FAQ - Understanding CS35L36A Interrupts

1 Scope

This document provides further details on how to use the interrupt (INT) pin on the CS35L36A device, with examples on how masking works. The aim is to help debug issues where the INT pin is asserting unexpectedly.

2 Supporting Documents

This document provides complementary information to the CS35L36A datasheet (DS1205F1) and should be read alongside it. You can access the datasheet here: https://github.com/CirrusLogic/product-support/blob/cs35l36a/datasheet/CS35L36A_DS1205F1.pdf

3 Assumptions & Pre-Requisites

This document assumes that:

- The INT pin is enabled and configured for the required polarity and drive type (open-drain or push-pull) according to Table 4-32 in the datasheet.

4 Question – How do the x_STS, x_EINT and x_MASK fields affect the INT pin?

Refer to Figure 4-29 in the datasheet:

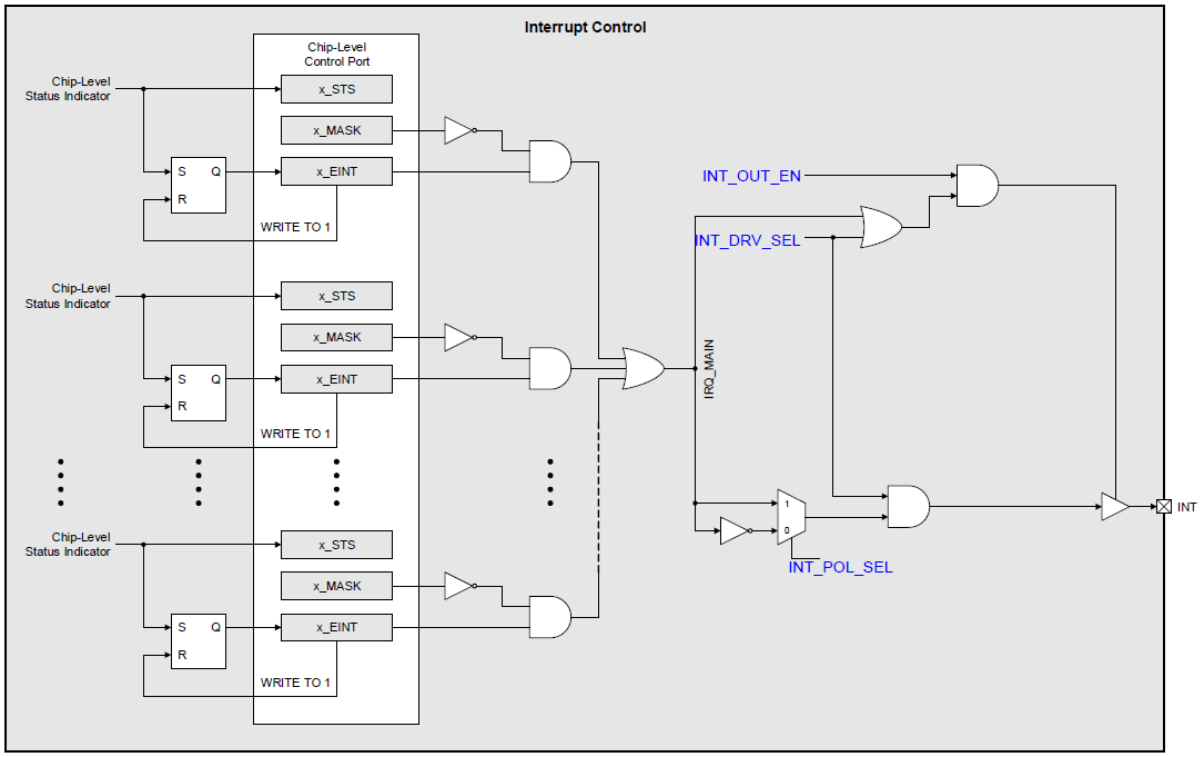


Figure 4-29. Interrupt Control

There are three fields per interrupt source:

- **x_STS**: contains the current status of an interrupt source
- **x_EINT**: provides a latched (or 'sticky') version of the x_STS bit
- **x_MASK**: controls whether the x_EINT will cause the INT pin to be asserted or not

If x_MASK = 1, then when x_EINT = 1 the INT pin will not be asserted. In other words, the interrupt is **masked**.

If x_MASK = 0, then when x_EINT = 1, the INT pin will be asserted. In other words, the interrupt is **unmasked**.

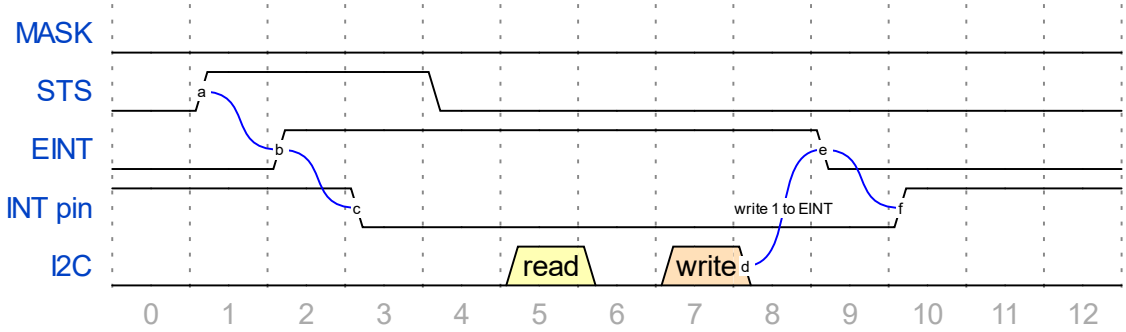
The INT pin will remain asserted for as long as any ***unmasked*** x_EINT field is a 1.

5 Question – How do I clear the INT pin?

To clear the sticky x_EINT fields, they must be written with a 1.

Writing a 0 will not clear the x_EINT field, and the INT pin will remain asserted.

5.1 Worked Example



- t0: MASK field is 0, meaning the EINT is unmasked.
- t1: STS goes 0→1
- t2: EINT goes 0→1 as a result of STS
- t3: EINT is unmasked (MASK = 0), so INT pin is asserted (assume active low)
- t4: STS goes 1→0, however EINT remains 1 because it is sticky and has not been cleared yet
- t5: AP reads EINT register to discover source of interrupt
- t7: AP writes EINT = 1 to clear the interrupt
- t9: EINT goes 1→0 because it has been cleared by AP
- t10: INT pin is de-asserted because EINT = 0

6 Question – How to debug issues with INT pin asserting unexpectedly?

The following steps should be followed immediately after INT pin is asserted:

1. What interrupt sources are unmasked? Check registers 0xD0_0040 to 0xD0_00FC.
2. What EINT fields are set to 1? Check registers 0xD0_0000 to 0x40_000C.

Any EINT bits set to 1 while their respective MASK bits are set to 0 will be causing the INT pin to assert.

7 Revision History

Revision History

Revision	Changes
28 OCT 2019	• Initial version.

Contacting Cirrus Logic Support

For all product questions and inquiries, contact a Cirrus Logic Sales Representative.
To find the one nearest you, go to www.cirrus.com.