Linux DM9051 Driver r2502_v3.9 support functions

dm9051.c

To manage various settings and parameters for the DM9051 SPI Fast Ethernet driver. Here is a detailed breakdown of the structure and its components:

Structure Definition

```
struct driver_config
{
	char *release_version;
	int interrupt;
	int mid;
	struct mod_config
	{
		char *test_info;
		u8 encpt_mode; /* encpt_mode */
		u8 encpt_pad;  /* encpt_setted_key */
		int skb_wb_mode;
		int tx_mode;
		int checksuming;
		struct
		{
			int burst_mode;
			size_t tx_blk; /* alignment, software_build_kernel_conf */
			size_t rx_blk;
		} align;
	} mod[MODE_NUM];
};
```

Overview

The struct driver_config is a configuration structure in the DM9051 driver, likely used to store various settings and parameters related to the driver's operation. It includes fields for the release version, interrupt number, nested mod_config structure selection index, and an array of mod_config structures.

Members

1. release_version
    - Type: char *
    - Description: A string representing the release version of the driver.
2. interrupt
    - Type: int
    - Description: An integer representing the interrupt mode. This could be one of several predefined modes such as MODE_POLL, MODE_INTERRUPT, or MODE_INTERRUPT_CLKOUT.
3. mid
    - Type: int

    - Description: An integer representing the mode index. This is used to select a specific configuration from the mod array.

4. mod
    - Type: struct mod_config[MODE_NUM]
    - Description: An array of struct mod_config structures, each representing a different configuration mode. The size of the array is defined by MODE_NUM.

Nested Structure:

The nested struct mod_config is a configuration structure for a specific designated platform.

5. test_info
    - Type: char *
    - Description: A string containing information about the test or configuration.
6. encpt_mode
    - Type: u8
    - Description: An 8-bit unsigned integer representing the encryption mode.
7. encpt_pad
    - Type: u8
    - Description: An 8-bit unsigned integer representing the encryption key padding.
8. int skb_wb_mode

- Type: int
  * An integer representing the SKB (Socket Buffer) write-back mode.

5. tx_mode
    - Type: int
    - Description: An integer representing the transmission mode.
6. checksuming
    - Type: int
    - Description: An integer representing whether checksum offloading is enabled or not.
7. align
    - Type: struct
    - Description: A nested structure containing alignment settings for burst mode.

Nested Nested Structure: align_config

The nested nested anonymous structure containing alignment-related parameters.

8. burst_mode
    - Type: int
    - Description: An integer representing whether burst mode is enabled or not.
9. tx_blk
    - Type: size_t
    - Description: A size_t value representing the alignment block size for transmission.
10. rx_blk
    - Type: size_t
    - Description: A size_t value representing the alignment block size for reception.

Usage

This structure is used to configure various aspects of the DM9051 driver, including encryption settings, skb boundary mode, transmission modes, checksum offloading, and alignment settings for burst mode. The mid member is used to select a specific configuration from the mod array, allowing for different configurations to be easily switched between.

Example

Here is an example of how this structure might be initialized:

```
const struct driver_config confdata = {
    .release_version = "lnx_dm9051_kt6631_r2502_v3.9",
    .interrupt = MODE_INTERRUPT_CLKOUT,
    .mid = MODE_A,
    .mod = {
        {
            .test_info = "Test in rpi5 bcm2712",
            .encpt_mode = FORCE_BUS_ENCPT_FAB_ON,
            .encpt_pad = 0x00,
            .skb_wb_mode = SKB_WB_ON,
            .tx_mode = FORCE_TX_CONTI_OFF,
            .checksuming = DEFAULT_CHECKSUM_ON,
            .align = {
                .burst_mode = BURST_MODE_ALIGN,
                .tx_blk = 32,
                .rx_blk = 64
            }
        },
        // Additional configurations for other modes...
    }
};
```

This example initializes the confdata structure with specific settings for one mode, including encryption, skb boundary mode, transmission mode, checksum offloading, and alignment settings.