ENT-AN1174
Application Note
Wake-on-LAN

November 2016

Microsemi
a Microchip company

Contents

1 Revision History

The revision history describes the changes that were implemented in the document. The changes are listed by revision, starting with the most current publication.

1.1 Revision 1.0

Revision 1.0 was the first publication of this document.

2 Wake-on-LAN

Wake-on-LAN (WoL) is a feature in which a PHY in a very low power state can be awakened by a host using a uniquely formatted packet called a magic packet. This requires decoding a magic packet and alerting a board controller by asserting an interrupt pin or another output pin when a packet match is detected.

The following Microsemi PHY device families support the IEEE 802.3az standard for Energy-Efficient Ethernet and the WoL feature.

- VSC8502 (VSC8501/VSC8502)
- VSC8541 (VSC8530/VSC8531/VSC8540/VSC8541)

This application note provides an overview of WoL. For additional information, consult the Wikipedia and Wireshark articles and AMD‘s white paper. For specific device information, see the VSC8502, VSC8531, and VSC8541 product documentation.

2.1 Overview

Ethernet connections—including home and work networks, wireless data networks, and the Internet itself—are based on packets of data sent between computers. WoL is implemented using a magic packet that is sent to all computers in a network, including the computer to be woken up. The magic packet contains the destination computer's Media Access Control (MAC) address, an identifying number built into each network interface card (NIC) or other Ethernet device in a computer that enables it to be uniquely recognized and addressed on a network. Powered-down or turned off computers capable of WoL will contain network devices able to listen to incoming packets in low-power mode while the system is powered down. If a magic packet is received that is directed to the device's MAC address, the NIC signals the computer's power supply or motherboard to initiate system wake-up, in the same way pressing the power button would.

WoL features include:

- Configurable magic packet
- Configurable magic packet repetition count (ranges from 1 to 16)
- Secure On enable/disable
- Configurable password length

It is a common misconception that because WoL is built upon broadcast technology, it can only be used within the current network subnet. While this is generally the case, there are some exceptions, and WoL can operate across any network, given appropriate configuration and hardware (including remote wake-up across the Internet). While the WoL approach and implementation will only work on WoL-capable hardware, it still will be fully backward-compatible legacy equipment.

2.1.1 Magic Packets

Magic packets are broadcast packets of arbitrary length that can be of any network layer and application protocol type. The only requirement is that the payload section contains 6 bytes of 0xFF as a synchronization stream, followed by the MAC address of the device to be wakened, repeated 16 times. Some applications may utilize a shorter repetition of the MAC address (for instance, 12 repetitions). The location of this data sequence within the payload is not fixed and can occur anywhere. The magic packet is sent on the data link layer (Layer 2 in the OSI model) and when sent, is broadcast to all attached devices on a given network using the network broadcast address—the IP address (Layer 3 in the OSI model) is not used.

2.1.2 Secure-On

To improve network security and reduce the chances of unauthorized access, some vendors provide support for an additional Secure-On feature. When the Secure-On feature is enabled, a 4- or 6-byte password is appended to the magic packet and the NIC wakes up the system only when the MAC address and password match. This security measure significantly decreases the risk of successful brute

force attacks by increasing the search space by 48 bits (6 bytes)—up to 296 combinations if the MAC address is entirely unknown. However, any network eavesdropping will expose the clear-text password, but abuse of the WoL feature only allows computers to be switched on; it does not in itself bypass password and other forms of security.

2.2 Energy and Power Savings

In order for WoL to work, parts of the network interface need to stay on. This consumes a small amount of standby power, much less than normal operating power. Therefore, disabling WoL when not needed can very slightly reduce power consumption on computers that are switched off but still plugged into a power socket.

There are two major areas of power savings on the system level.

- Physical layer devices (PHYs)
- Switches or servers

Potential energy savings at the switch/server level can be realized by shutting down system blocks and even by slowing down the control processors whenever there is no heavy-duty traffic. The WoL standard specifies the protocol used to signal information about the traffic activity status and then take such action to lower energy consumption.

2.3 Interrupt Status

WoL interrupt status is similar to any other PHY interrupt status, and is indicated on the MDINT pin associated with the device, provided that the interrupt masks are set accordingly. The interrupt bits for WoL are defined as follows:

- Interrupt mask, register 25, bit 6 (Reg25.6)
- Interrupt status, register 26, bit 6 (Reg26.6)
- Unified PHY API interrupt event, VTSS_PHY_LINK_WAKE_ON_LAN_INT_EV

There is only one MDINT signal for each PHY device. All enabled interrupts are then OR'd together and the output is indicated on the MDINT pin. The Unified PHY API automatically sets the MDINT mask for output anytime any interrupt mask bit is set. The PHY device could be a single port PHY or a Multi-port PHY. Unless otherwise specified, the Global Interrupt Status register (29G) must be read to determine the port that is the source of the active interrupt causing the MDINT pin to become active.

The WOL interrupt generator module is in parallel with the packet checker. This module monitors the received data for a magic packet sequence. All other configurable parameters are coming to this module from extended page 2 registers. WoL interrupts will be generated only when a magic packet sequence is detected as per the configuration and corresponding CRC status indicates no errors in the packet. The packet checker provides the CRC status to the WoL interrupt generator module.

2.3.1 MDINT_ n Pin Functionality

The interrupt pins on a multi-port PHY have additional requirements that need to be implemented in order to fulfill device specifications. By default, interrupt signals from all PHY ports are combined, then driven onto a single MDINT pin. This applies to all PHYs that support WoL functionality. In this mode of operation, the Global Interrupt Status register (29G) must be read to determine the source port of the active interrupt.

2.3.1.1 MDINT_ n Pin Considerations: VSC8502 Only

For the VSC8502 device, there are individual pins for port 0 (MDINT_0) and for port 1 (MDINT_1), which can be tied together through an external resistor to obtain one common MDINT, if desired. Extended LED Mode 23 is added for displaying the interrupt on any of the LED0/1/2/3 for the corresponding PHY.

Register 27E2 bit 1 controls whether the interrupt signals from both ports are combined and tied together or if the pins indicate interrupts separately on the MDINT_*n* pins. The default state of this register bit is such that interrupts from both ports are combined (see the VSC8502 datasheet for more information). Combining MDINT_0 and MDINT_1 allows a single interrupt pin on the controller to be used to monitor interrupts from both ports. However, when a single pin is used, the Global Interrupt Status register (29G) must be read to determine the port that is the source of the active interrupt. When the separate interrupt

3 WoL Control

The following section describes the two available options for controlling WoL functionality through software: an API or a Linux Driver (VSC8531 and VSC8541 only).

3.1 PHY API

The Microsemi PHY API, available for all Microsemi PHYs, provides software control for the WoL implementation. The API functions use of the following definitions for the WoL implementation.

```
/**< Defines the maximum size Wo0L MAC ADDR, ie. 6 Octets.

*/ #define MAX_WOL_MAC_ADDR_SIZE     6

/**< Defines the maximum size WoL Secure_On Password, ie. 6 Octets.

*/ #define MAX_WOL_PASSWD_SIZE                                  6

/**< Defines the minimum size WoL Secure_On Password, ie. 4 Octets.

*/ #define MIN_WOL_PASSWD_SIZE                                  4


/**

* \brief Structure for Wake-On-LAN Password Length configuration

*/

typedef enum {

VTSS_WOL_PASSWD_LEN_4 = 4,

/**< PasswdLen=4 bytes

*/ VTSS_WOL_PASSWD_LEN_6 = 6,

/**< PasswdLen=6 bytes */

} vtss_wol_passwd_len_type_t;


/**

* \brief Structure for Wake-On-LAN MAC Address

**< Wake-On-LAN MAC Address - 6 bytes */

*/

typedef struct

{

u8 addr[MAX_WOL_MAC_ADDR_SIZE];

} vtss_wol_mac_addr_t;


/**
```

```
* \brief Structure for Wake-On-LAN Secure-On Password

**< Secure-On Password, Can be 4 or 6 bytes */

*/

typedef struct

{

u8 passwd[MAX_WOL_PASSWD_SIZE];

} vtss_secure_on_passwd_t;


/**

* \brief Structure for Get/Set Wake-On-LAN configuration

*/

typedef struct {

/**< Enable/Disable for Secure-On Password

*/ BOOL                          secure_on_enable;

/**< Wake-On-LAN MAC Addr Definition vtss_wol_mac_addr_t     wol_mac;

*/

/**< Wake-On-LAN Password Definition

*/ vtss_secure_on_passwd_t     wol_pass;   vtss_wol_passwd_len_type_t

/**< Enumeration for Password Length options */ wol_passwd_len;

/**< Magic Packet Repetition Count (1-16 is valid) */ u16      magic_pkt_cnt;




} vtss_phy_wol_conf_t;
```

3.1.1 Enabling WoL Support

The Microsemi PHY API provides the following function to enable or disable the WoL function in the software. This function configures the interrupts masks as specified by the enable Boolean by setting or clearing the interrupt event VTSS_PHY_LINK_WAKE_ON_LAN_INT_EV. When any interrupt event is set, the MDINT mask is automatically enabled by the software.

```
/**

*  \brief function to Enable or Disable WoL by enabling or disabling the interrupt

*  \param inst      [IN] Target instance reference.

*  \param port_no   [IN] Port in question

*  \param enable   [IN] Boolean, Enable=TRUE or 1, Disable=False or 0

*  \return Return code. VTSS_RC_OK if no errors.
```

```
*/

vtss_rc vtss_phy_wol_enable(const vtss_inst_t                inst,

const vtss_port_no_t port_no, const BOOL enable);
```

3.1.2 Retrieving the WoL Configuration

The API provides the following function to retrieve the PHY WoL configuration in the software.

```
/**

*  \brief function to Get Wake-On-LAN configuration

*  \param inst      [IN] Target instance reference.

*  \param port_no   [IN] Port in question

*  \param conf     [IN] Ptr to WoL Structure vtss_phy_wol_conf_t to be filled out by the /

*  \return Return code. VTSS_RC_OK if no errors.

*/

vtss_rc vtss_phy_wol_conf_get(const vtss_inst_t                inst,

const vtss_port_no_t port_no, const vtss_phy_wol_conf_t *const conf);
```

3.1.3 Setting the PHY WoL Configuration

The API provides the following software function to set the PHY WoL configuration.

```
/ * \brief function to Set Wake-On-LAN configuration

*  \param inst      [IN] Target instance reference.

*  \param port_no   [IN] Port in question

*  \param conf      [IN] Ptr to WoL Structure vtss_phy_wol_conf_t filled out by User

*  \return Return code. VTSS_RC_OK if no errors.

*/

vtss_rc vtss_phy_wol_conf_set(const vtss_inst_t              inst,

const vtss_port_no_t port_no, const vtss_phy_wol_conf_t *const conf);
```

3.1.4 Setting the PHY Interrupt Configuration

The API provides the following software function to set the PHY interrupt configuration for the event

VTSS_PHY_LINK_WAKE_ON_LAN_INT_EV.

```
*  \brief Enabling / Disabling of events

*

*  \param inst [IN]           Target instance reference.

*  \param port_no [IN]     Port number

*  \param ev_mask [IN]   Mask containing events that are enabled/disabled

*  \param enable [IN]      Enable/disable of event
```

```
*

*  \return Return code.

**/


vtss_rc vtss_phy_event_enable_set(const vtss_inst_t          inst,

const vtss_port_no_t port_no, const vtss_phy_event_t ev_mask, const BOOL enable);
```

3.1.5 Polling for the PHY Interrupt Status

The API provides the following software function to poll for the PHY interrupt configuration for the event VTSS_PHY_LINK_WAKE_ON_LAN_INT_EV (for more information, see the Global Interrupt Status Register, Reg29G).

```
/**

*  \brief Polling for active events

*

*  \param inst [IN]            Target instance reference.

*  \param port_no [IN]      Port number

*  \param ev_mask [OUT]  Mask containing events that are active

*

*  \return Return code.

**/

vtss_rc vtss_phy_event_poll(const vtss_inst_t           inst,

const vtss_port_no_t port_no, vtss_phy_event_t *const ev_mask);
```

3.1.6 WoL Ethernet Packets Samples

The following sections provide sample magic packet descriptions for WoL using the PHY API functions.

3.1.6.1 PHY WoL Config and Ethernet Packet Description

```
64 Byte Sample Ethernet Packet:

Dest MAC Addr: FF FF FF FF FF FF

Src MAC Addr: AA AA AA AA AA A1

EthType: 08 00

WoL Magic Packet Repetition Count: 6 WoL MAC Addr: 00 00 00 00 00 55

WoL Password: 00 00 00 00 00 00
```

FF	FF	FF	FF	FF	FF	AA	AA	AA	AA	AA	A1	08	00	FF	FF
FF	FF	FF	FF	00	00	00	00	00	55	00	00	00	00	00	55
00	00	00	00	00	55	00	00	00	00	00	55	00	00	00	00
00	55	00	00	00	00	00	55	00	00	00	00				

3.1.6.2 PHY WoL Config and Ethernet Packet Description with WoL Password

64 Byte Sample Ethernet Packet:

Dest MAC Addr: FF FF FF FF FF FF

Src MAC Addr: AA AA AA AA AA A1

EthType: 08 00

WoL Magic Packet Repetition Count: 5 WoL MAC Addr: 00 00 00 00 00 55

WoL Password: 98 76 54 32 10 99

FF	FF	FF	FF	FF	FF	AA	AA	AA	AA	AA	A1	08	00	FF	FF
FF	FF	FF	FF	00	00	00	00	00	55	00	00	00	00	00	55
00	00	00	00	00	55	00	00	00	00	00	55	00	00	00	00
00	55	98	76	54	32	10	99	00	00	00	00				

3.2 Linux Driver

For the VSC8541 family of devices (VSC8530/VSC8531/VSC8540/VSC8541), Microsemi has provided a Linux Driver to the open source community that provides functionality to control WoL. This driver controls WoL implementation in the Linux kernel, which can be accessed through the Linux OS.

Currently, the WoL functionality through Linux is only available for the VSC8531 and VSC8541 devices, and can be found in Linux open source drivers revision 4.9 and later. Linux support for other PHYs may occur in later releases.

Note: Linux version 4.9 is scheduled to be available in December 2016, while Linux 4.10 is scheduled to be available in February 2017. The current implementation can be retrieved from https://git.kernel.org/cgit/linux/kernel/git/torvalds/linux.git/tree/drivers/net/phy/mscc.c?id=v4.9-rc1. After the initial release of Linux4.9, the software can be found in future releases at https://www.kernel.org.

Microsemi Headquarters
One Enterprise, Aliso Viejo,
CA 92656 USA
Within the USA: +1 (800) 713-4113
Outside the USA: +1 (949) 380-6100
Sales: +1 (949) 380-6136
Fax: +1 (949) 215-4996
Email: sales.support@microsemi.com
www.microsemi.com

VPPD-03834