Actel SmartFusion™ MSS Ethernet MAC Driver User's Guide

Version 2.0

Actel
POWER MATTERS

Actel Corporation, Mountain View, CA 94043

Printed in the United States of America

Part Number: 50200197-1

Release: February 2010

Trademarks

Table of Contents

Introduction

The SmartFusion™ microcontroller subsystem (MSS) includes a high-speed media access control (MAC) Ethernet controller. This software driver provides a set of functions for controlling the MSS Ethernet MAC as part of a bare metal system where no operating system is available. This driver can be adapted for use as part of an operating system but the implementation of the adaptation layer between this driver and the operating system's driver model is outside the scope of this driver.

The MSS Ethernet MAC is an AHB bus master on the SmartFusion MSS AHB bus matrix. The built-in DMA controller inside the MAC block, along with the AHB master interface, is used to automatically move data between external RAM and the built-in transmit FIFO and receive FIFO with minimal CPU intervention.

The MSS Ethernet MAC automatically fetches from transmit data buffers and stores the receive data buffers in external RAM. Internal memory is used as configurable FIFO memory blocks; both RX/TX memory blocks are available. The blocks are called descriptors, and both RX/TX are referenced within the driver.

Features

The MSS Ethernet MAC driver provides the following features:

- Support for initializing and configuring the MSS Ethernet MAC
- Support for initializing and configuring the external PHY
- Support for controlling the MSS Ethernet MAC and data packet transfers

The MSS Ethernet MAC driver is provided as C source code.

Supported Hardware IP

The MSS Ethernet MAC bare metal driver can be used with Actel's MSS_MAC IP version 0.6 or higher included in the SmartFusion MSS.

The supported PHY in the driver isDP83484C.

Files Provided

The files provided as part of the MSS Ethernet MAC driver fall into three main categories: documentation, driver source code, and example projects. The driver is distributed via the Actel Firmware Catalog, which provides access to the documentation for the driver, generates the driver's source files into an application project, and generates example projects that illustrate how to use the driver. The Actel Firmware Catalog is available from the Actel web site: www.actel.com/products/software/firmwarecat/default.aspx.

Documentation

The Actel Firmware Catalog provides access to these documents for the driver:

- User's guide (this document)
- A copy of the license agreement for the driver source code
- Release notes

Driver Source Code

The Actel Firmware Catalog generates the driver's source code into the *drivers\mss_ethernet_mac* subdirectory of the selected software project directory. The files making up the driver are detailed below.

mss_ethernet_mac.h

This header file contains the public application programming interface (API) of the MSS Ethernet MAC software driver.

This file should be included in any of your application's C source files that use the MSS Ethernet MAC software driver.

mss_ethernet_mac.c

This C source file contains the implementation of the MSS Ethernet MAC software driver.

mss_ethernet_mac_regs.h

This header file contains MSS Ethernet MAC register definitions.

This header file does not need to be included in the application's C source files. It is only used within the MSS Ethernet MAC driver implementation.

mss_ethernet_mac_conf.h

This header file contains definitions used to configure the MSS Ethernet MAC software driver.

This header file does not need to be included in the application's C source files. It is only used within the MSS Ethernet MAC driver implementation.

mss_ethernet_mac_desc.h

This header file contains definitions used for the RX/TX descriptors.

This header file does not need to be included in the application's C source files. It is only used within the MSS Ethernet MAC driver implementation.

mss_ethernet_mac_user_cfg.h

This header file contains definitions generated from the configuration settings entered on the MSS Ethernet MAC driver configuration dialog. The Firmware Catalog opens this configuration dialog when you generate the Ethernet MAC driver.

This header file does not need to be included in the application's C source files. It is only used within the MSS Ethernet MAC driver implementation.

crc32.h

This header file contains prototypes for the CRC32 calculations.

This header file does not need to be included in the application's C source files. It is only used within the MSS Ethernet MAC driver implementation.

crc32.c

This file contains the structure for the CRC32 calculations and also the function for calculating the 32-bit CRC value of given data.

phy.h

This header file contains the application programming interface for controlling the external PHY used with the MSS Ethernet MAC. The driver uses these functions internally and they are not normally exposed as part of the driver's public API.

However, these functions can be made available for debug purposes by including the *drivers/mss_ethernet_mac/phy.h* file in your application's C source files.

phy.c

This C source file contains the implementation of the PHY initialization, configuration, and control functions of the MSS Ethernet MAC software driver. This file contains the API for the DP83848C external PHY, which has been tested with MSS Ethernet MAC.

The following functions are provided:

- *PHY_reset()*
- *PHY_auto_negotiate()*
- *PHY_probe()*
- *PHY_link_status()*
- *PHY_link_type()*
- *PHY_set_link_type()*
- *PHY_set_loopback()*

Example Code

The Actel Firmware Catalog provides access to example projects illustrating the use of the driver. Each example project is self contained and is targeted at a specific processor and software toolchain combination. The example projects are targeted at the FPGA designs in the hardware development tutorials supplied with Actel's development boards. The tutorial designs may be found on the Actel Development Kit web page (www.actel.com/products/hardware).

Driver Deployment

This driver is intended to be deployed from Actel Firmware Catalog into a software project by generating the driver's source files into the project directory. The driver uses the SmartFusion Cortex Microcontroller Software Interface Standard – Peripheral Access Layer (CMSIS-PAL) to access MSS hardware registers. You must ensure that the SmartFusion CMSIS-PAL is either included in the software toolchain used to build your project or is included in your project. The most up-to-date SmartFusion CMSIS-PAL files can be obtained using the Actel Firmware Catalog.

The following example shows the intended directory structure for a SoftConsole ARM® Cortex™-M3 project targeted at the SmartFusion MSS. This project uses the MSS Ethernet MAC and MSS Watchdog drivers. Both of these drivers rely on SmartFusion CMSIS-PAL for accessing the hardware. The contents of the *drivers* directory result from generating the source files for each driver into the project. The contents of the *CMSIS* directory result from generating the source files for the SmartFusion CMSIS-PAL into the project.

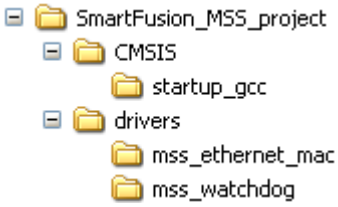

Figure 1 · SmartFusion MSS Project Example

When generating the MSS Ethernet MAC driver, the following configuration screen is presented. You must enter configuration settings appropriate to your hardware design here.

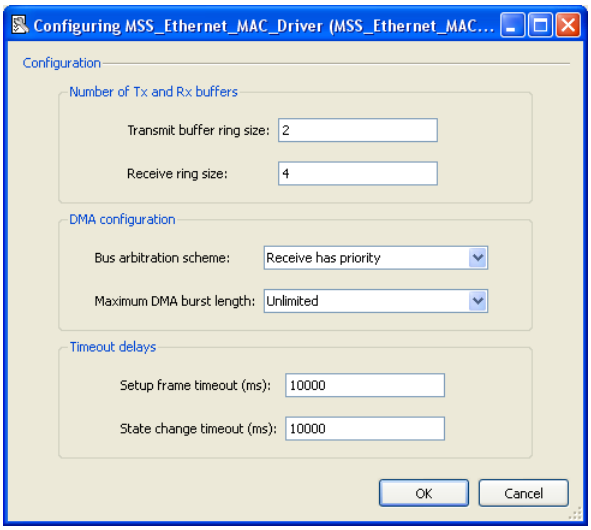

Figure 2 · Firmware Catalog Driver Configuration Options

- Transmit buffer ring size: Number of transmit buffers
- Receive buffer ring size: Number of receive buffers

- Bus arbitration scheme: Allocate the RX/TX priority to access the bus
- Maximum DMA burst length: Specifies the maximum number of words that can be transferred within one DMA transaction. When "Unlimited" the bursts are only limited by the threshold levels of the MAC's internal FIFOs.
- Setup frame timeout: Timeout value in milliseconds for sending setup frame
- State change timeout: Timeout value in milliseconds for state change of TX / RX processes

Firmware Catalog will generate definitions based on the settings entered on this dialog into the driver's *mss_ethernet_mac_user_cfg.h* file.

Driver Configuration

The configuration of all features of the MSS Ethernet MAC is covered by this driver with the exception of the SmartFusion IOMUX configuration. SmartFusion allows multiple non-concurrent uses of some external pins through IOMUX configuration. This feature allows optimization of external pin usage by assigning external pins for use by either the microcontroller subsystem or the FPGA fabric. The MSS Ethernet MAC RMII signals are routed through IOMUXes to the SmartFusion device external pins. These IOMUXes are automatically configured correctly by the MSS Configurator in the hardware flow when the MSS Ethernet MAC is enabled in that tool. You must ensure that the MSS Ethernet MAC is enabled by the MSS Configurator in the hardware flow; otherwise the RMII signals will not be connected to the chip's external pins. For more information on IOMUX, refer to the IOMUX section of the SmartFusion Microcontroller Susbsystem (MSS) User's Guide.

The base address, register addresses and interrupt number assignment for the MSS Ethernet MAC are defined as constants in the SmartFusion CMSIS-PAL. You must ensure that the SmartFusion CMSIS-PAL is either included in the software tool chain used to build your project or is included in your project.

Application Programming Interface

This section describes the driver's API. The functions and related data structures described in this section are used by the application programmer to control the MSS Ethernet MAC peripheral from the user's application.

Theory of Operation

The MSS Ethernet MAC driver functions are grouped into the following categories:

- Initialization
- Configuration
- MAC control and data packet transfer
- PHY control

Initialization

The MSS Ethernet MAC driver is initialized through a call to the *MSS_MAC_init()* function. The *MSS_MAC_init()* function also initializes the PHY.

The *MSS_MAC_init()* function must be called before any other MSS Ethernet MAC driver functions can be called.

Configuration

The functions available for configuring the MSS Ethernet MAC and the PHY are as follows:

- *MSS_MAC_configure()*
- *MSS_MAC_get_configuration()*

MAC Control and Data Packet Transfer

The functions available for controlling the MSS Ethernet MAC and data packet transfer are as follows:

- *MSS_MAC_tx_packet()*
- *MSS_MAC_rx_packet()*
- *MSS_MAC_rx_packet_ptr_set()*
- *MSS_MAC_prepare_rx_descriptor()*

PHY Control

The driver provides a set of functions to initialize, configure and control the PHY. The driver uses these functions internally and they are not normally exposed in the driver's public API. However they can be made available for debug purposes by including the *drivers/mss_ethernet_mac/phy.h* file in your application's C source files. The functions available for controlling the PHY are as follows:

- *PHY_reset()*
- *PHY_auto_negotiate()*
- *PHY_probe()*
- *PHY_link_status()*
- *PHY_link_type()*
- *PHY_set_link_type()*
- *PHY_set_loopback()*

Types

There are no MSS Ethernet MAC driver specific types.

Constant Values

MSS_PHY_ADDRESS_AUTO_DETECT

The MSS_PHY_ADDRESS_AUTO_DETECT constant is used as the *phy_address* parameter to the *MSS_MAC_init()* function when you do not know the PHY address.

MSS_MAC_BLOCKING

The MSS_MAC_BLOCKING constant is used as the *time_out* parameter to the *MSS_MAC_tx_packet()*, *MSS_MAC_rx_packet()* and *MSS_MAC_rx_packet_ptrset()* functions. When this constant is used, the called function will not timeout and will wait for the packet transmission to complete.

MSS_MAC_NONBLOCKING

The MSS_MAC_NONBLOCKING constant is used as the *time_out* parameter to the *MSS_MAC_tx_packet()*, *MSS_MAC_rx_packet()* and *MSS_MAC_rx_packet_ptrset()* functions. When this constant is used, the called function will not wait for a timeout period if the requested operation cannot be completed immediately.

MAC Configuration Parameters

These constants are configuration parameters for the MSS Ethernet MAC operation mode register (CSR6).

A logical OR of these configuration parameters is used build the *configuration* parameter for the *MSS_MAC_configure()* function. These configuration parameters can also be used as bitmasks with the return value of the *MSS_MAC_get_configuration()* function to read the current configuration of the MAC.

Note: Refer to the SmartFusion Handbook for details of the MSS Ethernet MAC operation mode register (CSR6).

Constant	Description
MSS_MAC_CFG_RECEIVE_ALL	MAC configuration parameter
MSS_MAC_CFG_TRANSMIT_THRESHOLD_MODE	MAC configuration parameter
MSS_MAC_CFG_STORE_AND_FORWARD	MAC configuration parameter
MSS_MAC_CFG_THRESHOLD_CONTROL_00	MAC configuration parameter
MSS_MAC_CFG_THRESHOLD_CONTROL_01	MAC configuration parameter
MSS_MAC_CFG_THRESHOLD_CONTROL_10	MAC configuration parameter
MSS_MAC_CFG_THRESHOLD_CONTROL_11	MAC configuration parameter
MSS_MAC_CFG_FULL_DUPLEX_MODE	MAC configuration parameter

Constant	Description
MSS_MAC_CFG_PASS_ALL_MULTICAST	MAC configuration parameter
MSS_MAC_CFG_PROMISCUOUS_MODE	MAC configuration parameter
MSS_MAC_CFG_INVERSE_FILTERING	MAC configuration parameter
MSS_MAC_CFG_PASS_BAD_FRAMES	MAC configuration parameter
MSS_MAC_CFG_HASH_ONLY_FILTERING_MODE	MAC configuration parameter
MSS_MAC_CFG_HASH_PERFECT_RECEIVE_FILTERING_MODE	MAC configuration parameter

Table 1 • MAC Configuration Parameters

Data Structures

There are no MSS Ethernet MAC driver specific data structures.

Functions

MSS_MAC_init

Prototype

```
void MSS_MAC_init
(
  uint8_t phy_address
);
```

Description

The *MSS_MAC_init()* function is used to reset and initialize the MSS Ethernet MAC hardware and driver internal data to a default configuration state. This function also resets and initializes the PHY to a default configuration state.

Parameters

phy_address

The *phy_address* parameter specifies the address of the PHY device, set in hardware by the address pins of the PHY device. The *phy_address* parameter can be a value from 0 to 31.

If you do not know the PHY address, you must use MSS_PHY_ADDRESS_AUTO_DETECT as the *phy_address* parameter, and the driver will interrogate the PHY for its address.

Return Value

This function does not return a value.

MSS_MAC_configure

Prototype

```
void MSS_MAC_configure
(
  uint32_t configuration
)
```

Description

The function MSS_MAC_*configure()* is used to configure the operation mode of MSS Ethernet MAC.

Parameters

configuration

The *configuration* parameter specifies the desired operation mode configuration for the MSS Ethernet MAC.

The driver defines a set of configuration parameters for the MSS Ethernet MAC operation mode register (CSR6). The *configuration* parameter is specified as the logical OR of the following configuration parameters:

- MSS_MAC_CFG_RECEIVE_ALL
- MSS_MAC_CFG_TRANSMIT_THRESHOLD_MODE
- MSS_MAC_CFG_STORE_AND_FORWARD
- MSS_MAC_CFG_THRESHOLD_CONTROL_[00,01,10,11]
- MSS_MAC_CFG_FULL_DUPLEX_MODE
- MSS_MAC_CFG_PASS_ALL_MULTICAST
- MSS_MAC_CFG_PROMISCUOUS_MODE
- MSS_MAC_CFG_PASS_BAD_FRAMES

Note: The *MSS_MAC_get_configuration()* function can be used to retrieve the current MAC configuration.

Return Value

This function does not return a value.

MSS_MAC_get_configuration

Prototype

```
int32_t MSS_MAC_get_configuration
(
  void
);
```

Description

The *MSS_MAC_get_configuration()* function is used to read the current configuration of the MSS Ethernet MAC.

Parameters

This function takes no parameters.

Return Value

The *MSS_MAC_get_configuration()* function returns the configuration of the MSS Ethernet MAC as a 32-bit integer.

The 32-bit integer return value is a logical OR of the following configuration parameter bitmasks:

- MSS_MAC_CFG_RECEIVE_ALL
- MSS_MAC_CFG_TRANSMIT_THRESHOLD_MODE
- MSS_MAC_CFG_STORE_AND_FORWARD
- MSS_MAC_CFG_THRESHOLD_CONTROL_[00,01,10,11]
- MSS_MAC_CFG_FULL_DUPLEX_MODE
- MSS_MAC_CFG_PASS_ALL_MULTICAST
- MSS_MAC_CFG_PROMISCUOUS_MODE
- MSS_MAC_CFG_INVERSE_FILTERING
- MSS_MAC_CFG_PASS_BAD_FRAMES
- MSS_MAC_CFG_HASH_ONLY_FILTERING_MODE
- MSS_MAC_CFG_HASH_PERFECT_RECEIVE_FILTERING_MODE

MSS_MAC_tx_packet

Prototype

```
int32_t MSS_MAC_tx_packet
(
  uint8_t * pacData,
  uint16_t pacLen,
  uint32_t time_out
)
```

Description

The *MSS_MAC_tx_packet()* function is used to send a packet to the MSS Ethernet MAC. This function writes *pacLen* bytes of the packet contained in *pacData* into the transmit FIFO and then activates the transmitter for this packet. If space is available in the FIFO, the function will return once *pacLen* bytes of the packet have been placed into the FIFO and the transmitter has been started. This function will not wait for the transmission to complete. If space is not available in FIFO, the function will keep trying until *time_out* expires. The function will wait for the transmission to complete when the *time_out* parameter is set to MSS_MAC_BLOCKING.

Parameters

pacData

The *pacData* parameter is a pointer to the packet data to be transmitted.

pacLen

The *pacLen* parameter is the number of bytes in the packet to be transmitted.

time_out

The *time_out* parameter is the timeout value for the transmission in milliseconds. The *time_out* parameter value can be one of the following values:

- Unsigned integer greater than 0 and less than 0x01000000
- MSS_MAC_BLOCKING – there will be no timeout.
- MSS_MAC_NONBLOCKING – the function will return immediately if the MSS Ethernet MAC does not have any available transmit descriptor. This would happen when several packets are already queued into the MSS Ethernet MAC transmit descriptor FIFO.

Return Value

The function returns zero if a timeout occurs otherwise it returns size of the packet.

MSS_MAC_rx_packet

Prototype

```
int32_t MSS_MAC_rx_packet
(
  uint8_t *pacData,
  uint16_t pacLen,
  uint32_t time_out
);
```

Description

The *MSS_MAC_rx_packet()* function is used to read a packet from the receive FIFO of the MSS Ethernet MAC and place it into *pacData*. If the *time_out* parameter is MSS_MAC_NONBLOCKING the function will return immediately after the copy operation if data is available. Otherwise the function will keep trying to read until the time out expires or data is read. The function will wait for the transmission to complete when the *time_out* parameter is set to MSS_MAC_BLOCKING.

Parameters

pacData

The *pacData* parameter is a pointer to the buffer where received packet data will be copied. Memory for the buffer should be allocated prior to calling this function. The packet data is copied from the RX descriptors to this buffer.

pacLen

The *pacLen* parameter is the size in bytes of the *pacData* buffer where the received data will be copied.

Time_out

The *time_out* parameter is the timeout value for the transmission in milliseconds. The *time_out* parameter value can be one of the following values:

- Unsigned integer greater than 0 and less than 0x01000000.
- MSS_MAC_BLOCKING – there will be no timeout. The function will only return when a packet has been received.
- MSS_MAC_NONBLOCKING – the function will return immediately if no packets have been received.

Return Value

The function returns the size of the packet if the packet fits in *pacData*. Returns 0 if there is no received packet.

MSS_MAC_rx_packet_ptrset

Prototype

```
int32_t MSS_MAC_rx_packet_ptrset
(
  uint8_t **pacData,
  uint32_t time_out
);
```

Description

The *MSS_MAC_rx_packet_ptrset()* function is very similar to the *MSS_MAC_rx_packet()* function, in that it receives data from the MSS Ethernet MAC. The difference is that it sets *pacData* to point to the memory buffer where the MSS Ethernet MAC copied the received packet instead of copying the received packet into a buffer provided by the application. After this function is called and data is used by the user application or copied to another buffer, the *MSS_MAC_prepare_rx_descriptor()* function must be called to free up the receive memory buffer used by the MSS Ethernet MAC.

Parameters

pacData

The *pacData* parameter is a pointer to a memory buffer pointer. The uint8_t pointer pointed to by the *pacData* parameter will contain the address of the memory buffer containing the received packet after this function returns. The value of *pacData* is only valid if the return value is larger than zero, indicating that a packet was received.

Time_out

The *time_out* parameter is the timeout value for the transmission in milliseconds. The *time_out* parameter value can be one of the following values:

- Unsigned integer greater than 0 and less than 0x01000000.
- MSS_MAC_BLOCKING – there will be no timeout.
- MSS_MAC_NONBLOCKING – the function will return immediately if no packets have been received.

Return Value

The function returns the size of the packet if the packet fits in *pacData*. Returns zero if there is no received packet.

Example

```
int32_t pckt_size;
uint8_t * rx_pckt_pointer;
pckt_size = MSS_MAC_rx_packet_ptrset( &rx_pckt_pointer, MSS_MAC_NONBLOCKING );
if ( pckt_size > 0 )
{
    uint8_t pckt_first_byte;
    uint8_t pckt_second_byte;
    pckt_first_byte = rx_pckt_pointer[0];
    pckt_second_byte = rx_pckt_pointer[1];
    process_first_two_bytes( pckt_first_byte,pckt_second_byte );
}
```

MSS_MAC_prepare_rx_descriptor

Prototype

```
void MSS_MAC_prepare_rx_descriptor
(
  void
);
```

Description

The *MSS_MAC_prepare_rx_descriptor()* function prepares the RX descriptor for receiving packets.

Parameters

This function takes no parameters.

Return Value

This function does not return a value.

Product Support

Actel backs its products with various support services including Customer Service, a Customer Technical Support Center, a web site, an FTP site, electronic mail, and worldwide sales offices. This appendix contains information about contacting Actel and using these support services.

Customer Service

Contact Customer Service for non-technical product support, such as product pricing, product upgrades, update information, order status, and authorization.

From Northeast and North Central U.S.A., call **650.318.4480**
From Southeast and Southwest U.S.A., call **650. 318.4480**
From South Central U.S.A., call **650.318.4434**
From Northwest U.S.A., call **650.318.4434**
From Canada, call **650.318.4480**
From Europe, call **650.318.4252** or **+44 (0) 1276 401 500**
From Japan, call **650.318.4743**
From the rest of the world, call **650.318.4743**
Fax, from anywhere in the world **650. 318.8044**

Actel Customer Technical Support Center

Actel staffs its Customer Technical Support Center with highly skilled engineers who can help answer your hardware, software, and design questions. The Customer Technical Support Center spends a great deal of time creating application notes and answers to FAQs. So, before you contact us, please visit our online resources. It is very likely we have already answered your questions.

Actel Technical Support

Visit the Actel Customer Support website (http://www.actel.com/support/search/default.aspx) for more information and support. Many answers available on the searchable web resource include diagrams, illustrations, and links to other resources on the Actel web site.

Website

You can browse a variety of technical and non-technical information on Actel's home page, at http://www.actel.com/.

Contacting the Customer Technical Support Center

Highly skilled engineers staff the Technical Support Center from 7:00 A.M. to 6:00 P.M., Pacific Time, Monday through Friday. Several ways of contacting the Center follow:

Email

You can communicate your technical questions to our email address and receive answers back by email, fax, or phone. Also, if you have design problems, you can email your design files to receive assistance. We constantly monitor the email account throughout the day. When sending your request to us, please be sure to include your full name, company name, and your contact information for efficient processing of your request.

The technical support email address is tech@actel.com.

Phone

Our Technical Support Center answers all calls. The center retrieves information, such as your name, company name, phone number and your question, and then issues a case number. The Center then forwards the information to a queue where the first available application engineer receives the data and returns your call. The phone hours are from 7:00 A.M. to 6:00 P.M., Pacific Time, Monday through Friday. The Technical Support numbers are:

650.318.4460
800.262.1060

Customers needing assistance outside the US time zones can either contact technical support via email (tech@actel.com) or contact a local sales office. Sales office listings can be found at www.actel.com/company/contact/default.aspx.

Actel is the leader in low-power FPGAs and mixed-signal FPGAs and offers the most comprehensive portfolio of system and power management solutions. Power Matters. Learn more at http://www.actel.com .

Actel Corporation *• 2061 Stierlin Court • Mountain View, CA 94043 • USA*
Phone 650.318.4200 • Fax 650.318.4600 • Customer Service: 650.318.1010 • Customer Applications Center: 800.262.1060

Actel Europe Ltd*. • River Court, Meadows Business Park • Station Approach, Blackwater • Camberley Surrey GU17 9AB • United Kingdom*
Phone +44 (0) 1276 609 300 • Fax +44 (0) 1276 607 540

Actel Japan *• EXOS Ebisu Building 4F • 1-24-14 Ebisu Shibuya-ku • Tokyo 150 • Japan*
Phone +81.03.3445.7671 • Fax +81.03.3445.7668 • http://jp.actel.com

Actel Hong Kong *• Room 2107, China Resources Building • 26 Harbour Road • Wanchai • Hong Kong*
Phone +852 2185 6460 • Fax +852 2185 6488 • www.actel.com.cn

50200197-1/2.10