AN-1554
APPLICATION NOTE

One Technology Way • P.O. Box 9106 • Norwood, MA 02062-9106, U.S.A. • Tel: 781.329.4700 • Fax: 781.461.3113 •
www.analog.com

Pairing the fido5100 and fido5200 REM Switches with a Host and Network Processor

by Matteo Crosio

INTRODUCTION

The use of industrial Ethernet devices is becoming incumbent in industrial automation and motion control applications, replacing legacy field bus connections such as the CANbus or the RS-485.

A high degree of determinism and reliability is needed to handle real-time applications, and many protocols have been introduced by major original equipment manufacturers (OEMs), such as Profinet, EtherCAT, Ethernet/IP, Sercos, and Modbus TCP. Compatibility with those protocols is an important consideration in every industrial communication design, as well as the possibility of needing updating for compatibility with future enhancements like time sensitive networking (TSN). Analog Devices, Inc., supports the most common industrial Ethernet protocols with real-time Ethernet multiprotocol (REM) switches, fido5100 and fido5200. This switch, paired with the fido1100 communication controller, enhance the programmable multiprotocol RapID platform. The fido5100 and fido5200 REM switches offer a precertified solution for Profinet IRT, EtherCAT, Ethernet/IP, Modbus TCP, and Powerlink.

This application note describes the host interface of the fido5100 and fido5200 REM switches with any microprocessor.

The memory bus interface is described, along with the additional signals needed to control the Ethernet interface.

In this application note, active low signals are identified by use of an overbar following the name of the signal (for example, $\overline{\text{RESET}}$).

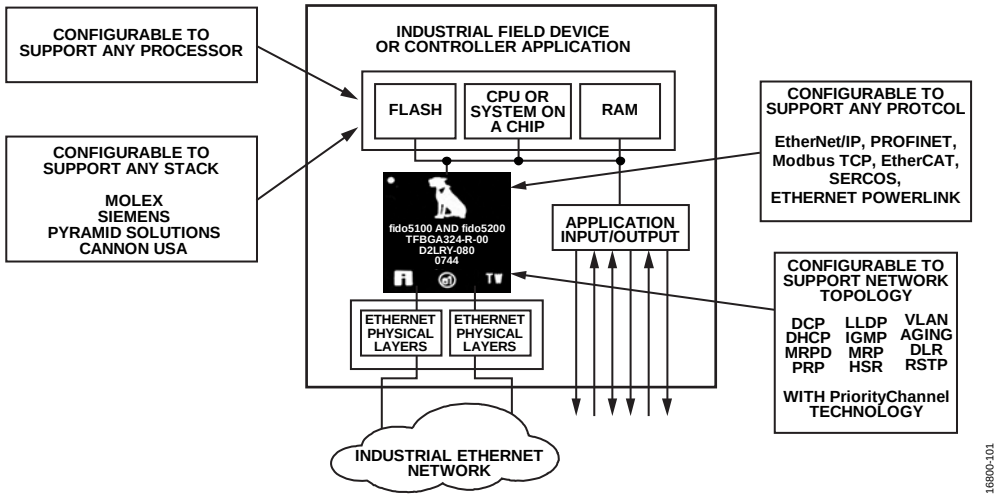

Figure 1. Industrial Ethernet Device Architecture Based on fido5100 and fido5200 and Supported Software Stacks

TABLE OF CONTENTS

REVISION HISTORY

6/2018—Revision 0: Initial Version

HOST INTERFACE

The fido5100 and fido5200 REM switches connect to a communication processor with a host interface designed as a standard asynchronous memory bus port.

The fido5100 and fido5200 REM switches have two Ethernet interfaces each, which support a media independent interface (MII) or reduced media independent interface (RMII). Ethernet physical layers are intentionally left out of the switch itself because of the different requirements on Ethernet physical layer performance.

The general architecture is shown in Figure 2.

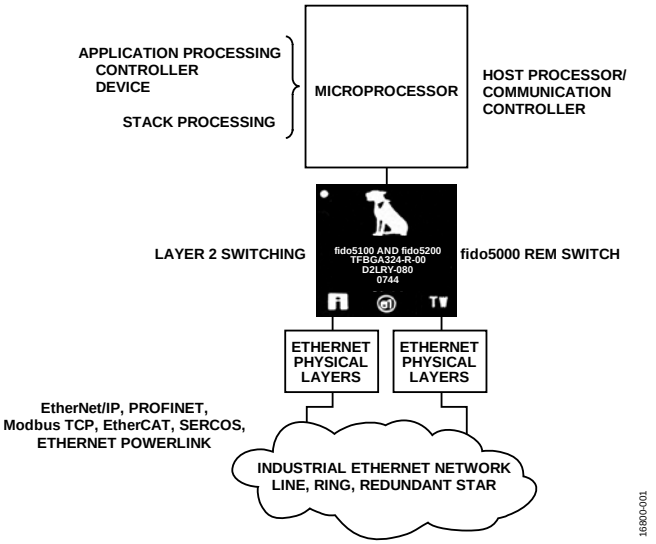

Figure 2. General Architecture

The host interface is either a 16-bit or 32-bit memory bus, with the possibility to multiplex address and data, reducing pin count.

Multiplex bus select (MBS) and data bus size (SIZE_32) signals select this bus mode. MBS and SIZE_32 are sampled on the rising edge of $\overline{\text{RESET}}$ signal.

Another aspect to consider is endianness, which is determined by the little endianness (LE) signal. The LE level is also sampled with the rising edge of $\overline{\text{RESET}}$.

The switch data bus is defined as follows:

- D0 is the least significant bit (LSB).
- D15 is the most significant bit (MSB) for 16-bit bus.
- D31 is the MSB for 32-bit bus.

All control and status registers are 16 bits wide, so that even using a 32-bit bus, data must be transferred in the D15 to D0 order. For more details on how to handle endianness and for a detailed pin function descriptions, refer to the fido5100/ fido5200 data sheet.

There are four bits of data for the address bus, giving access to 16 direct address registers. In case of a nonmultiplexed data bus (such as MBS = 0), the user must consider four additional pins for the address.

It is important to view the timing diagrams (see the fido5100/fido5200 data sheet), which show read and write operations for both nonmultiplexed and multiplexed mode.

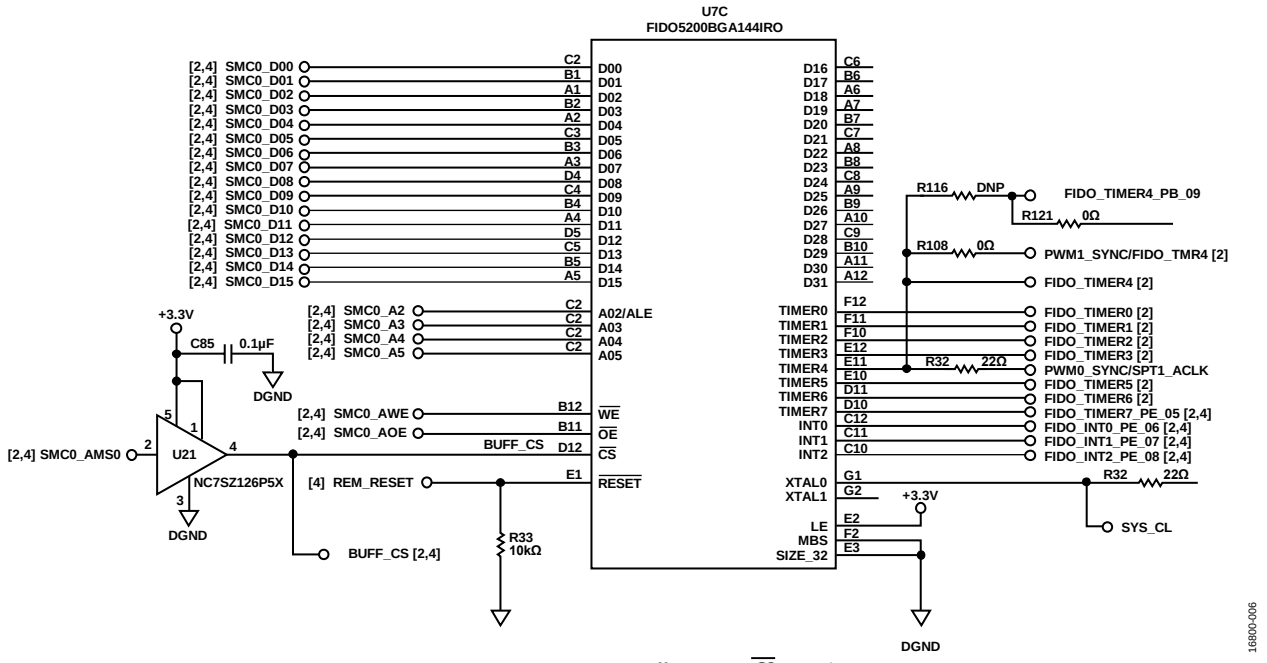

Figure 3. Noninverting Buffer Driving $\overline{\text{CS}}$ Signal

In case of nonmultiplexed operations, the address setup time (t_{AS}) is a critical parameter for asynchronous memory accesses. Depending on the microprocessor architecture, the address lines and $\overline{CS}$ can be driven at the same time ($t_{AS} = 0$) or with a small delay between them.

With regards to nonmultiplexed operations, many Cortex®-M4 families on the market (such as Analog Devices ADSP-CM408F or STM32F4) and Cortex-Ax families (such as the Analog Devices ADSP-SC589 and NXP imx6) drive $\overline{CS}$ and address lines with no delay. In other cases, like the Motorola 68000 architecture, $\overline{CS}$ is asserted after address lines are valid.

The fido5100 and fido5200 REM switches use the falling edge of $\overline{CS}$ to latch address lines, and a small delay t_{AS} is needed (minimum 20 ps).

When microprocessors assert $\overline{CS}$ and address at the same time, add a delay of several ns to $\overline{CS}$. For example, the ADSP-CM408F drives $\overline{CS}$ to the fido5100 and fido5200 REM switches with a fast one channel noninverting buffer (typical 2.5 ns added delay).

In case of multiplexed memory access, the timing relationship between the address and ALE signal valid may also require adding a short delay on the ALE signal, depending on the processor used.

INTERRUPTS

Three interrupt lines act as outputs for the fido5100 and fido5200 REM switches. The host microprocessor must monitor these lines.

MDIO INTERFACE

All Ethernet physical layers require configuration and can provide status information. Many devices use a management data input and output (MDIO) interface. This communication interface consists of two lines: a data line (MDIO) and a management data clock line (MDC).

A specific communication protocol is defined by the IEEE802.3 specification.

The fido5100 and fido5200 REM switches do not drive this communication interface. The host processor must be able to manage MDIO and MDC accordingly.

MEMORY REQUIREMENTS

Ideally, the REM switch driver needs 50 kB to 100 kB of read only memory (ROM). On the RapID platform, the fido1100 uses 46 kB of ROM.

In addition, 8 kB of RAM handles multiple packets in-flight in the processor simultaneously, based on the need for some industrial Ethernet protocols.

PIN COUNT

Table 1 summarizes the pin count for the configuration with the maximum pin count (125 MB/sec, nonmultiplexed bus) and the configuration with the minimum pin count (62.5 MB/sec, multiplexed bus), assuming MBS, SIZE_32, and LE are at a fixed level. In the nonmultiplexed bus, there are 45 pins, and in the multiplexed bus, there are 26 pins.

Table 1. Configuration Pin Count

Signals	Nonmultiplexed Bus (32-Bit)	Multiplexed Bus (16-Bit)
Data/Address	32	16
$\overline{RESET}$	1	1
A02 to A05	4	Not applicable
$\overline{WE}$	1	1
$\overline{CS}$	1	1
$\overline{OE}$	1	1
ALE (A02)	Not applicable	1
INTx	3	3
MDIO/MDC (for External Physical Layers)	2	2

INTERCONNECTION BLOCK DIAGRAM

The complete interconnection block diagram between the host or network processor and the fido5100 and fido5200 REM switches, along with the two Ethernet physical layers, is shown in Figure 4. The dotted line in Figure 4 indicates the connection for multiplexed mode only. In multiplexed mode, do not connect the four address lines, only connect A02/ALE. A02/ALE has double functionality. In multiplexed mode, the A02 line acts as the ALE signal to validate address lines.

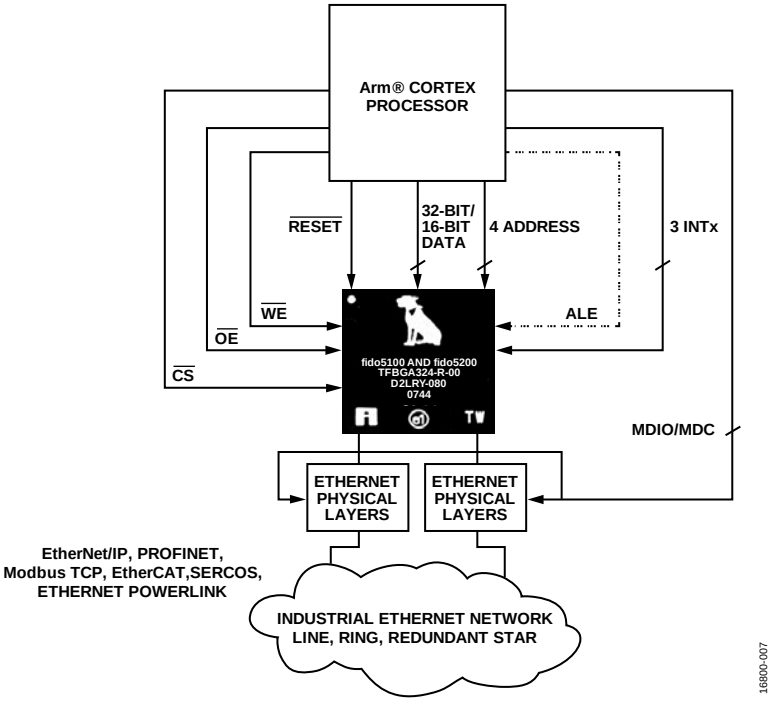

Figure 4. Interconnection Block Diagram

REM SWITCH SOFTWARE DRIVER

The REM switch software driver provides a standard, protocol-independent interface to the fido5100 and fido5200. The software driver is used for initialization, interrupt management, timer management, and protocol-independent packet transmission and receiving.

The REM switch software driver for each supported protocol (Profinet, Ethernet/IP, EtherCAT, ModbusTCP, and POWERLINK) is available as source code.

The driver is written in C language, and the driver configures the switch for the selected protocol.

This firmware is downloaded from the host processor.

The configuration is typically performed at power-up, but the configuration can be performed at any time after a system reset.

As shown in Figure 5, the C code is organized into a set of application programming interfaces (APIs), grouped into two major functional areas: a protocol specific interface and a standard switch interface.

Any transmission control protocol/internet protocol (TCP/IP) stack can be connected to the standard switch interface, and any protocol stack can be connected to the protocol specific interface. Because the standard switch interface is common across all drivers, the user only needs to connect to the TCP/IP stack once.

The user may choose the TCP/IP stack that comes as part of the operating system (OS) of the processor development environment and the PROFINET stack from a third-party vendor.

The protocol stack is then managed by the host processor and connects to protocol specific API in the REM switch driver, while the standard switch API in the REM driver connects to the TCP/IP stack in the OS.

Because the driver has no dependencies on any operating system resources (such as no threading and semaphores), porting is limited to defining how the host processor communicates with the REM switch and some debugging options.

The porting related code (**REMS_Port.h** and **REMS_Port.c**) is in the Porting directory (/**Porting/inc/** and /**Porting/src/**).

Those two files must be modified to support a specific hardware platform and are dependent on the host processor.

The REM software driver architecture is shown in Figure 6.

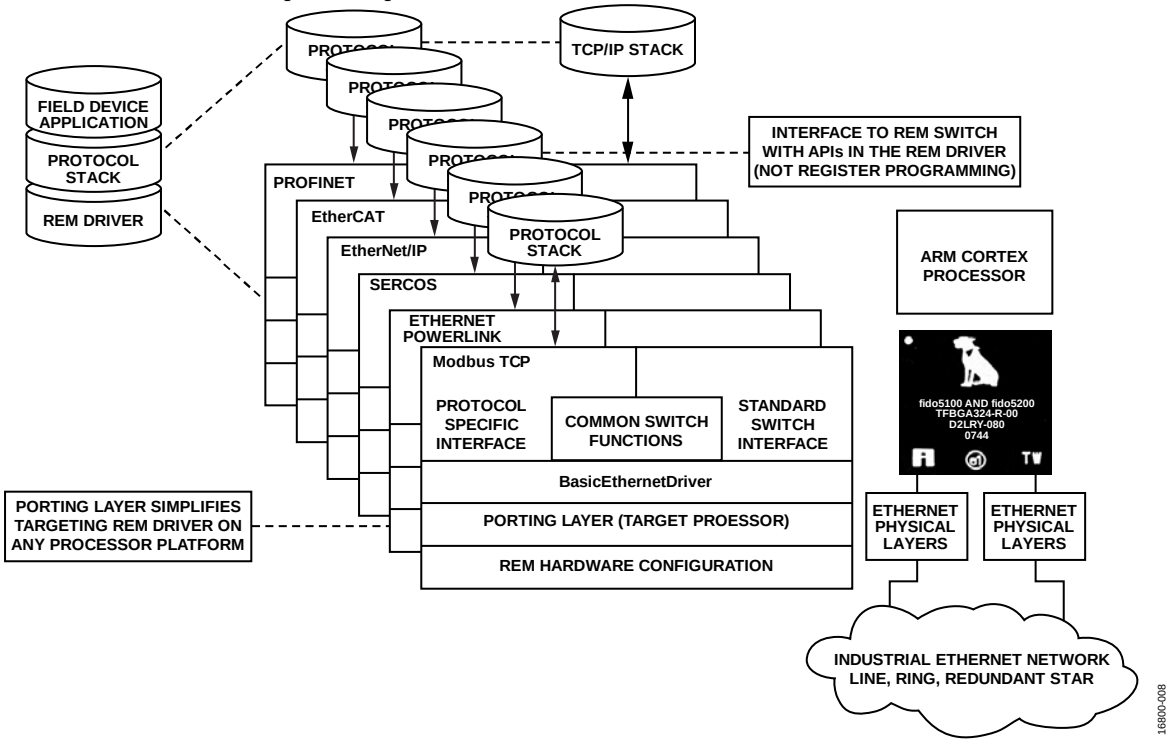

Figure 5. Multiprotocol Capability

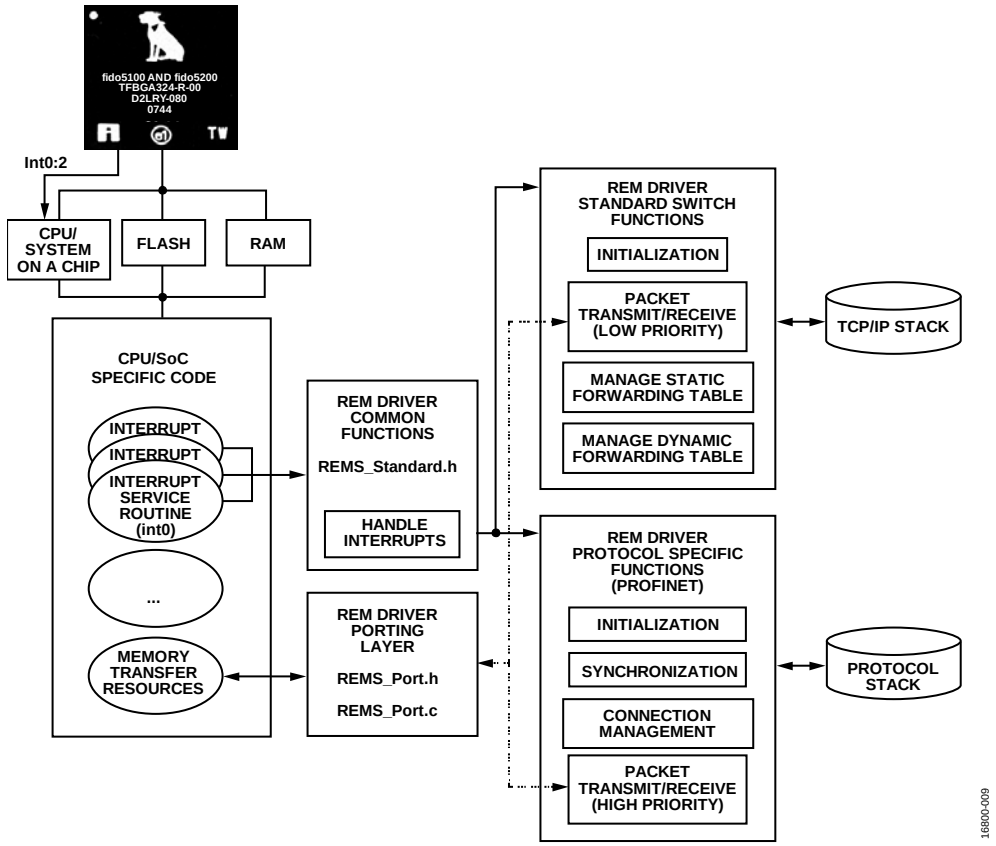

Figure 6. REM Software Driver Architecture

APPLICATION EXAMPLES

This section contains examples of pairing microprocessor architectures commonly used in industrial applications. Refer to specific documentation provided by the respective integrated circuits (IC) manufacturers for further details.

ANALOG DEVICES ADSP-CM408F (Arm® CORTEX-M4 WITH FPU)

The ADSP-CM408F comes with a flexible external memory interface. The static memory controller (SMC) can be programmed to control up to four banks of external memory. Asynchronous nonmultiplexed 16-bit operations are allowed.

Table 2 shows the connections between the fido5100 and fido5200 REM switches and the ADSP-CM408F processor.

Table 2. Connections to ADSP-CM408F

REM Switches Signals	ADSP-CM408F Signals
Data	SMC0_Dxx (from D00 to D15)
$\overline{\text{RESET}}$	GPIO pin
A02 to A05	SMC0_Axx
$\overline{\text{WE}}$	SMC0_AWE
$\overline{\text{CS}}$	SMC0_AMS0 (buffer needed)
$\overline{\text{OE}}$	SMC0_AOE
INTx	GPIO pins
MDIO/MDC (for External Physical Layers)	ETH0_MDIO/ETH0_MDC

ANALOG DEVICES ADSP-SC589 (SHARC®+ DUAL CORE DSP WITH ARM CORTEX-A5)

The ADSP-SC589 is equipped with SMC to control up to two blocks of external memory. Asynchronous nonmultiplexed 16-bit operations are allowed. As suggested in the ADSP-CM408F, the $\overline{\text{CS}}$ signal needs an external buffer, and the MDIO interface is managed by an embedded controller.

Table 3 shows the connections between the fido5100 and fido5200 REM switches and the dual SHARC + Arm processor.

Because the ADSP-SC589 does not have an embedded flash memory, the processor needs to program the REM switches at power-up to download code from an external ROM.

Table 3. Connections to ADSP-SC589

REM Switches Signals	ADSP-SC589 Signals
Data	SMC0_Dxx (from D00 to D15)
$\overline{\text{RESET}}$	GPIO pin
A02 to A05	SMC0_Axx
$\overline{\text{WE}}$	SMC0_AWE
$\overline{\text{CS}}$	SMC0_AMS0 (buffer needed)
$\overline{\text{OE}}$	SMC0_AOE
INTx	GPIO pins
MDIO/MDC (for External Physical Layers)	ETH0_MDIO/ETH0_MDC

ST MICROELECTRONICS STM32F42 FAMILY (ARM CORTEX-M4 WITH FPU)

The STM32F42 family comes with a flexible memory control (FMC) interface allowing asynchronous, 16-bit and 32-bit multiplexed and nonmultiplexed access operations. Because FMC_NE (chip select) can anticipate address valid as much as 2 ns maximum, a longer delay might be needed on this signal.

Table 4 shows the connections between the fido5100 and fido5200 REM switches and the STM32F42 processor.

Table 4. Connections to STM32F42

REM Switches Signals	Nonmultiplexed Bus (32-Bit) STM32F42 Signals	Multiplexed Bus (16-Bit) STM32F42 Signals
Data/Address	FMC_D[31:0]	FMC_D[15:0]
$\overline{\text{RESET}}$	GPIO pin	GPIO pin
A02 to A05	FMC_A[3:0]	Not applicable
$\overline{\text{WE}}$	FMC_NWE	FMC_NWE
$\overline{\text{CS}}$	FMC_NE1 (buffer needed)	FMC_NE1
$\overline{\text{OE}}$	FMC_NOE	FMC_NOE
ALE (A02)	Not applicable	FMC_NADV (inverter needed)
INTx	GPIO pins	GPIO pins
MDIO/MDC (for External Physical Layers)	ETH_MDIO/ ETH_MDC	ETH_MDIO/ ETH_MDC

In case of multiplexed operations, an inverter is needed because FMC_NADV is an active low signal but the fido5100 and fido5200 require an active high signal to latch address. In this case, there is no need to add a delay because the $t_{w\,(NADV)}$ time is sufficient for the fido5100 and fido5200 to latch address lines.

ST MICROELECTRONICS STM32F103 FAMILY (Arm CORTEX-M3)

The STM32F103 family has flexible static memory controller (FSMC) embedded, which can manage up to four memory banks of various types, SRAM included. FSMC allows asynchronous, multiplexed and nonmultiplexed 16-bit operations. Because FSMC_NE (chip select) can be asserted up to 3 ns before address lines are valid, a longer delay may be needed on this signal.

Table 5 shows the connections between the fido5100 and fido5200 REM switches and the STM32F103 processor.

Table 5. Connections to STM32F103

REM Switches Signals	Nonmultiplexed Bus (16-Bit) STM32F103 Signals	Multiplexed Bus (16-Bit) STM32F103 Signals
Data/Address	FSMC_D[15:0]	FSMC_D[15:0]
$\overline{\text{RESET}}$	GPIO pin	GPIO pin
A02 to A05	FSMC_A[3:0]	Not applicable
$\overline{\text{WE}}$	FSMC_NWE	FSMC_NWE
$\overline{\text{CS}}$	FSMC_NE1 (buffer needed)	FSMC_NE1
$\overline{\text{OE}}$	FSMC_NOE	FSMC_NOE
ALE (A02)	Not applicable	FSMC_NADV (inverter needed)
INTx	GPIO pins	GPIO pins
MDIO/MDC (for External Physical Layers)	GPIO pins	GPIO pins

In case of multiplexed operations, an inverter is needed because FSMC_NADV is an active low signal; however, the fido5100 and fido5200 require an active high signal to latch address. In this case, there is no need for adding a delay because the $t_{w\,(NADV)}$ time is sufficient for fido5100 and fido5200 to latch address lines.

This family of Cortex-M3 processors does not implement a MDIO interface.

Nevertheless, two GPIOs can be used instead, while the MDIO functionality must be emulate via software.

TEXAS INSTRUMENTS AM1808 SITARA FAMILY (Arm9)

The AM1808 is provided with an external memory interface (EMIFA) supporting asynchronous SRAM. EMIFA can manage 16-bit nonmultiplexed access. The chip select signal needs a buffer.

Table 6 shows the connections between the fido5100 and fido5200 REM switches and the AM1808 processor.

Table 6. Connections to AM1808

REM Switches Signals	AM1808 Signals
Data	EMA_D[15:0]
$\overline{\text{RESET}}$	GPIO pin
A02 to A05	EMA_A[3:0]
$\overline{\text{WE}}$	$\overline{\text{EMA\_WE}}$
$\overline{\text{CS}}$	$\overline{\text{EMA\_CS2}}$ (buffer needed)
$\overline{\text{OE}}$	$\overline{\text{EMA\_OE}}$
INTx	GPIO pins
MDIO/MDC (for External Physical Layers)	MDIO_D/MDIO_CLK

TEXAS INSTRUMENTS TMS320F2807 PICCOLO FAMILY (32-BIT FLOATING POINT MICROCONTROLLER)

The TMS320F2807 family comes with an EMIFA supporting asynchronous SRAM. The EMIFA can manage both 32-bit and 16-bit nonmultiplexed access. The chip select signal needs a buffer.

Table 7 shows the connections between the fido5100 and fido5200 REM switches and the TMS320F2807 processor.

Table 7. Connections to TMS320F2807

REM Switches Signals	TMS320F2807 Signals
Data	EM1D[x:0] (x = 31 or 15)
$\overline{\text{RESET}}$	GPIO pin
A02 to A05	EM1A[3:0]
$\overline{\text{WE}}$	$\overline{\text{EM1WE}}$
$\overline{\text{CS}}$	$\overline{\text{EM1CS2}}$ (buffer needed)
$\overline{\text{OE}}$	$\overline{\text{EM1OE}}$
INTx	GPIO pins
MDIO/MDC (for External Physical Layers)	GPIO pins

This family of 32-bit processors does not implement a MDIO interface. Regardless, two GPIOs can be used for this purpose, while the MDIO functionality must be emulated via software.

NXP i.MX 6ULL FAMILY (Arm CORTEX-A7)

The i.MX 6ULL processors are equipped with an external interface module (EIM), providing 16-bit nonmultiplexed and multiplexed access to SRAMs.

Table 8 shows the connections between the fido5100 and fido5200 REM switches and the i.MX 6ULL processor.

Table 8. Connections to i.MX 6ULL

REM Switches Signals	Nonmultiplexed Bus (16-bit) i.MX6 Signals	Multiplexed Bus (16-bit) i.MX6 Signals
Data/Address	EIM_DATAx [x = 15:0]	EIM_DATAx [x = 15:0]
$\overline{\text{RESET}}$	GPIO pin	GPIO pin
A02 to A05	EIM_ADx[3:0]	Not applicable
$\overline{\text{WE}}$	$\overline{\text{EIM\_WE}}$	$\overline{\text{EIM\_WE}}$
$\overline{\text{CS}}$	$\overline{\text{EIM\_CS0}}$ (buffer needed)	$\overline{\text{EIM\_CS0}}$
$\overline{\text{OE}}$	$\overline{\text{EIM\_OE}}$	$\overline{\text{EIM\_OE}}$
ALE (A02)	Not applicable	$\overline{\text{EIM\_LBA}}$ (inverter needed)
INTx	GPIO pins	GPIO pins
MDIO/MDC (for External Physical Layers)	ENET_MDIO/ ENET_MDC	ENET_MDIO/ ENET_MDC

In case of multiplexed operations, an inverter is needed because $\overline{\text{EIM\_LBA}}$ is an active low signal; however, the fido5100 and fido5200 require an active high signal to latch address.

Because $\overline{\text{EIM\_LBA}}$ deassertion (which becomes the latching time for address) depends on programmable parameters, take care when calculating W40A timing. A buffer may be necessary.

Timing W31 ($\overline{\text{EIM\_CSx}}$ valid to address valid) is programmable and can be negative ($\overline{\text{CS}}$ asserted before address is valid). A buffer may be necessary.