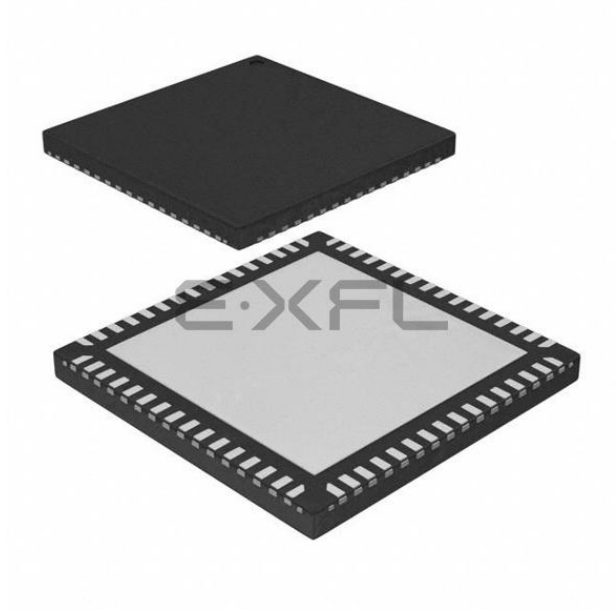


Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Active
Core Processor	ARM® Cortex®-M7
Core Size	32-Bit Single-Core
Speed	300MHz
Connectivity	I²C, IrDA, LINbus, SPI, SSC, UART/USART, USB
Peripherals	Brown-out Detect/Reset, DMA, I²S, POR, PWM, WDT
Number of I/O	44
Program Memory Size	2MB (2M x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	384K x 8
Voltage - Supply (Vcc/Vdd)	1.7V ~ 3.6V
Data Converters	A/D 5x12b; D/A 1x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 105°C (TA)
Mounting Type	Surface Mount
Package / Case	64-VFQFN Exposed Pad
Supplier Device Package	64-QFN (9x9)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/atsams70j21b-mnt

22.4 Functional Description

22.4.1 Embedded Flash Organization

The embedded Flash interfaces directly with the internal bus. The embedded Flash is composed of:

- One memory plane organized in several pages of the same size for the code
- A separate 2 x 512-byte memory area which includes the unique chip identifier
- A separate 512-byte memory area for the user signature
- Two 128-bit read buffers used for code read optimization
- One 128-bit read buffer used for data read optimization
- One write buffer that manages page programming. The write buffer size is equal to the page size. This buffer is write-only and accessible all along the 1 Mbyte address space, so that each word can be written to its final address.
- Several lock bits used to protect write/erase operation on several pages (lock region). A lock bit is associated with a lock region composed of several pages in the memory plane.
- Several bits that may be set and cleared through the EEFC interface, called general-purpose non-volatile memory bits (GPNVM bits)

The embedded Flash size, the page size, the organization of lock regions and the definition of GPNVM bits are specific to the device. The EEFC returns a descriptor of the Flash controller after a ‘Get Flash Descriptor’ command has been issued by the application (see the “Get Flash Descriptor Command” section).

31.20.19 PMC Fast Startup Polarity Register

Name: PMC_FSPR
Offset: 0x0074
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the PMC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	FSTP15	FSTP14	FSTP13	FSTP12	FSTP11	FSTP10	FSTP9	FSTP8
Access								
Reset	0	0	0	0	0	0	0	0

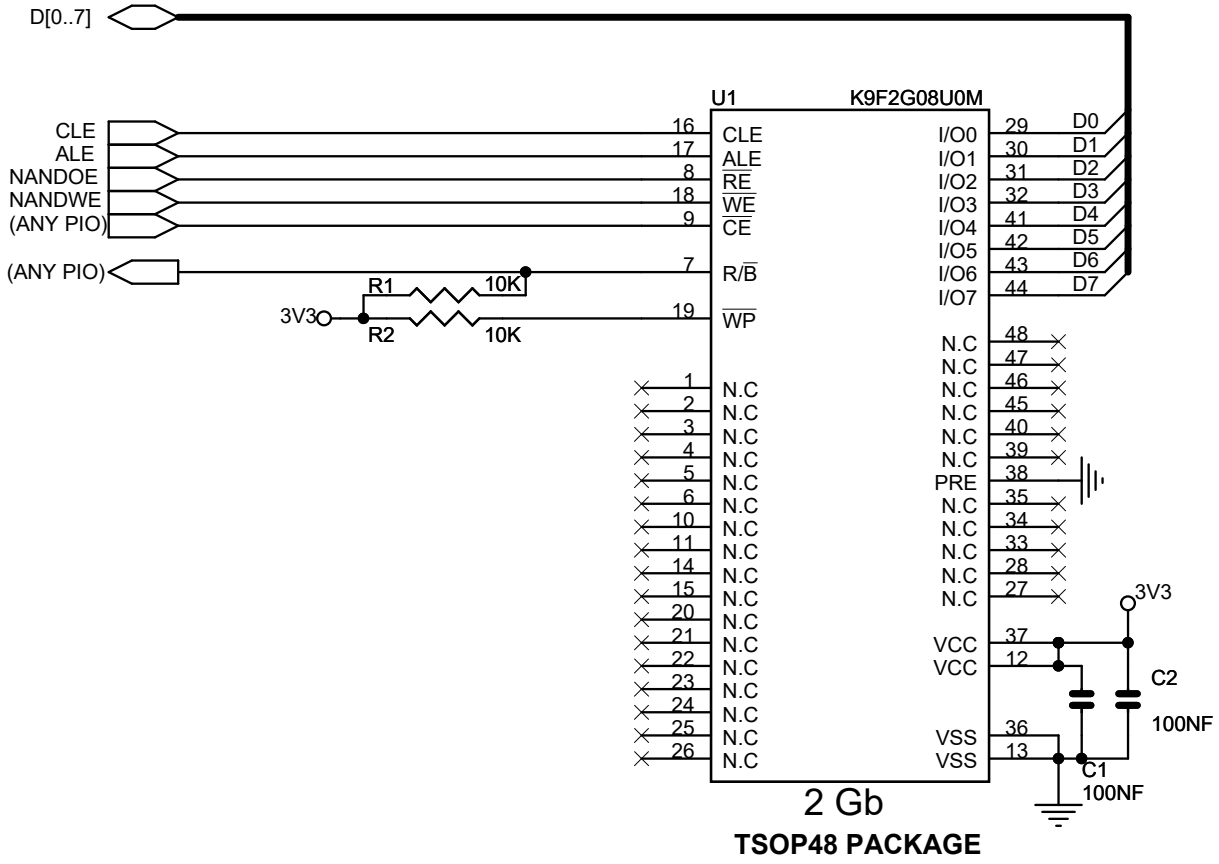
Bit	7	6	5	4	3	2	1	0
	FSTP7	FSTP6	FSTP5	FSTP4	FSTP3	FSTP2	FSTP1	FSTP0
Access								
Reset	0	0	0	0	0	0	0	0

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15 – FSTP Fast Startup Input Polarity x bits
Defines the active polarity of the corresponding wake-up input. If the corresponding wake-up input is enabled and at the FSTP level, it enables a fast restart signal.

35.8.1.1 8-bit NAND Flash

Hardware Configuration

Figure 35-7. 8-bit NAND Flash



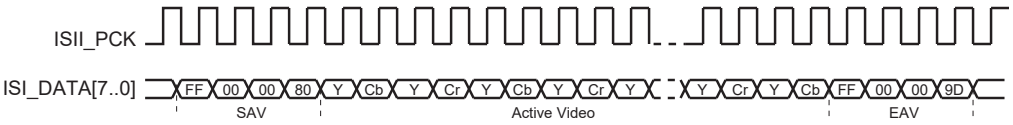
Software Configuration
Perform the following configuration:

1. Select the chip select used to drive the NAND Flash by setting the bit CCFG_SMCNFCS.SMC_NFCSx.
2. Reserve A21 / A22 for ALE / CLE functions. Address and Command Latches are controlled by setting the address bits A21 and A22, respectively, during accesses.
3. NANDOE and NANDWE signals are multiplexed with PIO lines. Thus, the dedicated PIOs must be programmed in Peripheral mode in the PIO controller.
4. Configure a PIO line as an input to manage the Ready/Busy signal.
5. Configure SMC CS3 Setup, Pulse, Cycle and Mode according to NAND Flash timings, the data bus width and the system bus frequency.

In this example, the NAND Flash is not addressed as a “CE don’t care”. To address it as a “CE don’t care”, connect NCS3 (if SMC_NFCS3 is set) to the NAND Flash CE.

The data timing using EAV/SAV sequence synchronization are shown in the following figure.

Figure 37-4. SAV and EAV Sequence Synchronization



37.5.2 Data Ordering

The RGB color space format is required for viewing images on a display screen preview, and the YCbCr color space format is required for encoding.

All the sensors do not output the YCbCr or RGB components in the same order. The ISI allows the user to program the same component order as the sensor, reducing software treatments to restore the right format.

Table 37-2. Data Ordering in YCbCr Mode

Mode	Byte 0	Byte 1	Byte 2	Byte 3
Default	Cb(i)	Y(i)	Cr(i)	Y(i+1)
Mode 1	Cr(i)	Y(i)	Cb(i)	Y(i+1)
Mode 2	Y(i)	Cb(i)	Y(i+1)	Cr(i)
Mode 3	Y(i)	Cr(i)	Y(i+1)	Cb(i)

Table 37-3. RGB Format in Default Mode, RGB_CFG = 00, No Swap

Mode	Byte	D7	D6	D5	D4	D3	D2	D1	D0
RGB 8:8:8	Byte 0	R7(i)	R6(i)	R5(i)	R4(i)	R3(i)	R2(i)	R1(i)	R0(i)
	Byte 1	G7(i)	G6(i)	G5(i)	G4(i)	G3(i)	G2(i)	G1(i)	G0(i)
	Byte 2	B7(i)	B6(i)	B5(i)	B4(i)	B3(i)	B2(i)	B1(i)	B0(i)
	Byte 3	R7(i+1)	R6(i+1)	R5(i+1)	R4(i+1)	R3(i+1)	R2(i+1)	R1(i+1)	R0(i+1)
RGB 5:6:5	Byte 0	R4(i)	R3(i)	R2(i)	R1(i)	R0(i)	G5(i)	G4(i)	G3(i)
	Byte 1	G2(i)	G1(i)	G0(i)	B4(i)	B3(i)	B2(i)	B1(i)	B0(i)
	Byte 2	R4(i+1)	R3(i+1)	R2(i+1)	R1(i+1)	R0(i+1)	G5(i+1)	G4(i+1)	G3(i+1)
	Byte 3	G2(i+1)	G1(i+1)	G0(i+1)	B4(i+1)	B3(i+1)	B2(i+1)	B1(i+1)	B0(i+1)

Table 37-4. RGB Format, RGB_CFG = 10 (Mode 2), No Swap

Mode	Byte	D7	D6	D5	D4	D3	D2	D1	D0
RGB 5:6:5	Byte 0	G2(i)	G1(i)	G0(i)	R4(i)	R3(i)	R2(i)	R1(i)	R0(i)
	Byte 1	B4(i)	B3(i)	B2(i)	B1(i)	B0(i)	G5(i)	G4(i)	G3(i)
	Byte 2	G2(i+1)	G1(i+1)	G0(i+1)	R4(i+1)	R3(i+1)	R2(i+1)	R1(i+1)	R0(i+1)
	Byte 3	B4(i+1)	B3(i+1)	B2(i+1)	B1(i+1)	B0(i+1)	G5(i+1)	G4(i+1)	G3(i+1)

$$\begin{bmatrix} Y \\ C_r \\ C_b \end{bmatrix} = \begin{bmatrix} C_0 & C_1 & C_2 \\ C_3 & -C_4 & -C_5 \\ -C_6 & -C_7 & C_8 \end{bmatrix} \times \begin{bmatrix} R \\ G \\ B \end{bmatrix} + \begin{bmatrix} Y_{\text{off}} \\ \text{Cr}_{\text{off}} \\ \text{Cb}_{\text{off}} \end{bmatrix}$$

An example of coefficients is given below:

$$\begin{cases} Y = 0.257 \cdot R + 0.504 \cdot G + 0.098 \cdot B + 16 \\ C_r = 0.439 \cdot R - 0.368 \cdot G - 0.071 \cdot B + 128 \\ C_b = -0.148 \cdot R - 0.291 \cdot G + 0.439 \cdot B + 128 \end{cases}$$

37.5.5.2 Memory Interface

Dedicated FIFOs are used to support packed memory mapping. YCrCb pixel components are sent in a single 32-bit word in a contiguous space (packed). Data is stored in the order of natural scan lines. Planar mode is not supported.

37.5.5.3 DMA Features

Like preview datapath, codec datapath DMA mode uses linked list operation.

38.8 Register Summary

Offset	Name	Bit Pos.								
0x00	GMAC_NCR	7:0	WESTAT	INCSTAT	CLRSTAT	MPE	TXEN	RXEN	LBL	
		15:8	SRTSM			TXZQPF	TXPF	THALT	TSTART	BP
		23:16						FNP	TXPBPF	ENPBPR
		31:24								
0x04	GMAC_NCFGR	7:0	UNIHEN	MTIHEN	NBC	CAF	JFRAME	DNVLAN	FD	SPD
		15:8	RXBUFO[1:0]		PEN	RTY				MAXFS
		23:16	DCPF	DBW[1:0]		CLK[2:0]			RFCS	LFERD
		31:24		IRXER	RXBP	IPGSEN		IRXFCS	EFRHD	RXCOEN
0x08	GMAC_NSR	7:0						IDLE	MDIO	
		15:8								
		23:16								
		31:24								
0x0C	GMAC_UR	7:0								
		15:8								
		23:16								
		31:24								
0x10	GMAC_DCFGR	7:0	ESPA	ESMA		FBLDO[4:0]				
		15:8					TXCOEN	TXPBMS	RXBMS[1:0]	
		23:16	DRBS[7:0]							
		31:24								DDRP
0x14	GMAC_TSR	7:0			TXCOMP	TFC	TXGO	RLE	COL	UBR
		15:8								HRESP
		23:16								
		31:24								
0x18	GMAC_RBQB	7:0	ADDR[5:0]							
		15:8	ADDR[13:6]							
		23:16	ADDR[21:14]							
		31:24	ADDR[29:22]							
0x1C	GMAC_TBQB	7:0	ADDR[5:0]							
		15:8	ADDR[13:6]							
		23:16	ADDR[21:14]							
		31:24	ADDR[29:22]							
0x20	GMAC_RSR	7:0					HNO	RXOVR	REC	BNA
		15:8								
		23:16								
		31:24								
0x24	GMAC_ISR	7:0	TCOMP	TFC	RLEX	TUR	TXUBR	RXUBR	RCOMP	MFS
		15:8		PFTR	PTZ	PFNZ	HRESP	ROVR		
		23:16	PDRSFR	PDRQFR	SFT	DRQFT	SFR	DRQFR		
		31:24			TSUTIMCMP	WOL	RXLPISBC	SRI	PDRSFT	PDRQFT
0x28	GMAC_IER	7:0	TCOMP	TFC	RLEX	TUR	TXUBR	RXUBR	RCOMP	MFS
		15:8	EXINT	PFTR	PTZ	PFNZ	HRESP	ROVR		
		23:16	PDRSFR	PDRQFR	SFT	DRQFT	SFR	DRQFR		

This bit is cleared by writing a '1' to it.

Offset	Name	Bit Pos.								
0x01D4	USBHS_DEVEPTIMR5 (ISOENPT)	7:0	SHORTPACKETE	CRCERRE	OVERFE	HBISOFLUSHE	HBISOINERRE	UNDERFE	RXOUTE	TXINE
		15:8		FIFOCON	KILLBK	NBUSYBKE		ERRORTRANSE	DATAXE	MDATAE
		23:16						RSTDT		EPDISHDMA
		31:24								
0x01D8	USBHS_DEVEPTIMR6	7:0	SHORTPACKETE	STALLEDE	OVERFE	NAKINE	NAKOUTE	RXSTPE	RXOUTE	TXINE
		15:8		FIFOCON	KILLBK	NBUSYBKE				
		23:16					STALLRQ	RSTDT	NYETDIS	EPDISHDMA
		31:24								
0x01D8	USBHS_DEVEPTIMR6 (ISOENPT)	7:0	SHORTPACKETE	CRCERRE	OVERFE	HBISOFLUSHE	HBISOINERRE	UNDERFE	RXOUTE	TXINE
		15:8		FIFOCON	KILLBK	NBUSYBKE		ERRORTRANSE	DATAXE	MDATAE
		23:16						RSTDT		EPDISHDMA
		31:24								
0x01DC	USBHS_DEVEPTIMR7	7:0	SHORTPACKETE	STALLEDE	OVERFE	NAKINE	NAKOUTE	RXSTPE	RXOUTE	TXINE
		15:8		FIFOCON	KILLBK	NBUSYBKE				
		23:16					STALLRQ	RSTDT	NYETDIS	EPDISHDMA
		31:24								
0x01DC	USBHS_DEVEPTIMR7 (ISOENPT)	7:0	SHORTPACKETE	CRCERRE	OVERFE	HBISOFLUSHE	HBISOINERRE	UNDERFE	RXOUTE	TXINE
		15:8		FIFOCON	KILLBK	NBUSYBKE		ERRORTRANSE	DATAXE	MDATAE
		23:16						RSTDT		EPDISHDMA
		31:24								
0x01E0	USBHS_DEVEPTIMR8	7:0	SHORTPACKETE	STALLEDE	OVERFE	NAKINE	NAKOUTE	RXSTPE	RXOUTE	TXINE
		15:8		FIFOCON	KILLBK	NBUSYBKE				
		23:16					STALLRQ	RSTDT	NYETDIS	EPDISHDMA
		31:24								
0x01E0	USBHS_DEVEPTIMR8 (ISOENPT)	7:0	SHORTPACKETE	CRCERRE	OVERFE	HBISOFLUSHE	HBISOINERRE	UNDERFE	RXOUTE	TXINE
		15:8		FIFOCON	KILLBK	NBUSYBKE		ERRORTRANSE	DATAXE	MDATAE
		23:16						RSTDT		EPDISHDMA
		31:24								
0x01E4	USBHS_DEVEPTIMR9	7:0	SHORTPACKETE	STALLEDE	OVERFE	NAKINE	NAKOUTE	RXSTPE	RXOUTE	TXINE
		15:8		FIFOCON	KILLBK	NBUSYBKE				
		23:16					STALLRQ	RSTDT	NYETDIS	EPDISHDMA
		31:24								
0x01E4	USBHS_DEVEPTIMR9 (ISOENPT)	7:0	SHORTPACKETE	CRCERRE	OVERFE	HBISOFLUSHE	HBISOINERRE	UNDERFE	RXOUTE	TXINE

Offset	Name	Bit Pos.								
		23:16	BUFF_ADD[23:16]							
		31:24	BUFF_ADD[31:24]							
0x0728	USBHS_HSTDMACONTROLx	7:0	BURST_LCK	DESC_LD_IT	END_BUFFIT	END_TR_IT	END_B_EN	END_TR_EN	LDNXT_DSC	CHANN_ENB
		15:8								
		23:16	BUFF_LENGTH[7:0]							
		31:24	BUFF_LENGTH[15:8]							
0x072C	USBHS_HSTDMASTATUSx	7:0		DESC_LDST	END_BF_ST	END_TR_ST			CHANN_ACT	CHANN_ENB
		15:8								
		23:16	BUFF_COUNT[7:0]							
		31:24	BUFF_COUNT[15:8]							
0x0730	USBHS_HSTDMANXTDSC4	7:0	NXT_DSC_ADD[7:0]							
		15:8	NXT_DSC_ADD[15:8]							
		23:16	NXT_DSC_ADD[23:16]							
		31:24	NXT_DSC_ADD[31:24]							
0x0734	USBHS_HSTDMAADDRESSx	7:0	BUFF_ADD[7:0]							
		15:8	BUFF_ADD[15:8]							
		23:16	BUFF_ADD[23:16]							
		31:24	BUFF_ADD[31:24]							
0x0738	USBHS_HSTDMACONTROLx	7:0	BURST_LCK	DESC_LD_IT	END_BUFFIT	END_TR_IT	END_B_EN	END_TR_EN	LDNXT_DSC	CHANN_ENB
		15:8								
		23:16	BUFF_LENGTH[7:0]							
		31:24	BUFF_LENGTH[15:8]							
0x073C	USBHS_HSTDMASTATUSx	7:0		DESC_LDST	END_BF_ST	END_TR_ST			CHANN_ACT	CHANN_ENB
		15:8								
		23:16	BUFF_COUNT[7:0]							
		31:24	BUFF_COUNT[15:8]							
0x0740	USBHS_HSTDMANXTDSC5	7:0	NXT_DSC_ADD[7:0]							
		15:8	NXT_DSC_ADD[15:8]							
		23:16	NXT_DSC_ADD[23:16]							
		31:24	NXT_DSC_ADD[31:24]							
0x0744	USBHS_HSTDMAADDRESSx	7:0	BUFF_ADD[7:0]							
		15:8	BUFF_ADD[15:8]							
		23:16	BUFF_ADD[23:16]							
		31:24	BUFF_ADD[31:24]							
0x0748	USBHS_HSTDMACONTROLx	7:0	BURST_LCK	DESC_LD_IT	END_BUFFIT	END_TR_IT	END_B_EN	END_TR_EN	LDNXT_DSC	CHANN_ENB
		15:8								
		23:16	BUFF_LENGTH[7:0]							
		31:24	BUFF_LENGTH[15:8]							
0x074C	USBHS_HSTDMASTATUSx	7:0		DESC_LDST	END_BF_ST	END_TR_ST			CHANN_ACT	CHANN_ENB
		15:8								
		23:16	BUFF_COUNT[7:0]							
		31:24	BUFF_COUNT[15:8]							
0x0750	USBHS_HSTDMANXTDSC6	7:0	NXT_DSC_ADD[7:0]							
		15:8	NXT_DSC_ADD[15:8]							
		23:16	NXT_DSC_ADD[23:16]							
		31:24	NXT_DSC_ADD[31:24]							

Value	Description
0	Cleared when USBHS_DEVEPTIDRx.NYETDISC = 1. This enables the USBHS to handle the high-speed handshake following the USB 2.0 standard.
1	Set when USBHS_DEVEPTIERx.NYETDISS = 1. This sends a ACK handshake instead of a NYET handshake in High-speed mode.

Bit 16 – EPDISHDMA Endpoint Interrupts Disable HDMA Request
This bit is set when USBHS_DEVEPTIERx.EPDISHDMAS = 1. This pauses the on-going DMA channel x transfer on any Endpoint x interrupt (PEP_x), whatever the state of the Endpoint x Interrupt Enable bit (PEP_x).

The user then has to acknowledge or to disable the interrupt source (e.g. USBHS_DEVEPTISRx.RXOUTI) or to clear the EPDISHDMA bit (by writing a one to the USBHS_DEVEPTIDRx.EPDISHDMAC bit) in order to complete the DMA transfer.

In Ping-pong mode, if the interrupt is associated to a new system-bank packet (e.g. Bank1) and the current DMA transfer is running on the previous packet (Bank0), then the previous-packet DMA transfer completes normally, but the new-packet DMA transfer does not start (not requested).

If the interrupt is not associated to a new system-bank packet (USBHS_DEVEPTISRx.NAKINI, NAKOUTI, etc.), then the request cancellation may occur at any time and may immediately pause the current DMA transfer.

This may be used for example to identify erroneous packets, to prevent them from being transferred into a buffer, to complete a DMA transfer by software after reception of a short packet, etc.

Bit 14 – FIFOCON FIFO Control
For control endpoints:

The FIFOCON and RWALL bits are irrelevant. Therefore, the software never uses them on these endpoints. When read, their value is always 0.

For IN endpoints:

0: Cleared (by writing a one to the USBHS_DEVEPTIDRx.FIFOCONC bit) to send the FIFO data and to switch to the next bank.

1: Set when the current bank is free, at the same time as USBHS_DEVEPTISRx.TXINI.

For OUT endpoints:

0: Cleared (by writing a one to the USBHS_DEVEPTIDRx.FIFOCONC bit) to free the current bank and to switch to the next bank.

1: Set when the current bank is full, at the same time as USBHS_DEVEPTISRx.RXOUTI.

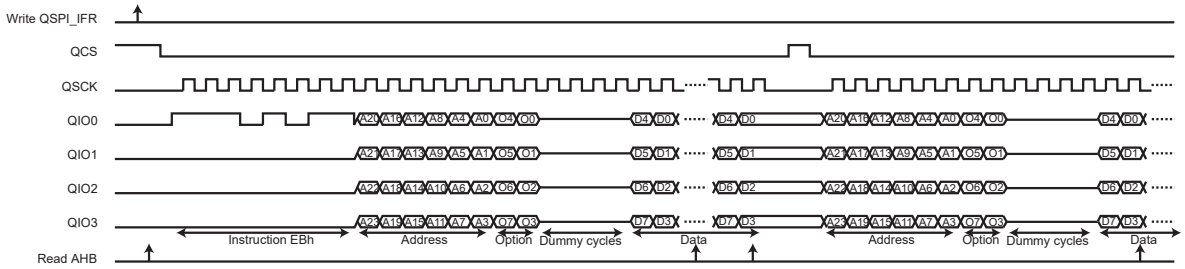
Bit 13 – KILLBK Kill IN Bank
This bit is set when the USBHS_DEVEPTIERx.KILLBKS bit is written to one. This kills the last written bank.

This bit is cleared when the bank is killed.

⚠ CAUTION The bank is really cleared when the “kill packet” procedure is accepted by the USBHS core. This bit is automatically cleared after the end of the procedure.

- Write a ‘1’ to QSPI_CR.LASTXFR.
- Wait for QSPI_SR.INSTRE to rise.

Figure 42-17. Instruction Transmission Waveform 7



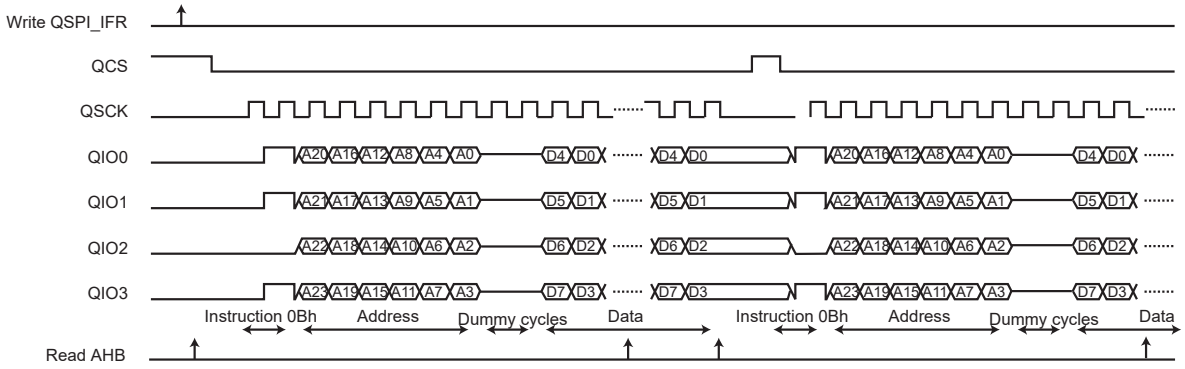
Example 8:

Instruction in Quad SPI, with address in Quad SPI, without option, with data read in Quad SPI, with two dummy cycles, with fetch.

Command: HIGH-SPEED READ (0Bh)

- Write 0x0000_000B in QSPI_ICR.
- Write 0x0002_20B6 in QSPI_IFR.
- Read QSPI_IFR (dummy read) to synchronize system bus accesses.
- Read data in the QSPI system bus memory space (0x80000000).
  Fetch is enabled, the address of the system bus read accesses is always used.
- Write a ‘1’ to QSPI_CR.LASTXFR.
- Wait for QSPI_SR.INSTRE to rise.

Figure 42-18. Instruction Transmission Waveform 8



Example 9:

Instruction in Quad SPI, without address, without option, with data read in Quad SPI, without dummy cycles, without fetch.

Command: HIGH-SPEED READ (05h)

- Write 0x0000_0005 in QSPI_ICR.
- Write 0x0000_0096 in QSPI_IFR.
- Read QSPI_IFR (dummy read) to synchronize system bus accesses.
- Read data in the QSPI system bus memory space (0x80000000).
  Fetch is disabled.
- Write a ‘1’ to QSPI_CR.LASTXFR.

42.7.9 QSPI Serial Clock Register

Name: QSPI_SCR
Offset: 0x20
Reset: 0x00000000
Property: Read/Write

This register can only be written if bit WPEN is cleared in the QSPI Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	DLYBS[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	SCBR[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
							CPHA	CPOL
Access							R/W	R/W
Reset							0	0

Bits 23:16 – DLYBS[7:0] Delay Before QSCK
This field defines the delay from QCS valid to the first valid QSCK transition.

When DLYBS equals zero, the QCS valid to QSCK transition is 1/2 the QSCK clock period.

Otherwise, the following equation determines the delay:

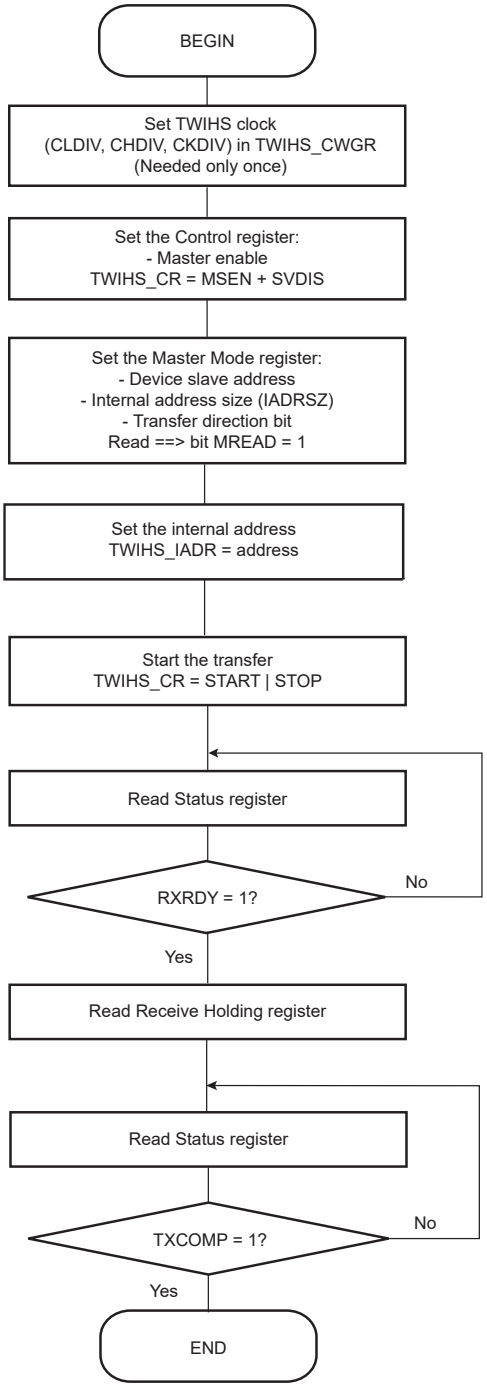
DLYBS = Delay Before QSCK × $f_{peripheral\ clock}$

Bits 15:8 – SCBR[7:0] Serial Clock Baud Rate
The QSPI uses a modulus counter to derive the QSCK baud rate from the peripheral clock. The baud rate is selected by writing a value from 0 to 255 in the SCBR field. The following equation determines the QSCK baud rate:

SCBR = ($f_{peripheral\ clock}$ / QSCK Baudrate) - 1

Bit 1 – CPHA Clock Phase
CPHA determines which edge of QSCK causes data to change and which edge causes data to be captured. CPHA is used with CPOL to produce the required clock/data relationship between master and slave devices.

Figure 43-22. TWIHS Read Operation with Single Data Byte and Internal Address



46.6.9.14.7 Header Timeout Error

This error is generated in slave node configuration, if the Header is not entirely received within the time given by the maximum length of the Header, $t_{Header\_Maximum}$.

This error is reported by flag US_CSR.LINHTE.

46.6.9.15 LIN Frame Handling

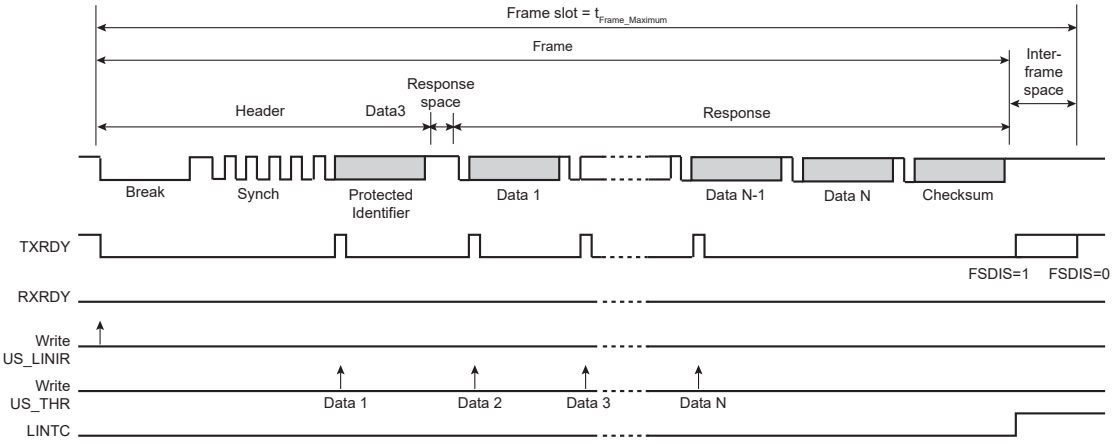
46.6.9.15.1 Master Node Configuration

- Write TXEN and RXEN in US_CR to enable both the transmitter and the receiver.
- Write USART_MODE in US_MR to select the LIN mode and the master node configuration.
- Write CD and FP in US_BRGR to configure the baud rate.
- Write NACT, PARDIS, CHKDIS, CHKTYPE, DLCM, FSDIS and DLC in US_LINMR to configure the frame transfer.
- Check that TXRDY in US_CSR is set to 1.
- Write IDCHR in US_LINIR to send the header.

What comes next depends on the NACT configuration:

- Case 1: NACT = PUBLISH, the USART sends the response
  - Wait until TXRDY in US_CSR rises.
  - Write TCHR in US_THR to send a byte.
  - If all the data have not been written, redo the two previous steps.
  - Wait until LINTC in US_CSR rises.
  - Check the LIN errors.
- Case 2: NACT = SUBSCRIBE, the USART receives the response
  - Wait until RXRDY in US_CSR rises.
  - Read RCHR in US_RHR.
  - If all the data have not been read, redo the two previous steps.
  - Wait until LINTC in US_CSR rises.
  - Check the LIN errors.
- Case 3: NACT = IGNORE, the USART is not concerned by the response.
  - Wait until LINTC in US_CSR rises.
  - Check the LIN errors.

Figure 46-45. Master Node Configuration, NACT = PUBLISH



46.7.14 USART Interrupt Mask Register (SPI_MODE)

Name: US_IMR (SPI_MODE)
Offset: 0x0010
Reset: 0x0
Property: Read-only

This configuration is relevant only if USART_MODE = 0xE or 0xF in the USART Mode Register.

The following configuration values are valid for all listed bit names of this register:

0: The corresponding interrupt is not enabled.

1: The corresponding interrupt is enabled.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset					0			

Bit	15	14	13	12	11	10	9	8
Access								
Reset						0	0	

Bit	7	6	5	4	3	2	1	0
Access								
Reset			0				0	0

Bit 19 – NSSE NSS Line (Driving CTS Pin) Rising or Falling Edge Event Interrupt Mask

Bit 10 – UNRE SPI Underrun Error Interrupt Mask

Bit 9 – TXEMPTY TXEMPTY Interrupt Mask

Bit 5 – OVRE Overrun Error Interrupt Mask

Bit 1 – TXRDY TXRDY Interrupt Mask

Bit 0 – RXRDY RXRDY Interrupt Mask

48.7.5 MediaLB System Data Register

Name: MLB_MSD
Offset: 0x024
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	SD3[7:0]							
Access								
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	SD2[7:0]							
Access								
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	SD1[7:0]							
Access								
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	SD0[7:0]							
Access								
Reset	0	0	0	0	0	0	0	0

Bits 31:24 – SD3[7:0] System Data (Byte 3)
Updated with MediaLB Data[31:24] when a MediaLB software system command is received in the system quadlet. If MLB_MSS.SWSYSCMD is already set, then SD3 is not updated.

Bits 23:16 – SD2[7:0] System Data (Byte 2)
Updated with MediaLB Data[23:16] when a MediaLB software system command is received in the system quadlet. If MLB_MSS.SWSYSCMD is already set, then SD2 is not updated.

Bits 15:8 – SD1[7:0] System Data (Byte 1)
Updated with MediaLB Data[15:8] when a MediaLB software system command is received in the system quadlet. If MLB_MSS.SWSYSCMD is already set, then SD1 is not updated.

Bits 7:0 – SD0[7:0] System Data (Byte 0)
Updated with MediaLB Data[7:0] when a MediaLB software system command is received in the system quadlet. If MLB_MSS.SWSYSCMD is already set, then SD0 is not updated.

The waveform polarity must be set before enabling the channel. This immediately affects the channel output level.

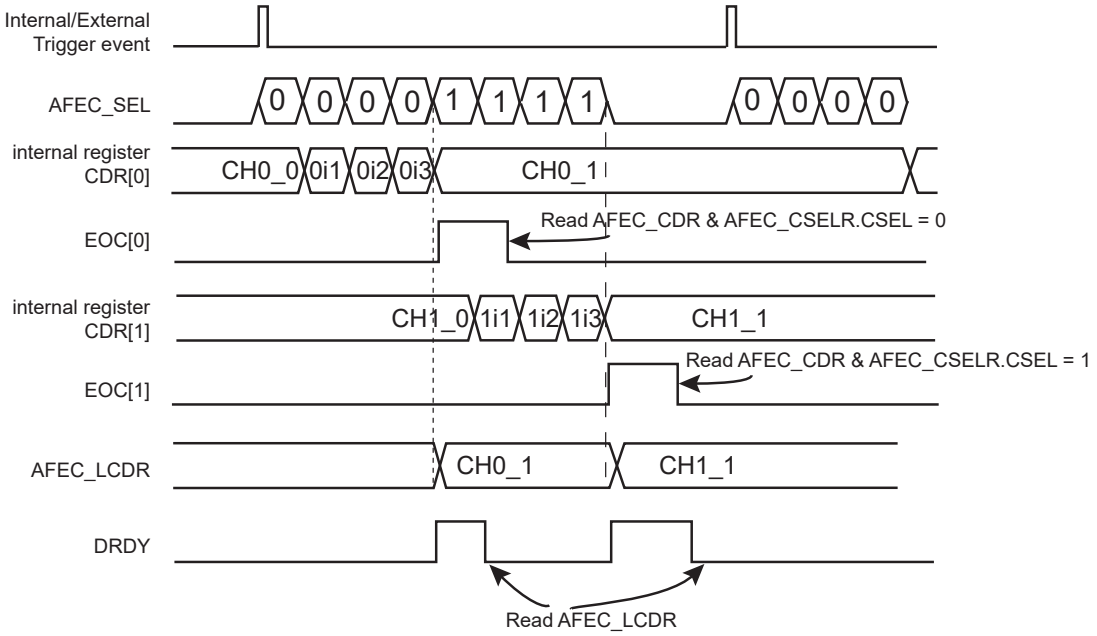
Modifying CPOL in PWM Channel Mode Register while the channel is enabled can lead to an unexpected behavior of the device being driven by PWM.

In addition to generating the output signals OCx, the comparator generates interrupts depending on the counter value. When the output waveform is left-aligned, the interrupt occurs at the end of the counter period. When the output waveform is center-aligned, the bit CES of PWM_CMRx defines when the channel counter interrupt occurs. If CES is set to ‘0’, the interrupt occurs at the end of the counter period. If CES is set to ‘1’, the interrupt occurs at the end of the counter period and at half of the counter period.

The figure below illustrates the counter interrupts depending on the configuration.

Figure 52-13. Digital Averaging Function Waveforms on a Single Trigger Event, Non-interleaved

AFEC_EMR.EMR = 2, STM = 1, AFEC_CHSR[7:0] = 0xFF and AFEC_MR.USEQ = 1
AFEC_SEQ1R = 0x1111_0000



Note: 0i1, 0i2, 0i3, 1i1, 1i2, 1i3 are intermediate results and CH0/1_0/1 are final results of average function.

52.6.15 Automatic Error Correction

The AFEC features automatic error correction of conversion results. Offset and gain error corrections are available. The correction can be enabled for each channel and correction values (offset and gain) are defined per Sample & Hold unit.

To enable error correction, the ECORR bit must be set in the AFEC Channel Error Correction register (AFEC_CECR). The offset and gain values used to compensate the results are set per Sample & Hold unit basis using the AFEC Correction Select register (AFEC_COSR) and the AFEC Correction Values register (AFEC_CVR). AFEC_COSR is used to select the Sample & Hold unit to be displayed in AFEC_CVR. This selection applies both to read and write operations in AFEC_CVR.

AFEC_CVR.OFFSETCORR and AFEC_CVR.GAINCORR must be filled with the values of corrective data. This data is computed from two measurement points in signed format. The correction is the same for all functional modes.

The final conversion result after error correction is obtained using the following formula, which is implemented after averaging in 2's complement format, with:

- OFFSETCORR—the offset correction value. OFFSETCORR is a signed value.
- GAINCORR—the gain correction value
- Gs—the value 15

$$\text{Corrected Data} = (\text{Converted Data} + \text{OFFSETCORR}) \times \frac{\text{GAINCORR}}{2^{(\text{Gs})}}$$

57.5.4 AES Interrupt Disable Register

Name: AES_IDR
Offset: 0x14
Reset: –
Property: Write-only

The following configuration values are valid for all listed bit names of this register:

0: No effect.

1: Disables the corresponding interrupt.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
								TAGRDY
Access								W
Reset								–

Bit	15	14	13	12	11	10	9	8
								URAD
Access								W
Reset								–

Bit	7	6	5	4	3	2	1	0
								DATRDY
Access								W
Reset								–

Bit 16 – TAGRDY GCM Tag Ready Interrupt Disable

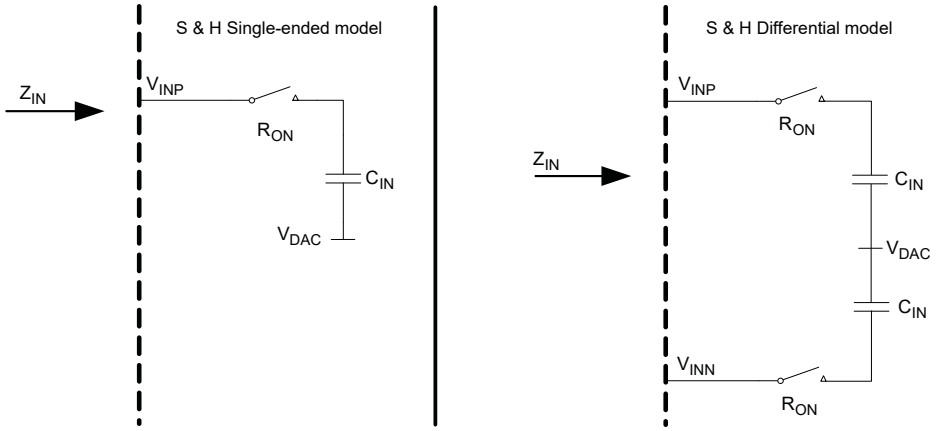
Bit 8 – URAD Unspecified Register Access Detection Interrupt Disable

Bit 0 – DATRDY Data Ready Interrupt Disable

Symbol	Parameter	Conditions	Min	Typ(1)	Max	Unit
		Gain=2	-0.3	0.3	1.4	
		Gain=4	-0.3	0.7	3.3	
Single-Ended Mode						
E_O	Single-ended Offset Error (see **Note 1**)	Gain=1	-20	–	35	LSB
E_G	Single-ended Gain Error	Gain=1	0.3	0.7	1.8	%
		Gain=2	0.3	1.3	3.6	
		Gain=4	0.3	1.7	4.7	

58.8.6 AFE Channel Input Impedance

Figure 58-15. Input Channel Model



where:

- Z_{IN} is input impedance in Single-ended or Differential mode
- C_{IN} = 2 to 8 pF ±20% depending on the gain value and mode (SE or DIFF); temperature dependency is negligible
- R_{ON} is typical 2 kΩ and 8 kΩ max (worst case process and high temperature)

The following formula is used to calculate input impedance:

$$Z_{\mathrm{IN}} = \frac{1}{f_S \times C_{\mathrm{IN}}}$$

where:

- f_S is the sampling frequency of the AFE channel
- Typ values are used to compute AFE input impedance Z_{IN}

Table 58-37. Input Capacitance (C_{IN}) Values

Gain Selection	Single-ended	Differential	Unit
1	2	2	pF
2	4	4	
4	8	8	