

32 个通过工作寄存器中的 6 个用于两两结合构成三个 16 位寄存器, 可用于间接寻址地址指针, 用于访问外部存储空间以及 FLASH 程序空间。LGT8XM 支持单周期的 16 位算术运算, 极大的提高了间接寻址的效率。LGT8XM 内核中这三个特殊的 16 位寄存器被命名为 X, Y, Z 寄存器, 将在后面详细介绍。

ALU 支持寄存器之间以及常数与寄存器之间的算术逻辑运算, 单个寄存器的运算也可以在 ALU 中执行。ALU 运算完成后, 运算结果对内核状态的影响更新到状态寄存器中(SREG)。

程序流程控制通过条件和无条件跳转/调用实现, 可以寻址到所有的程序区域。大部分 LGT8XM 指令为 16 位。每个程序地址空间对应一个 16 位或者 32 位的 LGT8XM 指令。

内核响应中断或子程序调用后, 返回地址(PC)被存储在堆栈中。堆栈被分配在系统的一般数据 SRAM 中, 因此堆栈的大小仅受限于系统中 SRAM 的大小和用法。所有的支持中断或子程序调用的应用, 必须首先初始化堆栈指针寄存器(SP), SP 可以通过 IO 空间访问。数据 SRAM 可以通过 5 种不同的寻址模式访问。LGT8XM 的内部存储空间都被线性的映射到一个统一的地址空间。具体请参考存储章节的介绍。

LGT8XM 内核包含了一个灵活的中断控制器, 中断功能可以通过状态寄存器中的一个全局中断使能位控制。所有的中断都有一个独立的中断向量。中断的优先级与中断向量地址有对应关系, 中断地址越小, 中断的优先级就越高。

I/O 空间包含了 64 个可以通过 IN/OUT 指令直接寻址的寄存器空间。这些寄存器实现对内核控制以及状态寄存器, SPI 以及其他 I/O 外设的控制功能。这部分空间可以通过 IN/OUT 指令直接访问, 也可以通过他们映射到数据存储器空间的地址访问(0x20 – 0x5F)。另外, LGT8FX8P 也包含扩展的 I/O 空间, 他们被映射到数据存储空间 0x60 – 0xFF, 这里只能使用 ST/STS/STD 以及 LD/LDS/LDD 指令访问。

为增强 LGT8XM 内核的运算能力, 指令流行线中增加了 16 位的 LD/ST 扩展。此 16 位 LD/ST 扩展配合 16 数字运算加速单元(uDSU)工作, 实现高效的 16 位数据运算。同时内核也增加对 RAM 空间的 16 位访问能力。因此 16 位 LD/ST 扩展可以在 uDSU, RAM, 以及工作寄存器之间传递 16 位的数据。具体细节请参考“数字运算加速器”章节。

算术逻辑运算单元 (ALU)

LGT8XM 内部包含了一个 16 位的算术逻辑运算单元, 能够在一个周期内完成 16 为数据的算术运算。高效的 ALU 与 32 个通用工作寄存器相连。能够在一个周期内完成两个寄存器或者寄存器与立即数之间的算术逻辑运算。ALU 的运算分为三种: 算术, 逻辑以及位运算。同时 ALU 部分也包含了一个单周期的硬件乘法器, 能够在一个周期内实现两个 8 位寄存器直接的有符号或者无符号运算。请参考指令集部分的详细介绍。

状态寄存器 (SREG)

状态寄存器中主要保存了因执行最近一次 ALU 运算而产生的结果信息。这些信息用于控制程序执行流程。状态寄存器是在 ALU 操作完全结束后更新, 这样就可以省去了使用单独的比较指令, 可以带来更加紧凑高效的代码实现。状态寄存器的值在响应中断和从中断中退出时并不会自动保存和恢复, 这需要软件去实现。