APPLICATION NOTE

ST9+ INTERRUPT RESPONSE TIME

by Microcontroller Division Applications

1 INTRODUCTION

This application note presents the ST9+ interrupt response time calculation for each kind of interrupt in the best and the worst cases. The interrupt response time is the time between the interrupt event occurrence and the start of the corresponding interrupt service routine.

The different phases of interrupt processing are described in Section 2.

The arbitration phase takes place in parallel with the program flow. During this phase, the interrupt request with the highest priority is selected in order to be serviced if its priority is higher than the Current Priority Level (CPL).
The second phase is to wait for the end of the current non-interruptable sequence in process.
The last phase is the exception handling, that is to say the time required to service the interrupt.

2 INTERRUPT PROCESSING PHASES

2.1 ARBITRATION PHASE

The interrupt arbitration protocol functions continuously and completely asynchronously from instruction flow. It analyzes the different request priorities and selects the interrupt which will be serviced next.

An arbitration lasts 5 clock cycles and is composed of 5 phases (1 CK per phase):

– Phase A (CK1): start of arbitration
– Phase B (CK2): PRL<2> of CPU/Peripherals arbitration
– Phase C (CK3): PRL<1> of CPU/Peripherals arbitration
– Phase D (CK4): PRL<0> of CPU/Peripherals arbitration
– Phase E (CK5): interrupt request information arbitration

PRL<x> means ‘Priority Level Bit x’. The various (software programmable) priority levels of the peripherals and CPU are arbitrated first, and then the pending interrupt requests.

If an interrupt (or DMA) request is pending, it is acknowledged during a sixth phase:

– Phase F (CK6): interrupt acknowledge

2.2 NON-INTERRUPTABLE SEQUENCE TIMING ANALYSIS

In the ST9+ microcontroller, an instruction can be interrupted if the Register File and the memory have not yet been modified by this instruction. This feature reduces the interrupt response time.

However, each instruction contains a non-interruptable sequence which must be completed before the pending interrupt can be serviced.

The longest non-interruptable sequence is located inside the DIV instruction and lasts 22 clock cycles. For the DIVWS and MUL instructions, it lasts 20 clock cycles. For other instructions, it can be considered 16 clock cycles long.
For further calculations the worst case can be taken into account, that is to say, 22 clock cycles. However, in most cases this phase is only 16 clock cycles long.

On the other hand, the shortest non-interruptable sequence is the last clock cycle of an instruction (1 clock cycle long).

2.3 EXCEPTION HANDLING PHASE

The exception handling phase performs the following steps:

- Disables all maskable interrupt requests by clearing the IEN bit in the CICR register.
- Pushes the 16-bit PC (Program Counter) onto the system stack.
- Pushes the FLAGR register onto the system stack.
- Loads the PC with the 16-bit vector stored in the vector table, pointed to by the Interrupt Vector Register (IVR).

This phase lasts 18 clock cycles when the system stack is located in memory

If the system stack is located in the Register File, 2 more clock cycles are needed.
Instructions operating on the Register File are normally faster than those operating on the memory. This is due to the fact that the Register File access is overlapped with instruction fetches. But in the case of interrupts, it is not possible to fetch anything before all pushes are completed, so the stack in the Register File is not a faster option.
Moreover, when the stack is in the memory, you can benefit from the bus parallelism between the Data Register Bus (DRB) and the Multiplexed Memory Bus (MMB).
This is why, during exception handling, using the system stack in the Register File is 2 clock cycles longer than using the system stack in memory.

If you need to access more than 64K bytes of interrupt routine code, you have to save the CSR register in the stack during exception handling, together with the PC and the Flag register. This is done by setting the ENCSR bit in the EMR2 register.
With this option, 2 additional clock cycles are needed to complete the interrupt cycle, as the CSR is pushed onto the system stack.

Depending on the options you have chosen in your application, you will have different results for your interrupt response duration. This is summarized in the following table.

Table 1. Exception handling duration

ENCSR	System Stack	Exception Handling Duration
0	memory	**18** Clock Cycles
1	memory	**20** Clock Cycles
0	Register File	**20** Clock Cycles
1	Register File	**22** Clock Cycles

3 INTERRUPT RESPONSE TIME CALCULATIONS

From the information described in the Section 2, the best and worst case interrupt response times have been calculated for the various existing sources of ST9+ interrupts:

– Internal interrupts
– External interrupts
– NMI pin Top Level Interrupt
– Watchdog timer Top Level Interrupt

For the times listed below, the phases during which the described event takes place are shown in brackets. These phases are described in the Section 2.1 (Arbitration Phase).

3.1 INTERNAL INTERRUPTS

Internal interrupts are interrupts generated by the ST9+ on-chip peripherals.
For example, the Multifunction Timer 'End of Count' interrupt, or the Serial Peripheral Interface 'end of byte transmission' interrupt.

■ **Best case**

– Valid arbitration + acknowledge (PhA,PhB,PhC,PhD,PhE,PhF): 6CK
– Shortest non-interruptable sequence: 1CK
– Exception handling duration (best case): 18CK

TOTAL 25CK

■ **Worst case**

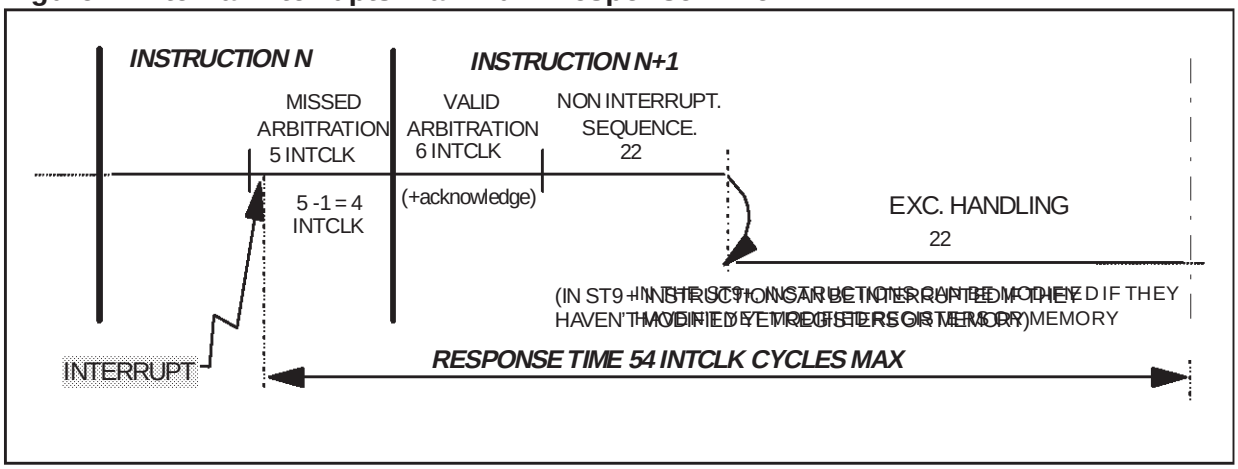
– Missed arbitration (PhB,PhC,PhD,PhE): 4CK
– Valid arbitration + acknowledge (PhA,PhB,PhC,PhD,PhE,PhF): 6CK
– Longest non interruptable sequence: 22CK
– Exception handling duration (worst case): 22CK

TOTAL 54CK

As an arbitration (without acknowledge) is 5 clock cycles long, the arbitration is missed if the interrupt occurs just after the beginning of the arbitration phase (during PhB): so, a missed arbitration is 4 clock cycles long.

Refer to Figure 1 for an illustation of the worst case response time with the various phases used in the calculation.

Figure 1. Internal interrupts maximum Response Time



3.2 EXTERNAL INTERRUPTS

External interrupts are interrupts coming from the external ST9+ interrupt pins.
For example, a falling edge on the INT2 pin.

Note: Interrupts coming from ST9+ on-chip peripherals but sharing an external interrupt channel are not external interrupts: they are internal interrupts. Please refer to the previous paragraph for the internal interrupts response time calculations.

- **Best case**
  - External interrupt sampling (PhE): 1CK
  - Valid arbitration + acknowledge (PhA,PhB,PhC,PhD,PhE,PhF): 6CK
  - Shortest non-interruptable sequence: 1CK
  - Exception handling duration (best case): 18CK

  TOTAL 26CK

- **Worst case**
  - External interrupt sampling (PhA): 1CK
  - Missed arbitration (PhB,PhC,PhD,PhE): 4CK
  - Valid arbitration + acknowledge (PhA,PhB,PhC,PhD,PhE,PhF): 6CK
  - Longest non interruptable sequence: 22CK
  - Exception handling duration (worst case): 22CK

TOTAL 55CK

3.3 NMI PIN TOP LEVEL INTERRUPT

The NMI pin Top Level Interrupt is the interrupt generated by a rising or a falling edge (depending on the chosen configuration) on the NMI pin of the ST9+. The NMI interrupt is a Top Level Interrupt The NMI source is selected when the TLIS bit of the EIVR register is reset (=default value).

- **Best case**
  - External NMI interrupt sampling (PhD): 1CK
  - Valid arbitration + acknowledge (PhE,PhF): 2CK
  - Shortest non interruptable sequence: 1CK
  - Exception handling duration (best case): 18CK

  TOTAL 22CK

As the NMI is a Top Level Interrupt, it overrides every other priority. It is why PhA, PhB, PhC, and PhD of the arbitration are not required in the Top Level Interrupt arbitration: a Top Level Interrupt sampled during PhD is thus taken into account (best case above).
As shown below, the worst case is when the NMI interrupt is sampled during the next phase (PhE), so that all the arbitration phases must be completed in this case (even if they are not useful for the Top Level Interrupt).

- **Worst case**
  - External NMI interrupt sampling (PhE): 1CK
  - Valid arbitration + acknowledge (PhA,PhB,PhC,PhD,PhE,PhF): 6CK
  - Longest non interruptable sequence: 22CK
  - Exception handling duration (worst case): 22CK

  TOTAL 51CK

3.4 WATCHDOG TIMER TOP LEVEL INTERRUPT

The Watchdog Timer 'End of Count' (EOC) interrupt is generated when the watchdog timer counter reaches 0 (if the Watchdog timer is used as a normal timer, not in Watchdog mode). This EOC interrupt can be configured to be the source of the INTA0 interrupt channel or the source of the Top Level Interrupt channel.
The Watchdog Timer interrupt configured to share the interrupt channel of INT0 pin is an internal interrupt (see §3.1 for corresponding calculation). The calculation below is relevant for the Watchdog timer interrupt configured as a source of the Top Level Interrupt request.

- **Best case**
  - Valid arbitration + acknowledge (PhE,PhF): 2CK
  - Shortest non-interruptable sequence: 1CK
  - Exception handling duration (best case): 18CK

  TOTAL 21CK

- **Worst case**
  - Valid arbitration + acknowledge (PhA,PhB,PhC,PhD,PhE,PhF): 6CK
  - Longest non-interruptable sequence: 22CK
  - Exception handling duration (worst case): 22CK

  TOTAL 50CK

4 SUMMARY

All the results of the different cases studied before are summarized in the following table. Moreover is calculated the time corresponding to the calculated clock cycles, with a ST9+ internal frequency of 24 MHz.

Table 2. Interrupt response time summary

Kind of Interrupt	WORST CASE		BEST CASE	
	Clock Cycles	µs (at 24MHz)	Clock Cycles	µs (at 24MHz)
Internal Interrupt	54	2.25	25	1.04
External Interrupt	55	2.29	26	1.08
NMI Top Level Interrupt	51	2.12	22	0.92
Watchdog Timer TLI	50	2.08	21	0.87

Thanks to these values, you can calculate the duration of any of your interrupts, by adding the corresponding interrupt response time value found in the table above to your interrupt code time, plus the length of the IRET instruction (12, 14, or 16 clock cycles, depending on the value of the ENCSR bit in the EMR2 register, and of the system stack location, in the same way as described in the exception handling duration calculation in the paragraph §2.3).

THE PRESENT NOTE WHICH IS FOR GUIDANCE ONLY AIMS AT PROVIDING CUSTOMERS WITH INFORMATION REGARDING THEIR PRODUCTS IN ORDER FOR THEM TO SAVE TIME. AS A RESULT, STMICROELECTRONICS SHALL NOT BE HELD LIABLE FOR ANY DIRECT, INDIRECT OR CONSEQUENTIAL DAMAGES WITH RESPECT TO ANY CLAIMS ARISING FROM THE CONTENT OF SUCH A NOTE AND/OR THE USE MADE BY CUSTOMERS OF THE INFORMATION CONTAINED HEREIN IN CONNEXION WITH THEIR PRODUCTS.