SIEMENS

Microcontrollers AP2924

ApNote

Interrupt Register behavior of the CAN module in Siemens 16-bit Microcontrollers

This document describes the update behavior of the CAN INTID value after INTPND bit is reset.

Author: Tobias Wenzel / HL DC AT/ Siemens Semiconductor

AP292401 ApNote - Revision History		
Actual Revision : Rel. 1.0	Previous Revison: none	
Page of actual Rel.	Page of prev. Rel.	

1. Behaviour Description

The INTID value in the CAN Interrupt Register contains the interrupt request (see also INTPND) of the CAN buffer with the highest priority (error status int, obj15, obj1..obj14). When no CAN interrupt is pending, INTID will have the value ‚0'. In order to update INTID the INTPND bit in Message Control Register of the dedicated buffer has to be reset. As this update process is done by the internal CAN state machine there is a delay time between reseting INTPND and the update of INTID. Basically this time depends on the current status of the CAN state machine. The CAN state machine rotates between three different states: Match, Read, Write. Each state needs two clock periods (=8TCL) execution time, which leads to a maximum delay time of 6 clock periods (24 TCL).

In case of consecutive CPU accesses to the CAN memory when a CAN message from the CAN bus is stored, additional clock periods are added between Match, Read and Write states. The reason for this behavior is the single ported RAM structure of the CAN memory. This increases the delay time for the INTID update process when CAN prescaler values lower than ‚2' (BRP=0) are used.

The following Table1 summarizes the behavior:

Baud Rate Prescaler	Time quanta (tq) per CAN bit time	Maximum INTID Update Delay time (in TCL) without CPU access	Worst case INTID Update Delay (in TCL) with consecutive CPU accesses and CAN message storage from CAN bus
1 (BRP=0)	6	24	248
1 (BRP=0)	>6	24	104
>1 (BRP>0)	>=6	24	24

2. Application Hint

This delay time of INTID can not be avoided as the CAN state machine is working independent from the CPU. In order to evaluate the delay time in CPU cycles the user has to take care about effects of the Core pipeline, when exactly INTID is reset (see next chapter)! It is recommended to reset INTPND in the interrupt routine as soon as possible.

3. Software Considerations

In order to transfer the delay times from Table1 into application software the following parameters need to be considered:

- Execution of the program code from the external or internal memory
- in case of external program execution: configuration of involved BUSCONx registers
- Pipeline of the C166 core (time calculation between clear INTPND instruction in the write back phase and read INTID in the decode phase)
- Access to the CAN interface is fixed to 2 waitstates and 16-bit demux bus (see respective C16x Manual)

The following examples include the considerations above. Eamples 1 and 2 are based on a BUSCON value of 04AFH (external 16-bit demux bus, 0 waitstates):

Example1: execution from external memory with BRP>0:

```
MOV MSG_CTLx,RX             ; reset INTPND from buffer x
NOP                         ; wait1
NOP                         ; wait2
NOP                         ; wait3
NOP                         ; wait4
NOP                         ; wait5
MOV RX, CAN_INTID           ; read updated INTID
```

Example2: execution from external memory with BRP=0:

```
ATOMIC #3                   ; interrupts are not allowed for the next three instructions
MOV MSG_CTLx,RX             ; reset INTPND from buffer x
NOP                         ; wait1
ATOMIC #3                   ; interrupts are not allowed for the next two instructions
NOP                         ; wait3
NOP                         ; wait4
NOP                         ; wait5
MOV RX, CAN_INTID           ; read updated INTID
```

Example3: execution from internal memory with BRP>0:

```
MOV MSG_CTLx,RX             ; reset INTPND from buffer x
NOP                         ; wait1
NOP                         ; wait2
NOP                         ; wait3
NOP                         ; wait4
NOP                         ; wait5
NOP                         ; wait6
NOP                         ; wait7
NOP                         ; wait8
MOV RX, CAN_INTID           ; read updated INTID
```

Example4: execution from internal memory with BRP=0:

```
ATOMIC #4                   ; interrupts are not allowed for the next four instructions
MOV MSG_CTLx,RX             ; reset INTPND from buffer x
NOP                         ; wait1
NOP                         ; wait2
ATOMIC #4                   ; interrupts are not allowed for the next four instructions
NOP                         ; wait4
NOP                         ; wait5
NOP                         ; wait6
ATOMIC #1                   ; interrupts are not allowed for the next four instructions
NOP                         ; wait8
MOV RX, CAN_INTID           ; read updated INTID
```