LGT8FX8P Series – Programming Manual 1.0.4J
Errata Page 14
Jake 2019 Oct 9 5:30pm
Discrepancy My edited text ATMEL text Original+Google Translate

----

As can be seen in the figure above, the second instruction will be fetched during the execution of
the first instruction. When the second instruction enters execution, the third instruction is fetched.
This eliminates the need for an additional CPU clock cycle to fetch the instruction before that
instruction can be executed.

(This description is not in the Atmel DS)

As can be seen from the above figure, the second instruction will be read during the execution of
the first instruction. When the second instruction enters execution, the third instruction is read. This
eliminates the need for additional cycles for reading instructions during the entire execution, and
from the pipeline, the efficiency of executing one instruction per Monday is achieved.

从上图可以看出,第一条指令的执行期间同时会读出第二条指令。当第二条指令进入执行期间,同时又会读
出第三条指令。这样在整个执行期间,并不需要为读取指令花费额外的周期,从流水线上看,实现了每个周一
执行一条指令的效率。

The following figure shows the internal timing concept for the Register File. In a single clock cycle
an ALU operation using two register operands is executed, and the result is stored back to the
destination register.

Figure 4-5 shows the internal timing concept for the Register File. In a single clock cycle an ALU
operation using two register operands is executed, and the result is stored back to the destina-
tion register.

The following figure shows the access timing of the general working registers. In one cycle, the
ALU operation uses two registers as operands and writes the ALU execution result to the
destination register during this cycle.

下图展示通用工作寄存器的访问时序,在一个周期内,ALU 操作使用到两个寄存器作为操作数,并在这个周
期内将 ALU 执行结果写入到目标寄存器中。

The LGT8XM supports multiple interrupt sources. These interrupts and the separate Reset Vector
each have a separate program vector in the program memory space. In general, all interrupts are
assigned separate control bits. When this control bit is set and the Global Interrupt Enable bit of the
Status Register is enabled, the CPU can respond to this interrupt.

The AVR provides several different interrupt sources. These interrupts and the separate Reset
Vector each have a separate program vector in the program memory space. All interrupts are
assigned individual enable bits which must be written logic one together with the Global Interrupt
Enable bit in the Status Register in order to enable the interrupt. / the following is not present in the
LGT datasheet but is in the Atmel DS/ Depending on the Program
Counter value, interrupts may be automatically disabled when Boot Lock bits BLB02 or BLB12
are programmed. This feature improves software security. See the section ”Memory Program-
ming” on page 294 for details.

The LGT8XM supports multiple interrupt sources. These interrupts and reset vectors correspond to a separate program vector entry in program space. In general, all interrupts have separate control bit controls. When this control bit is set and the global interrupt enable bit of the core is enabled, the kernel can respond to this interrupt.

LGT8XM 支持多个中断源。这些中断以及复位向量在程序空间都对应一个独立的程序向量入口。一般而言,所有的中断都有单独的控制位控制。当设置了该控制位,并且使能了内核的全局中断使能位后,内核才能响应这个中断。

The lowest addresses in the program memory space are reserved by default as the Reset and Interrupt Vector area. For a complete list of interrupts supported by the LGT8FX8P, please refer to the Interrupts section. This list also determines the priority of different interrupts. The lower the address, the higher the corresponding interrupt priority. RESET has the highest priority, then INT0 – External Interrupt Request 0. The start address of the interrupt vector table (except the RESET vector) *can be moved to the beginning of any 256-byte alignment, via the MCU Control Register's IVSEL bit in (MCUCR) and the IVBASE vector base address register implementation.*

The lowest addresses in the program memory space are by default defined as the Reset and Interrupt Vectors. The complete list of vectors is shown in "Interrupts" on page 57. The list also determines the priority levels of the different interrupts. The lower the address the higher is the priority level. RESET has the highest priority, and next is INT0 – the External Interrupt Request 0. The Interrupt Vectors can be moved to the start of the Boot Flash section by setting the IVSEL bit in the MCU Control Register (MCUCR). Refer to "Interrupts" on page 57 for more information. The Reset Vector can also be moved to the start of the Boot Flash section by programming the BOOTRST Fuse, see "Boot Loader Support – Read-While-Write Self-Programming, ATmega88P, ATmega168P and ATmega328P" on page 277.

The lowest program space is reserved by default as the reset and interrupt vector area. For a complete list of interrupts supported by the LGT8FX8P, please refer to the section on interrupts. This list also determines the priority of different interrupts. The lower the vector address, the higher the corresponding interrupt priority. Reset (RESET) has the highest priority, then INT0 – External Interrupt Request 0. The start address of the interrupt vector table (except the reset vector) can be redefined to the beginning of any 256-byte alignment, via the MCU Control Register The IVSEL bit in (MCUCR) and the IVBASE vector base address register implementation.
Note: the Atmega does not have the IVSEL (0x75) register at all. It is a reserved register.
最低的程序空间默认保留为复位以及中断向量区域。LGT8FX8P 支持的完整的中断列表请参考中断章节的介绍。这个列表同时也决定了不同中断的优先级。向量地址越低的中断,对应的中断优先级就越高。复位(RESET)具有最高的优先级,然后是 INT0 – 外部中断请求 0.中断向量表的起始地址(复位向量除外)可以被重新定义到任何 256 字节对齐的开始处,需要通过 MCU 控制寄存器(MCUCR)中的 IVSEL 位以及 IVBASE 向量基地址寄存器实现。

When the CPU responds to an interrupt, the Global Interrupt Enable flag (I) is automatically cleared by hardware effectively disabling interrupts. The user can nest interrupts by enabling the I-bit in software. If this is done in software, any subsequent interrupts will interrupt the current interrupt service routine. The Global Interrupt Enable I-bit is automatically set after the execution of the interrupt return instruction (RETI), so that it can respond normally to subsequent interrupts.

When an interrupt occurs, the Global Interrupt Enable I-bit is cleared and all interrupts are disabled. The user software can write logic one to the I-bit to enable nested interrupts. All enabled interrupts can then interrupt the current interrupt routine. The I-bit is automatically set when a Return from Interrupt instruction – RETI – is executed.

When the kernel responds to an interrupt, the global interrupt enable flag is set to I and is automatically cleared by hardware. The user can nest interrupts by enabling the 1-bit enable. Any subsequent interrupts will interrupt the current interrupt service routine. The I bit is automatically set after the execution of the interrupt return instruction (RETI), so that it can respond normally to subsequent interrupts.

当内核响应中断后,全局中断使能标志为 I 会被硬件自动清除。用户可以通过将 I 位使能实现中断嵌套。这样任何随后发生的中断都会中断当前的中断服务程序。I 位在执行中断返回指令(RETI)后自动置位,从而可以正常响应随后发生的中断。

There are basically two types of interrupts. The first type is triggered by an event that sets the interrupt flag. For this kind of interrupt, the Program Counter (PC) value is directly replaced with the actual interrupt vector address in order to execute the interrupt handling routine, and hardware automatically clears the interrupt flag bit. Interrupt Flags can also be cleared by writing a logic one to the flag bit position(s) to be cleared. If another interrupt condition occurs while the interrupt enable bit has been cleared, this new interrupt flag will be remembered until the interrupt enable I-bit is set, then this new interrupt will be serviced. This flag can also be cleared by software to bypass the interrupt. Similarly, if the global interrupt enable bit (SERG.I) is cleared when an interrupt occurs, the corresponding interrupt flag bit will also be set to record the interrupt event, and wait until the global interrupt enable bit is set, and will then be executed by order of priority.

There are basically two types of interrupts. The first type is triggered by an event that sets the Interrupt Flag. For these interrupts, the Program Counter is vectored to the actual Interrupt Vector in order to execute the interrupt handling routine, and hardware clears the corresponding Interrupt Flag. Interrupt Flags can also be cleared by writing a logic one to the flag bit position(s) to be cleared. If an interrupt condition occurs while the corresponding interrupt enable bit is cleared, the Interrupt Flag will be set and remembered until the interrupt is enabled, or the flag is cleared by software. Similarly, if one or more interrupt conditions occur while the Global Interrupt Enable bit is cleared, the corresponding Interrupt Flag(s) will be set and remembered until the Global Interrupt Enable bit is set, and will then be executed by order of priority.

There is a basic type of interrupt. The first type is triggered by an event, and the interrupt flag is set after the interrupt event occurs. For this kind of interrupt, after the kernel responds to the interrupt request, the current PC value is directly replaced with the actual interrupt vector address, and the corresponding interrupt service subroutine is executed, and the hardware automatically clears the interrupt flag bit. The interrupt flag can also be cleared by writing a 1 to the location of the interrupt flag. If the interrupt enable bit is cleared when an interrupt occurs, the interrupt flag will still be set to record the interrupt event. After the interrupt is enabled, the interrupt event of this record will be immediately responded. Similarly, if the global interrupt enable bit (SERG.I) is cleared when an interrupt occurs, the corresponding interrupt flag bit will also be set to record the interrupt event, and wait until the global interrupt enable bit is set. Will be executed in order of priority.

有种基本的中断类型。第一种类型由事件触发,中断事件发生后置位中断标志位。对于这种中断来说,内核响应中断请求后,当前的 PC 值被直接替换为实际的中断向量地址,执行对应的中断服务子程序,同时硬件自动清除掉中断标志位。中断标志位也可以通过向中断标志位的位置写 1 清除。如果在发生中断时,中断使能位被清除,中断标志位仍然会被设置以记录中断事件。等到中断使能后,这个记录的中断事件会被立即响应。同样,如果在中断发生时,全局中断使能位(SERG.I)被清除,对应的中断标志位也会被设置以记录中断事件,等到全局中断使能位被设置后,这些被记录的中断将会依照优先级依次执行。