EZ_Port Pin Standard Config Pull Up Down

Search all content

Options

06-30-2020 01:20 AM • 453 Views

katzorkej
Contributor II

Hi,

i am using a K22FN1M0VLQ12 and was wondering if the EZP_CS! pin has an internal pullup.

I am having trouble with a board because EZP_CS! is externally pulled low, so its entering EZPort mode and i cannot JTAG the device. On other boards this pin is floating and JTAG is working fine, so my conclusion is that this pin is internally pulled high, correct?

Thank you.

Add tags

1 Kudo

REPLY SHARE

Sort: Oldest to Newest

All forum topics ‹ Previous Topic Next Topic ›

EZ Replies

06-30-2020 09:59 AM • 124 Views

mykepredkomimet
Senior Contributor II

Hi ,

I think that's a safe assumption that EZ_CSi (or EZ_CS_b as I see it in the documentation) is pulled high. I've tried using the pin functions available through the mux with EZ_CS_b with really no success coming up with a practical way of having the pin connected to a high impedance input during reset so that it can be used in IO limited situations.

After more experimentation than I would care to admit, I would recommend leaving EZ_CS_b floating when using it with JTAG for programming (and driven low when programming through the EzPort) with no plan for using the pin for any other function.

I should point out that I think EZ_CS_b is also the NMI interrupt pin on the Kinetis parts (at least all the devices I've worked with) and adding this functionality comes under the heading of being more work than it's worth - leave the pin floating.

myke

Add tags

1 Kudo

REPLY SHARE

07-01-2020 02:48 AM • 124 Views

katzorkej
Contributor II

Hi Myke,

thank you for your answer!

In another project we are using the PTA4 as an output to three 74HC4051 mux ICs controlling the Enable! pin and it is working fine.

This time i messed up because the port is used as an input and the connected IC is pulling it low during boot.

The problem is that this was the last free pin on the µC as everything is in use already.

Normally i would agree and say dont use this pin. It is always good for some trouble in toolchains and hardware.

Add tags

0 Kudos

REPLY SHARE

07-01-2020 08:08 AM • 124 Views

ErichS
Senior Contributor II

Just jumping in and this maybe helps:

what I'm doing is disabling the EZport pin, see Disabling EzPort on NXP Kinetis to Solve Power-On Issues | MCU on Eclipse (with Processor Expert).

But there should be a configuration value too, similar to

Disabling NMI (Non Maskable Interrupt) Pin | MCU on Eclipse

That should solve the problem too?

Add tags

0 Kudos

REPLY SHARE

07-01-2020 10:24 AM • 124 Views

mykepredkomimet
Senior Contributor II

,

I've done exactly the same thing and when I looked at what I had to do to ensure the pin was high during boot, I found it to be easier/cheaper to use a Kinetis in a package with more pins.

,

Thank you for the post. The only thing I would say is that kind of solution is only good for production with devices provided to the board stuffer pre-programmed and the boards are cheap enough that they can be discarded if they don't work at the end of the line. The solution is problematic when you are doing initial product bring up and prototyping as well as debugging defective product. Years ago I was involved in the production of a board that used another OEM's SoC that the designer used the ability to change the base configuration to make more IO pins available - to make a long story short, we ended up respinning the board to use another device in the family with more IO and things magically became more manageable and the overall program cost was reduced.

Add tags

0 Kudos

REPLY SHARE

07-02-2020 01:13 AM • 124 Views

katzorkej
Contributor II

Ah ok, i think my advantage is that we have an external watchdog which holds the processor in reset for a few 100ms at bootup, so i dont have issues with levels not settling in time. I already switched two pins so that the EZP_CS_now used as an output pin and i pulled it externally high additionally. This seems to work well, but i now have to do a hardware revision.

I already found your blog post while googling and i really like it. Unfortunately it doesnt work with my problem because i am not able to write to the FOPT register because EZPort is already enabled when booting and the only way to write to FOPT would be by using EZPort, whe im not totally wrong.

Well, while i was typing this i started to doubt. How do you program the FOPT to disable the EZPort when your board seems to enter EZPort mode due to your pin rising slow? Which programmer do you use for programming with Processor Expert.

Add tags

0 Kudos

REPLY SHARE

07-02-2020 03:08 AM • 124 Views

ErichS
Senior Contributor II

katzorkej@mx-vk.eu,

I'm not using the EzPort in any of my projects, that's why I always disable it to avoid any troubles, and so I can use the pins any way I want. I'm disabling the functionalty like the NMI pin, as outlined in Disabling NMI (Non Maskable Interrupt) Pin | MCU on Eclipse : means setting the FOPT/configuration register in my application and that way it gets programmed/flashed and thus disabled from the beginning. I'm using various debug/programming devices: from re-using the OpenSDA circuit for the very low cost projects (see Using the Freedom Board as SWD Programmer | MCU on Eclipse and even embedd it on my own boards) up to using the really good external P&E and SEGGER debug probes, along with the inexpensive LPC-Link2 (Custom 3D Printed Enclosure for NXP LPC-Link2 Debug Probes | MCU on Eclipse) which works both for Kinetis and LPC.

Erich

PS: thanks :-)

Add tags

REPLY SHARE

07-02-2020 03:20 AM • 124 Views

katzorkej
Contributor II

The dilemma i see is that you can only disable the EZPort/NMI functions via FOPT when you are able to program your device.

Or other way round in my case, EZPort is by default enabled in Kinetis but the hardware circuitry is preventing me to program my device by JTAG so i cannot disable EZPort to program my device via JTAG.

I was looking for a way around this but it seems that the only possibilities are writing the flash via ezport to disable ezport or to redesign my circuit.

The lessons learned for me are

- dont use the PTA4/EZP_CS! as long as you got alternatives

- if you use it, try to use it as an output with external pullups to be sure

- disable NMI/EZPort when possible

Add tags

0 Kudos

REPLY SHARE

07-02-2020 08:12 AM • 124 Views

mykepredkomimet
Senior Contributor II

Just thinking about the situation and upon some thought, I would recommend that you change the "lessons learned" bullet two to:

- if you use EZP_CS!, put in a jumper that allows you to float EZP_CS! during programming

It's just that you can't always know what's going on with inputs of logic chips, especially during the power up sequence.

You haven't said whether or not this is for a one off project (put in the jumper) or for a product. If it was for a product, I would plan for the Kinetis devices to be preprogrammed before assembly and I would put in a zero Ohm resistor at EZP_CS! that you can pull easily to float the pin if you need to do some programming down the line - not optimal but better/cheaper than a jumper.

Add tags

1 Kudo

REPLY SHARE

07-02-2020 11:58 PM • 124 Views

katzorkej
Contributor II

I like your idea and for a few development boards i would build it like you suggested.

This project i am working on is for "mass" production (in the low 1000 per anno). We have some possibilities with a fully automatic flying probe tester at the end of the production line which we use for boundary scan and JTAG programming. It should be able to pull pins in some direction during JTAG programming, but changing the PCB assembly by removing SMD jumper is not an option.

I understand your point with the floating and input pins during power up, but in my case with an external watchdog, releasing the reset pin delayed by several 100ms the levels of the inputs should be settled. This should work, shouldnt Or are you thinking about some latchup behaviour?

Add tags

👍 0 Kudos

REPLY SHARE

07-03-2020 07:41 AM • 124 Views

mykepredkomimet
Senior Contributor II

Hey ,

Interesting - my experience with flying probe testers was last about 15 years ago so I'm not familiar with the state of the art. We didn't have anything that would provide JTAG programming - can you send me a link to the equipment you're using?

I'm recommending the jumper/zero Ohm to eliminate the cost and complexity of adding an external watchdog and ensuring that you don't get any kind of latchup behaviour.

Regardless, at 1k units, it should be cheaper for you to preprogram the parts before putting them on the board and use a zero Ohm as a jumper (to allow debugging later).

Good luck! This sounds interesting.

myke

Add tags

👍 1 Kudo

REPLY SHARE

07-05-2020 11:47 PM • 124 Views

katzorkej
Contributor II

We cannot get rid of the external WatchDog as it is mandatory in the standards we are using (industrial safety devices VdS,UL,FM, etc).

I changed my prototype with switching pins and a pullup and it seems to work. I will try this solution because preprogramming is not common practice here.

Thank you!

Add tags

0 Kudos

REPLY SHARE

07-06-2020 08:02 AM • 124 Views

mykepredkomimet
Senior Contributor II

Hi ,

Very cool - good luck and if you have any questions, you know where to look!

myke

Add tags

1 Kudo

REPLY SHARE

Reply to the topic...

POST REPLY

©2006-2021 NXP Semiconductors. All rights reserved.