build drivers/pci/host/pci-pegmatite.c时报如下warning

```
| WARNING: drivers/built-in.o(.text+0x29b80): Section mismatch in reference from the function pegmatite_pcie_enable() to the function .init.text:pci_assign_unassigned_resources()
| The function pegmatite_pcie_enable() references
| the function __init pci_assign_unassigned_resources().
| This is often because pegmatite_pcie_enable lacks a __init
| annotation or the annotation of pci_assign_unassigned_resources is wrong.
|
| WARNING: drivers/built-in.o(.data+0x1314): Section mismatch in reference from the variable pegmatite_pcie_driver to the function .init.text:pegmatite_pcie_probe()
| The variable pegmatite_pcie_driver references
| the function __init pegmatite_pcie_probe()
| If the reference is valid then annotate the
| variable with __init* or __refdata (see linux/init.h) or name the variable:
| *_template, *_timer, *_sht, *_ops, *_probe, *_probe_one, *_console
```

root cause 如下

```c
static int __init pegmatite_pcie_probe(struct platform_device *pdev)
{
    struct device_node *np = pdev->dev.of_node;
    int i = 0, ret;
    struct phy *phy;
    struct device *dev = &pdev->dev;
    struct device_node *child;

......
    return 0;
}

static int pegmatite_pcie_remove(struct platform_device *pdev) {
    int i;
    pegmatite_pcie_top_t *pcie = platform_get_drvdata(pdev);
    pegmatite_pcie_controller_t *controller;
    struct pci_bus *bus_itr;

    pcie_free_intx_irq_domain();

    bus_itr = pci_find_next_bus(NULL);
    while (bus_itr) {
        pci_stop_root_bus(bus_itr);
        pci_remove_root_bus(bus_itr);
        bus_itr = pci_find_next_bus(NULL);
    }

    for (i=0; i<pcie->nr_controllers; ++i) {
        controller = pcie->controllers[i];

        pci_remove_proc(controller);

        release_resource(&controller->mem);
        irq_dispose_mapping(controller->irq);
        iounmap(controller->pcie_reset_control);
        iounmap(controller->pcie_app_base);
        iounmap(controller->pcie_trgcfg_base);
    }

    return 0;
}

static const struct of_device_id of_pcie_table[] = {
    {.compatible = "marvell,pegmatite-pcie"},
    {},
};
MODULE_DEVICE_TABLE(of, of_pcie_table);

static struct platform_driver pegmatite_pcie_driver = {
    .probe  = pegmatite_pcie_probe,
    .remove = pegmatite_pcie_remove,
    .driver = {
        .owner  = THIS_MODULE,
```

```
        .name   = "pegmatite-pcie",
        .of_match_table = of_match_ptr(of_pcie_table),
    },
};
```

原因就是static int `__init` pegmatite_pcie_probe(struct platform_device *pdev)
in include/linux/init.h

```
#define __init      __section(.init.text) __cold notrace
```

即pegmatite_pcie_probe()的code会被放入".init.text" section，即在kernel初始化结束后会丢弃该section中的code，但
`static struct platform_driver pegmatite_pcie_driver` 是在普通的data section中的，不会被丢弃。
在不会被丢弃的section中引用到初始化后会被丢弃的function，所以link报warning了。

```
CONFIG_DEBUG_SECTION_MISMATCH Kconfig symbol:

The section mismatch analysis checks if there are illegal
references from one section to another section.
Linux will during link or during runtime drop some sections
and any use of code/data previously in these sections will
most likely result in an oops.
In the code functions and variables are annotated with
__init, __devinit etc. (see full list in include/linux/init.h)
which results in the code/data being placed in specific sections.
The section mismatch analysis is always done after a full
kernel build but enabling this option will in addition
do the following:

Add the option -fno-inline-functions-called-once to gcc
When inlining a function annotated __init in a non-init
function we would lose the section information and thus
the analysis would not catch the illegal reference.
This option tells gcc to inline less but will also
result in a larger kernel.
Run the section mismatch analysis for each module/built-in.o
When we run the section mismatch analysis on vmlinux.o we
lose valueble information about where the mismatch was
introduced.
Running the analysis for each module/built-in.o file
will tell where the mismatch happens much closer to the
source. The drawback is that we will report the same
mismatch at least twice.
Enable verbose reporting from modpost to help solving
the section mismatches reported.

```