



南京凌鸥创芯电子有限公司

LKS08x/LKS05x/LKS07x 异同 (07x Change List)

© 2022, 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
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
6	OPA	13
6.1	LKS08x	13
6.2	LKS05x	13
6.3	LKS07x	14
7	DAC	15
7.1	LKS08x	15
7.2	LKS05x	15
7.3	LKS07x	15
8	GPIO	16
8.1	LKS08x	16
8.2	LKS05x	16
8.3	LKS07x	16
9	SWD 复用	17
9.1	LKS08x	17
9.2	LKS05x	17
9.3	LKS07x	17
10	DMA	18
10.1	LKS08x	18
10.2	LKS05x	19
10.3	LKS07x	19
11	UART	21
11.1	LKS08x	21
11.2	LKS05x	21
11.3	LKS07x	21
12	MCPWM	22
12.1	LKS08x	22
12.2	LKS05x	22
12.3	LKS07x	23
13	CAN	24

13.1	LKS08x.....	24
13.2	LKS05x.....	24
13.3	LKS07x.....	24
14	SIF.....	25
15	SPI/IIC	26
15.1	LKS08x.....	26
15.2	LKS05x.....	26
15.3	LKS07x.....	26
16	CMP.....	27
16.1	LKS08x.....	27
16.2	LKS05x.....	27
16.3	LKS07x.....	27
17	Encoder	28
18	UTimer.....	29
18.1	LKS08x.....	29
18.2	LKS05x.....	29
18.3	LKS07x.....	30
19	DSP	31
19.1	LKS08x.....	31
19.2	LKS05x.....	31
19.3	LKS07x.....	31
20	CLU	32
21	休眠唤醒.....	33
21.1	LKS08x.....	33
21.2	LKS05x.....	33
21.3	LKS07x.....	33
22	WatchDog.....	34
22.1	LKS08x.....	34
22.2	LKS05x.....	34
22.3	LKS07x.....	34
23	掉电监测.....	35
23.1	LKS08x.....	35
23.2	LKS05x.....	35
23.3	LKS07x.....	35
24	温度传感器.....	36
24.1	LKS08x.....	36
24.2	LKS05x.....	36
24.3	LKS07x.....	36

25 版本历史.....37



表格目录

表 2-1 Flash 规格对比.....	8
表 3-1 SRAM 规格对比.....	9
表 2-1 时钟规格对比	10
表 13-1 SIF 资源对比.....	25
表 16-1 Encoder 资源对比.....	28
表 17-1 UTIMER_UNT0_CFG Timer 0 配置寄存器.....	29
表 24-1 文档版本历史.....	37

图片目录

图 9-1 LKS08x DMA 总线架构.....	18
图 9-2 LKS08x 外设 DMA 请求.....	19
图 9-3 LKS07x DMA 总线架构.....	20

1 概述

本文档主要比较了 LKS08 系列、LKS05 系列、LKS07 系列的异同，便于熟悉前序系列 MCU 的工程师快速上手 LKS07 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。

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



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



3 SRAM

表 3-1 SRAM 规格对比

	SRAM 容量
LKS08x	8kB
LKS05x	2.56kB
LKS07x	12kB

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

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。



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

有些应用下，会将运放的正端输出连至比较器做比较判断，此时为去除运放输出共模电压引入的误差，需要测量运放输入为 0 时的运放正端输出的共模电压。对于 08x/05x 来说，需要在上电之初、电机未运行时（此时运放输入为 0）将运放的正端信号输出到某个带有 AD 功能的 IO 口进行实际测量（详细请参考运放运用笔记里，差分 and 单端工作模式的说明），但对于 07x 来说可简化操作。正常来说，ADC 信号通道里的运放信号，采样的都是差分信号。但将模拟寄存器表里的 OPAHFLF_EN 设置为'1'后，ADC 采集运放通道信号时，采样的都是运放正端信号。因此省去将运放正端信号配置到 IO 口的操作，也可节省 IO 资源。

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



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。



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, 有 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。



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）



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。

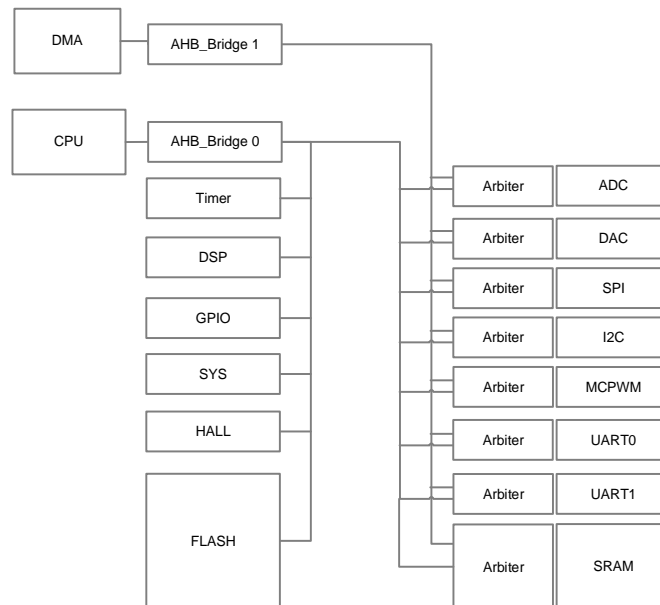


图 10-1 LKS08x DMA 总线架构

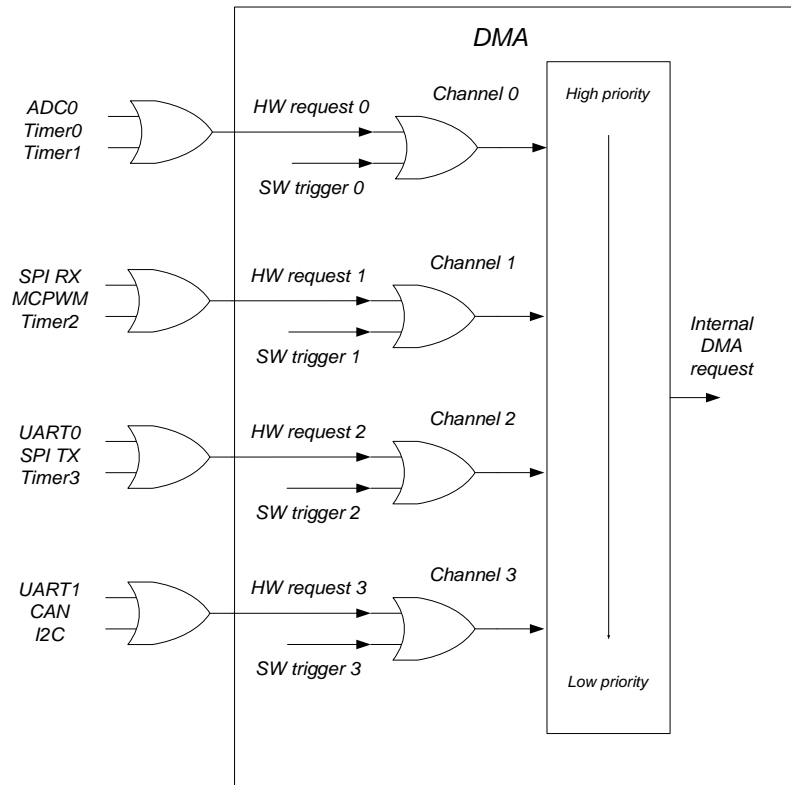


图 10-2 LKS08x 外设 DMA 请求

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。

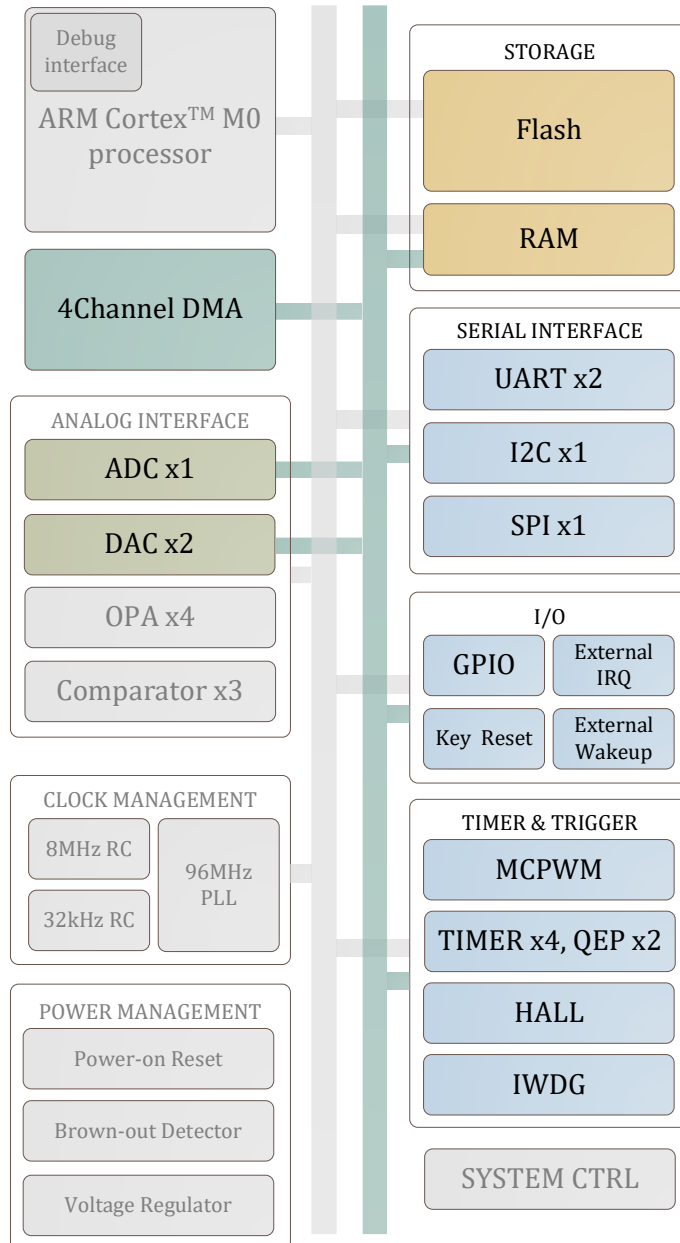


图 10-3 LKS07x DMA 总线架构

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



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_IOxx 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。



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。



14 SIF

表 14-1 SIF 资源对比

	是否配备
LKS08x	有
LKS05x	无
LKS07x	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）



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。



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

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。



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。



20 CLU

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。



21 休眠唤醒

21.1 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。

21.2 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。

21.3 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 WatchDog

22.1 LKS08x

支持 2s、4s、8s、64s 四档复位时间可选。

22.2 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

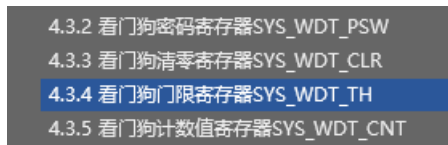


图 21-1

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。

22.3 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 $4096/32\text{kHz}\approx 128\text{ms}$ 。

`IWDG_RTH=0x1FF000` 对应独立看门狗最大复位时间间隔为 $511\times 4096/32\text{kHz}\approx 64\text{s}$ 。

看门狗支持作为休眠定时唤醒源。

23 掉电监测

23.1 LKS08x

掉电阈值设定:

00: 4.5V

01: 4.2V

10: 3.9V

11: 3.6V

23.2 LKS05x

无

23.3 LKS07x

掉电阈值设定:

00: 4.00V

01: 3.75V

10: 3.50V

11: 3.25V



24 温度传感器

24.1 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。

24.2 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。

24.3 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 版本历史

表 25-1 文档版本历史

时间	版本号	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