36 Debugging support

36.1 Overview

The FM33LC0XX chip is based on the ARM Cortex-M0 processor and supports corresponding debug features.

(breakpoint) and data watchpoint (watchpoint), the debugger can stop the CPU at specific instruction fetches and data accesses

The kernel runs, inspects the kernel registers and system peripheral states, and resumes kernel operation as needed.

The simulation debugging host is connected to the FM33LC0xx chip through the SWD interface to realize simulation debugging.

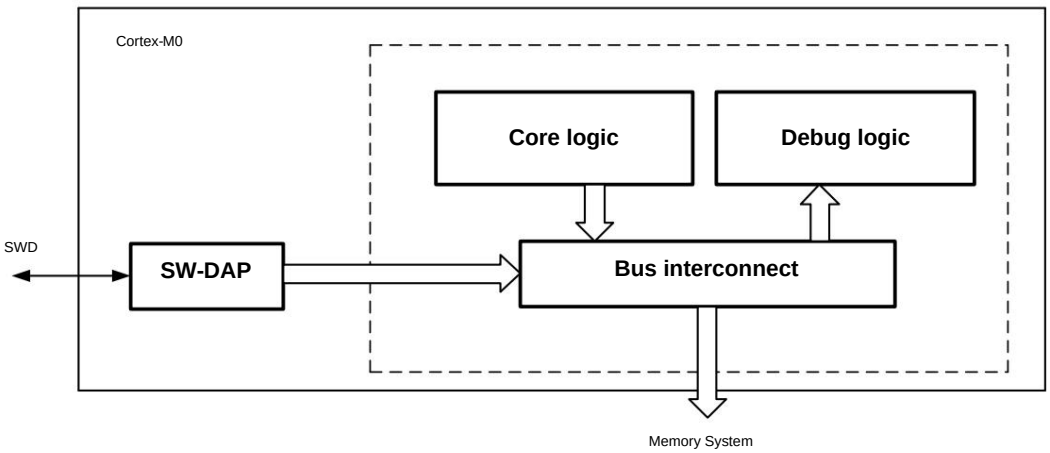


Figure **36-1 Schematic diagram of the Cortex-M0** debugging system

For the debug features of the Cortex-M0 core, please refer to ARM's Cortex-M0 Technical Reference Manual.

36.2 Debug Pin

36.2.1 SWD Pin

The SWD pin locations of the FM33LC0xx series MCUs are shown in the following table:

SWD pins	**Debug Function**	Pin Definition
SWEDING	SWD data input/output	PD8
SWCLK	SWD clock input	PD7

Note: After the chip is reset *PD7* and*PD8* The default is input state, which is the same as most*GPIO* different.

36.2.2 Pull-up Resistors

After the chip is reset, the SWDIO pin has an internal pull-up (about 100K ohms) enabled by default, and the SWCLK pin has an internal pull-up disabled by default.

Therefore, users need to connect an external pull-up resistor on the PCB to prevent the input pin from floating and causing increased leakage.

36.3 SWD Interface Protocol

36.3.1 Protocol Introduction

The SWD protocol uses LSB-first for data transmission and reception. Through the SWD interface, the debug host can read and write the DPACC and APACC registers.

Group.

Each time SWIO switches the data direction, a turn-around time is inserted on the bus. During this time, neither the master nor the slave will drive

SWIO. Between two transfers, the host must drive the line low to enter the idle state, or continue to send the start of a new transfer.

The SWD bit continues to be transmitted. After a data packet is transmitted, the host can also be idle, keeping the line high or pulled up by a pull-up resistor.

The protocol does not have an explicit reset signal, and the host or target will detect a reset if the expected signal is not seen.

The line is high for 50 clock cycles followed by a read ID request to ensure resynchronization after an error is detected or reset.

Step successful.

36.3.2 Transmission Sequence

Each SWD communication transmission sequence consists of three parts:

1. Packet request (8 bits), sent by the host

2. ACK response (3 bits), sent back by the slave

3. Data transmission (33 bits), sent by the host or slave

The packet request bytes are defined as follows:

Bit	**Name**	describe
0	Start	Start bit, must be 1
1	ApnDP	AP/DP Selection 0: DP access 1: AP access
2	RnW	Read and write selection 0: Write request 1: Read request
4:3	A[3:2]	DP/AP register address field
5	Parity	Bit0~Bit4 data check bit
6	Stop	0
7	Park	The host does not drive, through the bus pull-up, the slave reads 1

After the packet request is sent, there is always a 1-bit turn-around time on the bus.

The ACK response is defined as follows:

Bit	**Name**	describe
0:2	ACK	001ÿFAULT

		010ÿWAIT 100ÿOK

If the host initiates a read operation, or the ACK is WAIT or FAULT, a turn-around time must be inserted after the ACK.

The data transmission format is as follows:

Bit	**Name**	describe
0:31	Data	Data read or written
32	Parity	32-bit data check bit

36.3.3 SW-DP ID code

The SW-DP of Cortex-M0 has a fixed ID code: 0x0BB11477

The SW-DP is inactive until the host reads the ID code.

ÿ After the chip is reset, or SWIO is pulled high for 50 SWCLK cycles, SW-DP is in RESET state

ÿ After pulling SWIO low for at least 2 SWCLK cycles, SW-DP enters IDLE state

ÿ When SW-DP is in RESET, the host must first put it into IDLE and then read the ID code register to activate it.

Otherwise, the slave will respond to the host's communication with a FAULT response.

36.3.4 Host Read Operation

A successful read operation consists of the following three stages:

ÿ An 8-bit read packet request from the host to the target.

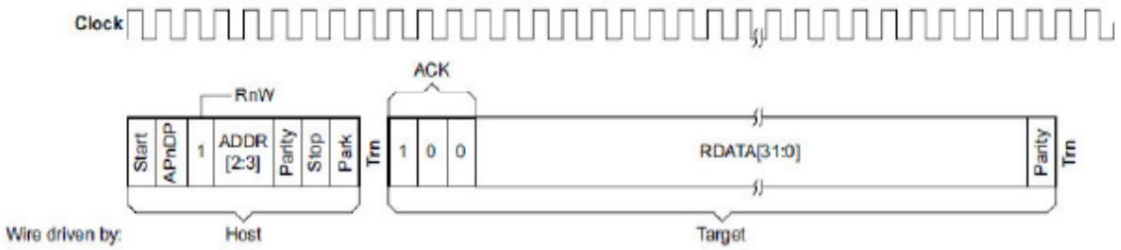
ÿ A 3-bit acknowledgment (ack) from the target to the host. A successful OK response is 100, a WAIT response is 010, and a FAULT response is 011.

is 001.

ÿ A 33-bit data read phase (payload) from the host to the target.

By default, there is a clock turnaround cycle between the first and second phases and after the third phase. A successful read operation

As shown below.



36.3.5 Host Write Operation

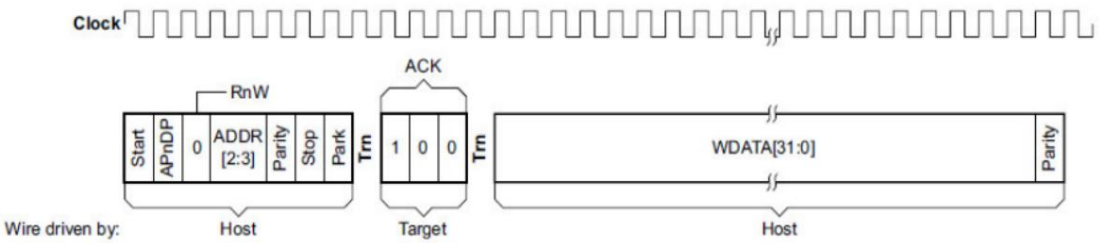
A write operation consists of the following three stages:

ÿ An 8-bit write packet request (header) from the host to the target.

ÿ A 3-bit acknowledgement (ack) from the target to the host. A successful OK response is 100, a FAULT response is 001.

ÿ A 33-bit data write phase (payload) from the host to the target.

By default, there is a clock turnaround cycle between each two stages. A successful write operation is shown in the following figure.



36.4 SWD-DP Register

36.4.1 Register List

Address (A[3:2])	DPBANKSEL	Name	Access
00	x	DHCSR	RO
		ABORT	WHERE
01	0x0	CTRL/STAT	RW
	0x1	DLCR	RW
10	x	RESEND	RO
		SELECT	WHERE
11	x	RDBUFF	RO

For a detailed description of the registers, please refer to the Cortex-M0 Technical Reference Manual.

36.5 Core debug registers

Core debugging can be achieved by operating the core debug registers. The host accesses the following core debug registers through SW-DP.

Address	Name	Type	Function
0xE000EDF0	DHCSR	RW	Debug Halting Control and Status Register
0xE000EDF4	DCRSR	WO	Debug Core Register Selector Register
0xE000EDF8	DCRDR	RW	Debug Core Register Data Register
0xE000EDFC	DEMCR	RW	Debug Exception and Monitor Control Register
0xE000EE00 to 0xE000EEFF			Reserved for Debug Extension

The above debug registers are not affected by system reset, but only by power-on reset.

haltÿ

ÿ Set bit0 of the DEMCR register (VC_CORRESET)

ÿ Set bit0 (C_DEBUGEN) of the DHCSR register

ÿ Perform a system reset

36.6 Debug related configuration items

By configuring the DBG_CR register, you can set whether the timer and watchdog circuit inside the chip continue to work in the debug state.

For details, see 6.5.1DEBUG Configuration Register (DBG_CR).