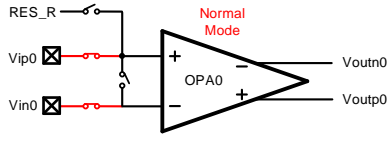
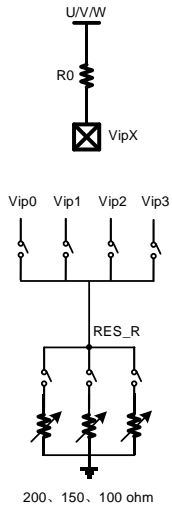
当进行电流采样时，反电动势分压电阻与 OPA 输入端断开。相电流流经采样电阻后的信号经过匹配电阻 $R0$ 后接入 OPA 正负端，进行放大。

当进行反电动势电压采样时，片外匹配电阻 $R0$ 与内置反电动势分压电阻 R_BEMF 构成分压网络，分压后的反电动势电压被 OPA 放大并进行采样。

等效电路图如下。



下图总结了 OPA 的三种工作模式。其中 Normal Mode 为正常工作模式，可进行相电流采样；SW Test Mode，IP 端有下拉电阻到地，可与外部电阻构成分压网络，可进行反电动势分压测量采样；Offset Test Mode IP/IN 端与外部连接断开，内部短接接地，可用于 OPA 零漂测量。



7 DAC

7.1 LKS08x

1 个 12bit DAC。三档量程 1.2V、3.6V 和 4.8V。切换控制位为 SYS_AFE_REG1[7:6]

7.2 LKS05x

1 个 12bit DAC。两档量程 1.2V 和 4.8V。切换控制位为 SYS_AFE_REG3[15]

7.3 LKS07x

2 个 12bit DAC。两档量程 1.2V 和 4.85V。DAC 支持使用 TIMER 进行触发递增或递减，用于发生斜坡信号，使用 SYS_AFE_DAC_CTRL 进行控制。可用于数字电源应用。

7.4 LKS09x

1 个 12bit DAC，1.2/4.8V 档位；1 个 8bit DAC。8bit DAC 接比较器负端 CMP_IN，3V/1.2V 档位。DAC 可分别连接至 CMP 的正负端，以进行正负电流的过流比较。

8 GPIO

8.1 LKS08x

四组 GPIO，有 GPIO 锁定保护；I2C 对应的 GPIO 无内部上拉；无位操作寄存器 GPIOx_BSRR 和 GPIOx_BRR；无滤波寄存器 GPIOx_PFLT。

8.2 LKS05x

三组 GPIO，无 GPIO 锁定保护；I2C 对应的 GPIO 有内部上拉；有位操作寄存器 GPIOx_BSRR 和 GPIOx_BRR；有滤波寄存器 GPIOx_PFLT。部分 GPIO 在芯片增加了使用 32kHz LRC 时钟进行的滤波，滤波时间为 4 个 LRC 时钟周期，可使用 GPIOx_PFLT[15:0]开启可关闭。默认是关闭的！需要注意的是，GPIO 唤醒和中断信号没有经过滤波。

8.3 LKS07x

四组 GPIO，无 GPIO 锁定保护；I2C 对应的 GPIO 有内部上拉；有位操作寄存器 GPIOx_BSRR 和 GPIOx_BRR；有滤波寄存器 GPIOx_PFLT。

外部唤醒增加至 8 个 IO，外部中断引脚增加至 16 个。

8.4 LKS09x

三组 GPIO。

将部分 IIC 的 GPIO 管脚复用改为 EEPROM_SCL 和 EEPROM_SDA 的 GPIO 管脚复用



9 SWD 复用

9.1 LKS08x

LKS08x 系列中 SWD IO 的复用在封装层面实现，即 SWCLK/SWDAT 与其他同一引脚的 IO（如 P0.0）在芯片中为不同 IO，只是在封装到同一个引脚。所以 GPIO 操作仍需要操作 P0.0。且需要注意 GPIO 操作不应导致 SWD 误动作。

SWCLK 和 SWDAT 不建议同时复用为 GPIO。

9.2 LKS05x

LKS05x 的 IO 复用在芯片层面实现，即 SWCLK 和 P2.13 为同一个 IO，SWDAT 和 P2.0 同一个 IO。

通过 SYS_RST_CFG[6]控制复用为 GPIO 还是 SWD。

SWCLK(P2.13)和 SWDAT(P2.0)可同时复用为 GPIO，防止锁死，上电后 30ms 内强制为 SWD，即软件配置为 GPIO，也要在上电 30ms 后才生效（若 30ms 也不够安全的话，建议应用额外增加等待时间，推荐使用原厂离线下载器进行擦写）

9.3 LKS07x

LKS07x 的 IO 复用在芯片层面实现，即 SWCLK 和 P2.14 为同一个 IO，SWDAT 和 P2.15 同一个 IO。

通过 SYS_IO_CFG.SWDMUX 控制复用为 GPIO 还是 SWD。默认为 0，为 SWD 功能。

SWCLK(P2.14)和 SWDAT(P2.15)可同时复用为 GPIO，防止锁死，上电后 30ms 内强制为 SWD，即软件配置为 GPIO，也要在上电 30ms 后才生效（若 30ms 也不够安全的话，建议应用额外增加等待时间，推荐使用原厂离线下载器进行擦写）

9.4 LKS09x

无变化，保持与 LKS07x 一致。



10 DMA

10.1 LKS08x

有 DMA 模块

如下图所示，仅端口有仲裁器(Arbiter)的外设支持 DMA 传输。不支持 flash 的 DMA 传输。
单 DMA 引擎，4 个通道。

支持多轮×多次传输。

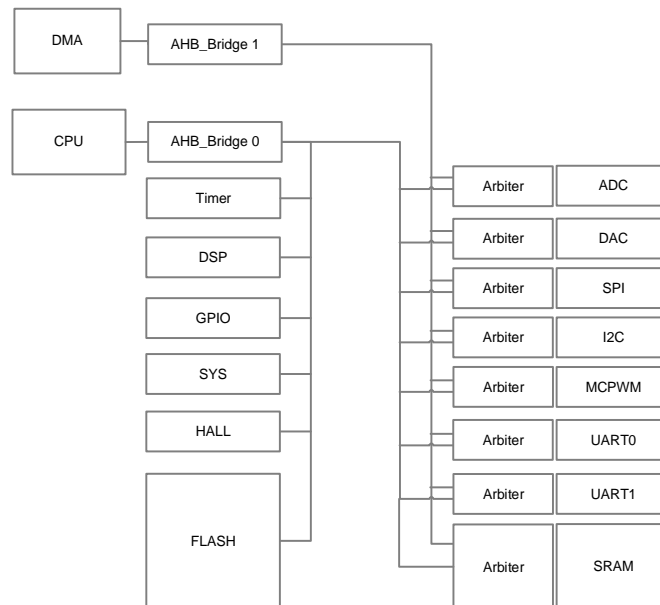


图 10-1 LKS08x DMA 总线架构

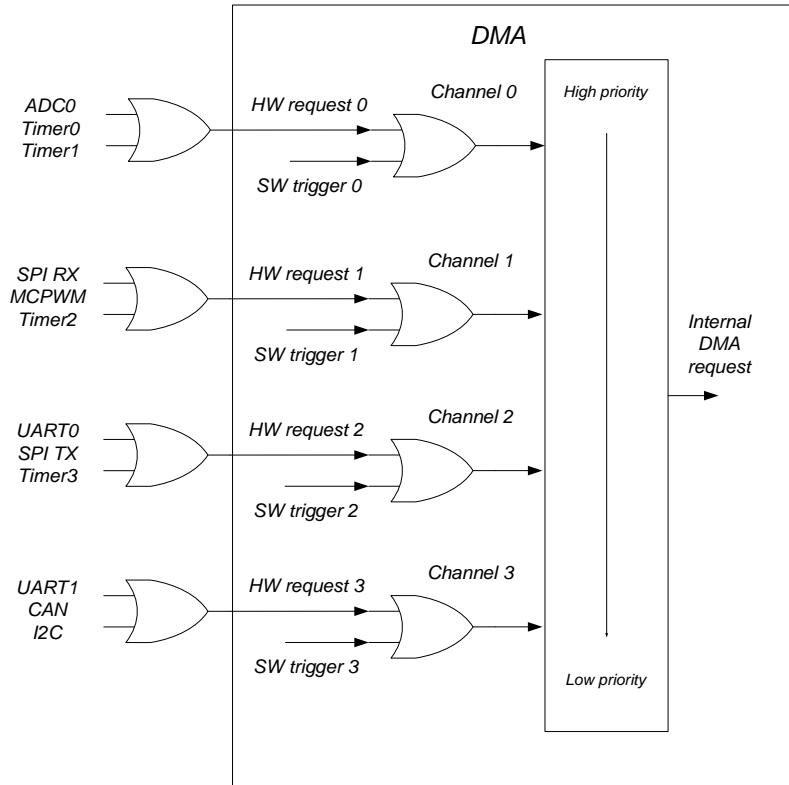


图 10-2 LKS08x 外设 DMA 请求

10.2 LKS05x

无 DMA 模块

10.3 LKS07x

有 DMA 模块，且大部分外设可被 DMA 访问。大部分设备事件可作为 DMA 请求源。典型地，相比 LKS08x，所有串行接口 UART/I2C/SPI/CAN 的 TX/RX 请求均拆分为不同的 DMA 请求，例如 UART0/1 的 TX/RX，便于使用两个 DMA 通道同时进行 UART 的收和发。

不支持多轮×多次传输，支持多轮传输，每轮传输一次数据，如 UART 收发；或一轮传输多次数据，如某些外设到内存的数据连续搬运。

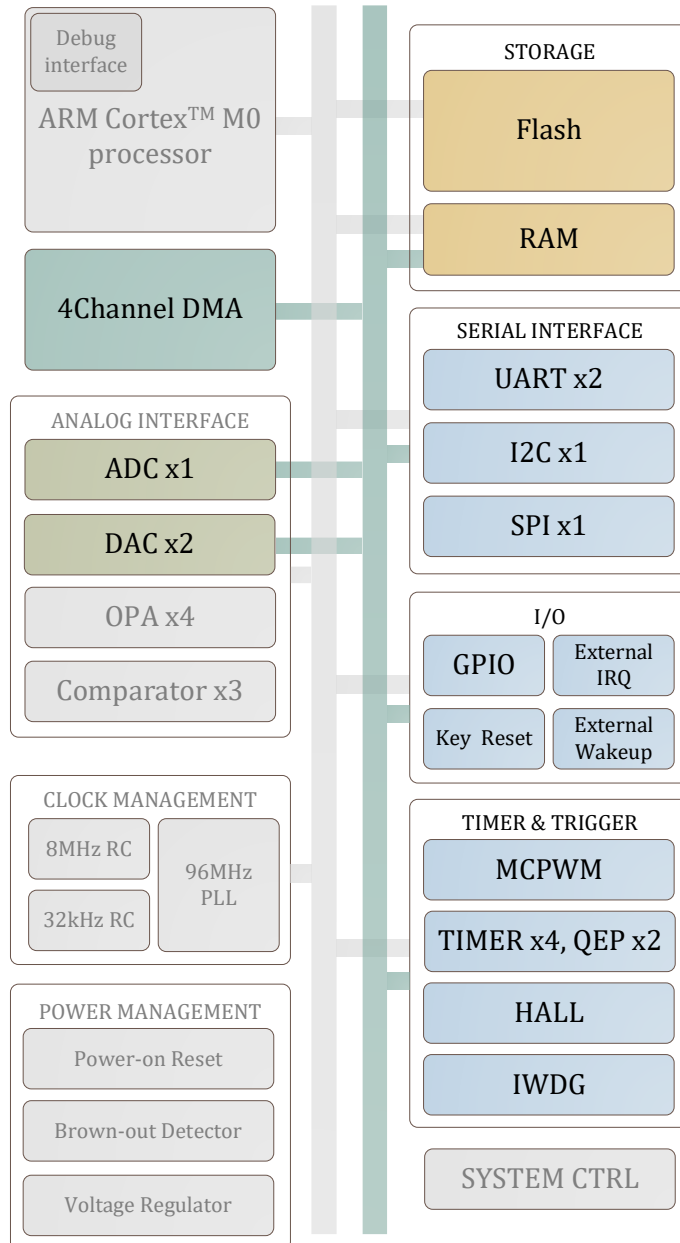


图 10-3 LKS07x DMA 总线架构

10.4 LKS09x

有 DMA 模块, DMA 通道 0/3 支持的最大搬运次数为 4095 次, DMA 通道 1/2 支持的最大搬运次数是 255。

11 UART

11.1 LKS08x

UART 有 DMA 模式
支持 7/8bit 长度数据

11.2 LKS05x

UART 无 DMA 模式
支持 7/8bit 长度数据
支持一主多从的 Multi-drop Slave/Master 模式

11.3 LKS07x

UART 有 DMA 模式
支持 8/9bit 长度数据
支持 LIN 模式 break character 收发
支持空闲帧检测
更好地支持一主多从的 Multi-drop Slave/Master 模式

11.4 LKS09x

UART 模块数量增加至 3 个。
UART0 模块增加波特率自适应功能。
支持 LIN Break Character 检测



12 MCPWM

12.1 LKS08x

4 对 PWM 通道。

对存在影子寄存器的地址进行读写访问，写入值写入到预设值寄存器，读回值读回的是**预设寄存器**的值。

MCPWM_CNT 不存在影子寄存器，软件写入立即更新。

需要注意的是，MCPWM 内部实际上对-MCPWM_TH~MCPWM_TH 的计数范围进行了平移，实际 MCPWM_CNT 的计数范围是 $0x8000-MCPWM_TH \sim 0x8000+MCPWM_TH$ 。因此，软件读出 MCPWM_CNT 的范围是 $0x8000-MCPWM_TH \sim 0x8000+MCPWM_TH$ ，写入时要按如下方式写入，写入 MCPWM_CNT 后执行一次 dummy 写操作完成更新：

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

LKS32MC08x 的 T0 时刻为 $MCPWM_CNT == 0x8000 - MCPWM_TH$ 。

PWM 通道电平只有在 $MCPWM_CNT == MCPWM_TH_{xx}$ 时才会进行改变。如果软件修改 MCPWM_CNT 使之从 $<MCPWM_TH_{xx}$ 直接修改为 $>MCPWM_TH_{xx}$ ，则通道电平不变。

发生 FAIL 事件时，所有 PWM 通道被切换至软件设置的默认电平。

12.2 LKS05x

4 对 PWM 通道。

对存在影子寄存器的地址进行读写访问，写入值写入到预设值寄存器，读回值读回的是**影子寄存器**的值，这一点与 LKS32MC08x 不同。预设值在合适时机被加载至影子寄存器。只有更新完毕，两者才一致。

在 LKS32MC05x 中 MCPWM_CNT 寄存器存在对应影子寄存器，软件写入不会立即生效。

MCPWM_CNT 可以通过向 MCPWM_UPDATE[13]写 1 进行手动更新，也可以使用硬件自动更新。如果要使能 MCPWM_CNT 的硬件自动更新，需要设置 $MCPWM_AUEN[13]=1$ 。

MCPWM_CNT 不受 MCPWM_PRT 保护。

LKS32MC05x 的 T0 时刻修改为 $MCPWM_CNT == 0x8000 + MCPWM_TH$ 。修改目的是如果需要在 T0 时刻硬件更新 MCPWM_TH，则下个周期立即使用新的 MCPWM_TH 作为 PWM 周期进行计数。否则在 LKS32MC08x 上，会出现 T0 时刻 PWM 已经用旧的 MCPWM_TH 开始计数，导致这一 PWM 周期是从 $-TH_{旧}$ 计数到 $TH_{新}$ 。通过这个改动可以 MCPWM_TH 更新，支持回零点更新且当前周期立即生效。

新增 MCPWM_AUEN 寄存器，可以选择哪些 MCPWM 寄存器被自动更新。新增 MCPWM_UPDATE[13]用于控制 MCPWM_CNT 手动更新。可以先完成 MCPWM_CNT 预加载寄存器的赋值，然后向 MCPWM_UPDATE[13] 写 1 把 MCPWM_CNT 预加载值加载到 MCPWM 内部影子计数



器。

发生 FAIL 事件时，所有 PWM 通道被切换至软件设置的默认电平。

12.3 LKS07x

6 对 PWM 通道。

MCPWM_AUEN 沿用 LKS05x 的 MCPWM 设计。

MCPWM 时基从 1 个增加为 2 个，相应的 MCPWM_CNT 增加为 2 个 MCPWM_CNT0/1，MCPWM_TH 增加为 2 个 MCPWM_TH0/1。MCPWM 的通道 0/1/2 工作于时基 0，使用 MCPWM_TH0/MCPWM_CNT0；通道 3/4/5 工作于时基 1，使用 MCPWM_TH1/MCPWM_CNT1。两个时基周期不同，初始相位可以独立设置，可通过设定某一个时基的某一个时刻作为另一个时基启动的触发，便于在驱动双电机时设置两个时基正交。

MCPWM_DTH00/01 控制通道 0/1/2 的死区设置，MCPWM_DTH00/01 控制通道 3/4/5 的死区设置。

支持 MCPWM_CNT 软件改写更新后，PWM 通道电平随之改变。

使用 MCPWM_CH_DEF 设置 6 对 PWM 通道的默认值，不再放在 MCPWM_FAIL 寄存器。

使用 MCPWM_CH_MSK 设置发生 FAIL 事件时，6 对 12 个通道中的某一个通道是否切换为默认电平，即 12 个通道可独立设置是否被 FAIL 保护。

PWM 增加了数字电源相关功能，相关寄存器为 MCPWM_TCLK、MCPWM_STT_HYST、MCPWM_ZCS_DELAY。

12.4 LKS09x

单时基(MCPWM_CNT)，4 对 PWM 通道。

去除了数字电源的相关功能。

死区控制寄存器只有 DTH00/DTH01，分别控制通道 0/1/2/3 的 P 通道的死区宽度和 N 通道的死区宽度。

MCPWM_IO 寄存器增加影子寄存器，支持 3 相 PFC 120° 相差。

MCPWM_TCLK.CLK_DIV 分频系数从 1/2/4/8 增加到 1/2/4/8/16/32/64/128。

MCPWM 增加了 MCPWM0_SW_FAIL 寄存器，实现了软件置为 FAIL_IF 功能。

13 CAN

13.1 LKS08x

有 CAN 模块,

13.2 LKS05x

无 CAN 模块

13.3 LKS07x

有 CAN 模块, CAN ID 滤波由 2 组增加为 4 组。接收 buffer, 可以接收 10 帧数据, 发送 buffer 可以存 2 帧。

13.4 LKS09x

无 CAN 模块



14 SIF

表 14-1 SIF 资源对比

	是否配备
LKS08x	有
LKS05x	无
LKS07x	有
LKS09x	有

15 SPI/IIC

15.1 LKS08x

IIC 有 DMA 模式；SPI 有 DMA 模式，数据传输长度单位不可配置（固定 8-Bit）

15.2 LKS05x

IIC 无 DMA 模式；SPI 无 DMA 模式，数据传输长度单位可配置（8-Bit 至 16-Bit）

15.3 LKS07x

IIC 有 DMA 模式；SPI 有 DMA 模式，数据传输长度单位可配置（8-Bit 至 16-Bit）

15.4 LKS09x

保持与 LKS07x 一致。



16 CMP

16.1 LKS08x

2 路比较器。

16.2 LKS05x

2 路比较器。

增加了比较器输出到 **timer** 的通路, 4 对 **timer** 可以选择 **gpio** 作为输入, 也可以选择比较器 0/1 的输出作为 **timer** 输入进行捕获。 **timer** 的捕获滤波对比较器的信号也有效, 并可选。注, **timer** 的比较器信号输入是来自模拟比较器的原始输出, 而非数字 **CMP** 模块。

增加 **CMP_DATA** 寄存器, 其中包括来自模拟比较器模块的输出原始值, 以及经过数字 **CMP** 模块滤波后的值。

比较器的信号通道选择和 **LKS08x** 相比, 有所改变, 详见手册。

16.3 LKS07x

3 路比较器。3 路比较器信号滤波宽度可单独设置, 滤波宽度范围更为宽泛。
DAC0_OUT 可以送至比较器 0 正端输入。

16.4 LKS09x

增加比较器中点输出到 **IO** 功能。比较器中心点的电阻从 **8.2kΩ** 改为 **47kΩ**。某一路 **OPA_OUT** 送至两个比较器正端 **CMP_IP**。

将 **CMP_IRQ** 拆分成 **CMP0_IRQ/CMP1_IRQ1**。即每个比较器有独立的中断编号。

比较器正负翻转点通过出厂 **Trim** 更对称。

比较器正负端支持更多输入信号。

17 Encoder

表 17-1 Encoder 资源对比

	是否配备
LKS08x	有（非标配，具体参见 Datasheet），编码器 UTIMER_ECDx_CNT 无法软件修改
LKS05x	无
LKS07x	有，加入 Z 信号输入（清零功能），QEP_CNT 寄存器可以软件修改，QEP 数量为两个
LKS09x	有，但是 QEP 数量减少为 1 个，将 TIMER 的 QEP 模块的输入固定为 TIMER2 的 GPIO 输入，使其不再受 TIMER2 的输入通道选择的影响。并且滤波系数也不与 TIMER2 共用。在 QEPO_CFG 中增加了 FLT 字段, FLT 作为 QEP 的 T1/T2/Z 信号的滤波

18 UTimer

18.1 LKS08x

4 路 Timer，Timer0/1 为 16bit，Timer2/3 位 32bit。

TIMER 在比较模式下通道电平只在 UTIMER_UNTO_CNT= UTIMER_UNTO_CMP0/1 时发生变化，如果软件修改 UTIMER_UNTO_CNT 由 UTIMER_UNTO_CNT< UTIMER_UNTO_CMP0 改为 UTIMER_UNTO_CNT> UTIMER_UNTO_CMP0，通道输出电平不会跟随变化。

18.2 LKS05x

TIMER 在比较模式下通道电平只在 UTIMER_UNTO_CNT= UTIMER_UNTO_CMP0/1 时发生变化，如果软件修改 UTIMER_UNTO_CNT 由 UTIMER_UNTO_CNT< UTIMER_UNTO_CMP0 改为 UTIMER_UNTO_CNT> UTIMER_UNTO_CMP0，通道输出电平不会跟随变化。

UTimer 增加软件单次触发功能，详见 UTIMER_UNTx_CFG[14]。

表 18-1 UTIMER_UNTO_CFG Timer 0 配置寄存器

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	RW
	0	0	0	0		0	0	0	0	0	0	0	0	0	0

位置	位名称	说明
[31:15]		未使用
[14]	ONE_TRIG	在比较模式下，且 UTIMER_CFG[4]为 0 时，写 1 触发 Timer0 发送一个周期的特定占空比的脉冲，此位在脉冲发送期间内为 1，一个 Timer 周期后，自动清零。

增加比较器输出到 timer 输入的通路

用于将比较器输出作为 Utimer 捕获模式的信号来源

详见 UTIMER_UNTx_CFG[13:12]

[13]	SRC1	Timer0 捕获模式通道 1 信号来源。默认为 0。 0: 芯片 GPIO (参见 Datasheet 及应用配置) 1: 比较器 1 的输出
[12]	SRC0	Timer0 捕获模式通道 0 信号来源。默认为 0。 0: 芯片 GPIO (参见 Datasheet 及应用配置) 1: 比较器 0 的输出

UTimer 捕获模式下输入信号滤波时间

在 LKS08 的 UTimer 模块中, 信号要 $8 \times n$ 个 96MHz 系统时钟周期稳定才能通过滤波器。其中 n 可以为 0~15, n 为 0 时, 不进行滤波。

且这个滤波的时钟周期与 Timer 的 1/2/4/8 倍分频系数 `UTIMER_UNT0/1/2/3_CFG[9:8]` 无关, 始终是使用系统时钟!

在 LKS05 中修改为滤波器时钟 Timer 运行时钟, 即 1/2/4/8 倍分频后的时钟, 使得 Timer 滤波时间常数范围更大。

18.3 LKS07x

4 路 Timer 拆分独立, IE/IF 等寄存器独立。

单次触发写入时清零 `cnt`。

支持两个通道同时捕一个 Timer 通道的信号, 这样可以便于进行输入信号上升沿下降沿的捕获, 一个捕获上升沿, 一个捕获下降沿, 便于计算占空比

支持边沿对齐 PWM 输出、带死区互补 PWM 功能输出、支持外部时钟计数

Timer 支持上升沿、下降沿清零

Timer 两个通道滤波系数相同, 不再分开配置。

Timer 的 `TIMER_CMP0=0` 是通道输出全 1, `CMP0=TH+1` 通道输出全 0。

Timer 两个通道异或后可以作为一个捕获信号源。

Timer 修改 `TIMER_CNT` 后, 通道电平立即跟随变化。如果软件修改 `UTIMER_UNT0_CNT` 由 `UTIMER_UNT0_CNT < UTIMER_UNT0_CMP0` 改为 `UTIMER_UNT0_CNT > UTIMER_UNT0_CMP0`, 通道输出电平跟随变化。

`TIMER0` 使用中心计数模式可以产生互补 PWM 输出, `TIMER0` 的 `TH/CMP0/CMP1` 增加影子寄存器, 过零时更新。是否启用影子寄存器可软件配置。增加 FAIL 保护机制。

18.4 LKS09x

减少 `TIMER` 数量。只有 3 路 Timer, 2 路 16bit, 1 路 32bit。Timer0/1 16bit, Timer2 32bit。

新增重复触发使能, 通过 `TIMERx_CFG2.RTE` 字段的配置可选择是否响应重复的单次触发行为。

当 `RTE=0` 时, `TIMER` 不会响应重复的单次触发行为。当 `RTE=1` 时, 如果 `TIMER` 单次触发仍在计数周期内再次发生单次触发, `TIMER` 会立即响应, 重新开始一个周期的单次计数。

外部启动功能与单次触发功能联动时, `TIMER` 完成一个周期的计数之后 `ONE_TRIG` 信号不再被硬件清零。

在 `TIMER` 外部重置功能的基础上新增了计数重置功能。外部重置可以和 `TIMERx_EVT.EVT_TH`



和 `TIMERx_CMP1` 配合使用。增加 `TIMERx_EVT.EVT_TH` 字段，需要发 `TIMERx_EVT.EVT_TH+1` 次外部重置事件才将 `TIMERx_CNT` 重置。当 `TIMERx_CFG2.CMP1TMIN=1` 时，`TIMERx_CMP1` 被作为重置发生的计数门限值 `TMIN` 使用，即只有当 `TIMERx_CNT>TMIN=TIMERx_CMP1` 时，才可以发生 `TIMERx_CNT` 重置。即使外部重置事件已经累积到足够次数，也需要等待 `TIMERx_CNT` 计数至 `TIMERx_CMP1` 才会重置 `TIMERx_CNT`。这种配置下，由于 `TIMERx_CMP1` 被作为重置的计数门限值 `TMIN` 使用，通常不再使用 `TIMER` 的通道 1 输出信号，通道 0 的输出信号可以正常使用。

LKS09x 只有 `TIMER0/1` 的比较/捕获事件可以作为 `ADC` 的采样触发事件。由于中心对称模式下，`CMP0/1` 在递增和递减阶段总共会命中两次，导致 `ADC` 采样也被触发两次。LKS09x `TIMER` 增加了 `TIMERx_CFGx.TRIGGER_MODE` 字段用来选择递增计数时候触发，或递减计数触发，或者递增计数和递减计数都可以触发 `adc` 采样。

`TIMER SRC` 增加 `LRC` 时钟。



19 DSP

19.1 LKS08x

有 DSP 模块，有独立的程序空间和数据空间，可自主运行 DSP 程序；
除法指令需要 10 个总线周期（96MHz）完成。
开方指令需要 8 个总线周期（96MHz）完成。
三角函数指令需要 8 个总线周期（96MHz）完成。

19.2 LKS05x

无 DSP 模块，简化为协处理模块，DSP 不再具备独立运行 DSP 程序的功能。仅实现了 CORDIC 和 SQRT 功能，无除法功能；16 个系统周期完成一次三角函数运算。
开方指令需要 8 个总线周期（96MHz）完成。
三角函数指令需要 16 个总线周期（96MHz）完成。

19.3 LKS07x

有 DSP 模块，有独立的程序空间和数据空间，可自主运行 DSP 程序
修正 cordic arctan 模式向量长度溢出问题。
修正 DSP div 除法极限操作数问题

除法指令需要 12 个总线周期（96MHz）完成。
开方指令需要 8 个总线周期（96MHz）完成。
三角函数指令需要 20 个总线周期（96MHz）完成。

SQRT 结果应为 16 位无符号数
DSP 暂停时支持软件直接写 PC 来进行跳转，便于 ARM 主机软件直接进行 DSP 函数调用
DSP 乘法从 16 位改为 32 位

DSP 可与 CPU、DMA、GPIO 和 CLU 模块进行交互，从而允许用户使用 DSP 模拟 UART 收发。

19.4 LKS09x

09 系列 DSP 模块仅支持除法运算和开方运算，不再有独立的程序空间和数据空间，无法自主运行 DSP 程序。

除法的被除数和商位宽均为 32 位有符号数，除数和余数为 32 位有符号数。
被开方数为 32 位无符号数，平方根为 16 位无符号数。
除法指令需要 12 个总线周期（96MHz）完成。



开方指令需要 8 个总线周期（96MHz）完成。

除法器可保留两组计算结果，第 0 组和第 1 组，允许两个不同优先级的中断或一个中断和主函数调用，即可以重入一次。



20 CLU

表 20- 1 CLU 资源对比

	是否配备
LKS08x	无
LKS05x	无
LKS07x	有，CLU 为 LKS07x 新加入模块，用于增加片内各模块的互联，增加联动机制，具体请参考 UM
LKS09x	有



21 EEPROM

表 21- 1 EEPROM 资源对比

	是否配备
LKS08x	无
LKS05x	无
LKS07x	无
LKS09x	有，集成 EEPROM 控制器，详情可见 LKS09xUM

22 休眠唤醒

22.1 LKS08x

通过如下流程进入休眠

```
SYS_CLK_SLP=0xDEAD;  
_WFIQ;
```

外部 GPIO 唤醒可以使用 P0.0/P0.1/P1.0/P1.1。通过 GPIO 的 EXTI 相关寄存器进行设置。

定时唤醒支持 0.25~32s 时间间隔，通过 SYS_RST_CFG.WK_INTV 设置。定时唤醒无法关闭，每次唤醒后软件判读是 GPIO 外部唤醒（是否有 EXIT 标志）还是定时唤醒。

22.2 LKS05x

通过如下流程进入休眠

```
SYS_CLK_SLP=0xDEAD;  
_WFIQ;
```

外部 GPIO 唤醒可以使用 P0.0/P0.1/P1.0/P1.1。通过 GPIO 的 EXTI 相关寄存器进行设置。

定时唤醒支持 0.25~32s 时间间隔，通过 SYS_RST_CFG.WK_INTV 设置。定时唤醒无法关闭，每次唤醒后软件判读是 GPIO 外部唤醒（是否有 EXIT 标志）还是定时唤醒。

22.3 LKS07x

共 16 个外部 GPIO 中断源，8 个 GPIO 唤醒源，更为丰富，详细引脚功能请参考 DS。通过 AON_IO_WAKE_EN 和 AON_IO_WAKE_POL 设置。

通过如下流程进入休眠

```
_WFIQ;
```

定时唤醒可以使能和禁用，通过 IWDG 模块进行定时唤醒，唤醒时间可配置粒度更小。

22.4 LKS09x

同时支高电平和低电平的唤醒信号，由单个 IO 唤醒信号扩展为八个 IO 支持唤醒信号



23 WatchDog

23.1 LKS08x

支持 2s、4s、8s、64s 四档复位时间可选。

23.2 LKS05x

05 系列寄存器访问接口由原来的 1 个寄存器增加为 4 个寄存器

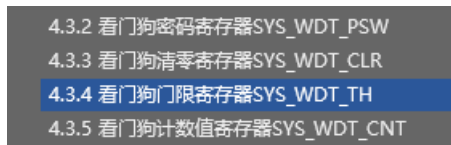


图 21-1

增加了读取计数器值的通路，即 `SYS_WDT_CNT`，主要用于时钟频率比例的判断，便于进行独立时钟自检认证。

增加门限值寄存器 `SYS_WDT_TH`，低 12 位恒为 0，[20:12]可配，复位时间范围 0.0625s~32s，配置步长间隔为 0.0625s。向 TH 写入全 0 会导致 TH 被写为 0x1000，对应最小复位间隔 0.0625s。看门狗从 TH 值开始递减，当看门狗计数器计数到 0x7~0x0 的时候产生复位，复位宽度为 8 个 LRC 时钟周期。

新增看门狗密码寄存器 `SYS_WDT_PSW`，只有在写入 0xA6B4 时，才能对 `WDT_CLR/WDT_TH` 等进行写操作，对 `SYS_WDT_CLR` 或 `SYS_WDT_TH` 的写操作会将密码清空，因此每次对看门狗进行写操作前都需要写入密码。

写入 `SYS_WDT_CLR` 和 `SYS_WDT_TH` 均有喂狗作用。

上电后默认的 TH 复位值为 32s。

23.3 LKS07x

看门狗模块改为 IWDG。IWDG_RTH=0x001000 对应独立看门狗最小复位时间间隔为 4096/32kHz≈128ms。

IWDG_RTH=0x1FF000 对应独立看门狗最大复位时间间隔为 511×4096/32kHz≈64s。

看门狗支持作为休眠定时唤醒源。

23.4 LKS09x

IWDG_RTH=0x000100 对应独立看门狗最小复位时间间隔为 256/64kHz≈4ms。

IWDG_RTH=0x1FF000 对应独立看门狗最大复位时间间隔为 511×4096/64kHz≈32s。



24 掉电监测

24.1 LKS08x

掉电阈值设定:

00: 4.5V

01: 4.2V

10: 3.9V

11: 3.6V

24.2 LKS05x

无

24.3 LKS07x

掉电阈值设定:

00: 4.00V

01: 3.75V

10: 3.50V

11: 3.25V

24.4 LKS09x

保留掉电检测模块，使用高功耗基准源

掉电检测阈值 V_{th} 选择

00:4.0V

01:3.75V

10:3.5V

11:2.8V

25 温度传感器

25.1 LKS08x

上电后，温度传感器 Gain 校正系数 A 存放在 RAM 变量 A；温度传感器 Offset 校正系数 B 存放在 RAM 变量 B；软件代码，结合 ADC 采样值，计算出温度。

25.2 LKS05x

上电后，温度传感器 Gain 校正系数 A 存放在寄存器 SYS_TMP_A；温度传感器 Offset 校正系数 B 存放在寄存器 SYS_TMP_B；软件代码，结合 ADC 采样值，计算出温度。

增加了温度传感器系数 A 寄存器 SYS_TMP_A 和 温度传感器系数 B 寄存器 SYS_TMP_B。

25.3 LKS07x

上电后，温度传感器 Gain 校正系数 A 存放在 RAM 变量 A；温度传感器 Offset 校正系数 B 存放在 RAM 变量 B；软件代码，结合 ADC 采样值，计算出温度。

25.4 LKS09x

保持与 LKS07x 一致

26 版本历史

表 26-1 文档版本历史

时间	版本号	说明
2025.02.28	1.5	细化部分 LKS09x 描述，并增加 09x highlight
2024.12.04	1.4	增加 LKS09x 版本的异同对比
2023.08.24	1.3	去除关于 OPAHFLF_EN 的说明
2022.12.29	1.2	加入 LKS07x 的异同对比
2020.09.16	1.1	LKS05x VS LKS08x
2020.04.15	1.0	初始版本

