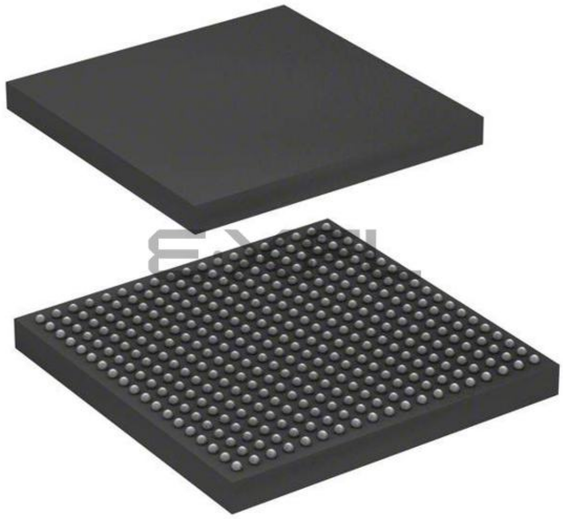
Microchip Technology - ATSAMA5D44B-CU Datasheet



Welcome to E-XFL.COM

Understanding Embedded - Microprocessors

Embedded microprocessors are specialized computing chips designed to perform specific tasks within an embedded system. Unlike general-purpose microprocessors found in personal computers, embedded microprocessors are tailored for dedicated functions within larger systems, offering optimized performance, efficiency, and reliability. These microprocessors are integral to the operation of countless electronic devices, providing the computational power necessary for controlling processes, handling data, and managing communications.

Applications of Embedded - Microprocessors

Embedded microprocessors are utilized across a broad spectrum of applications, making them indispensable in

Details	
Product Status	Active
Core Processor	ARM® Cortex®-A5
Number of Cores/Bus Width	1 Core, 32-Bit
Speed	600MHz
Co-Processors/DSP	Multimedia; NEON™ SIMD
RAM Controllers	LPDDR, LPDDR2, DDR2
Graphics Acceleration	Yes
Display & Interface Controllers	LCD, Touchscreen
Ethernet	10/100Mbps (1)
SATA	-
USB	USB 2.0 (3)
Voltage - I/O	1.2V, 1.8V, 3.3V
Operating Temperature	-40°C ~ 85°C (TA)
Security Features	AES, SHA, TDES, TRNG
Package / Case	361-TFBGA
Supplier Device Package	361-TFBGA (16x16)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/atsama5d44b-cu

17.9.17 AIC Interrupt Clear Command Register

Name:AIC_ICCR

Address:0xFC06E048 (AIC), 0xFC068448 (SAIC)

Access:Write-only

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–

23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–

15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	–

7	6	5	4	3	2	1	0
–	–	–	–	–	–	–	INTCLR

INTCLR: Interrupt Clear

Clears one the following depending on the setting of the INTSEL bit FIQ, SYS, PID2-PID127

0: No effect.

1: Clears the interrupt source selected by INTSEL.

22.6.13 RTC TimeStamp Time Register 0

Name: RTC_TSTR0

Address:0xFC068760

Access: Read-only

31	30	29	28	27	26	25	24
BACKUP	–	–	–	TEVCNT			
23	22	21	20	19	18	17	16
–	AMPM	HOUR					
15	14	13	12	11	10	9	8
–	MIN						
7	6	5	4	3	2	1	0
–	SEC						

RTC_TSTR0 reports the timestamp of the first tamper event after reading RTC_TSSR0.

This register is cleared by reading RTC_TSSR0.

SEC: Seconds of the Tamper

MIN: Minutes of the Tamper

HOUR: Hours of the Tamper

AMPM: AM/PM Indicator of the Tamper

TEVCNT: Tamper Events Counter

Each time a tamper event occurs, this counter is incremented. This counter saturates at 15. Once this value is reached, it is no more possible to know the exact number of tamper events.

If this field is not null, this implies that at least one tamper event occurs since last register reset and that the values stored in timestamping registers are valid.

BACKUP: System Mode of the Tamper

0: The state of the system is different from backup mode when the tamper event occurs.

1: The system is in backup mode when the tamper event occurs.

For example, if fuses 0 to 31 must be programmed, Data Register 0 (SFC_DR0) must be written. If fuses 32 to 61 must be programmed, Data Register 1 (SFC_DR1) must be written. Only the data bits set to level ‘1’ are programmed.

3. Wait for flag PGMC to rise in the Status Register (SFC_SR) by polling or interrupt.
4. Check the value of flag PGMF: if it is set to 1, it means that the programming procedure failed. After programming, the fuses are read back in the corresponding SFC_DRx.

25.4.4.3 Fuse Masking

It is possible to mask a fuse array. Once the fuse masking is enabled, the data registers from SFC_DRx to SFC_DRx are read at a value of ‘0’, regardless of the fuse state (the registers that are masked depend on the SFC hardware customizing).

To activate fuse masking, the MSK bit of the SFC Mode Register (SFC_MR) must be written to level ‘1’. The MSK bit is set-only. Only a hardware reset can disable fuse masking.

The MSK bit has no effect on the programming of masked fuses.

25.4.5 Fuse Functions

The “Fuse Box Controller” section defines the fuse bits that can be used as general purpose bits when standard boot is used.

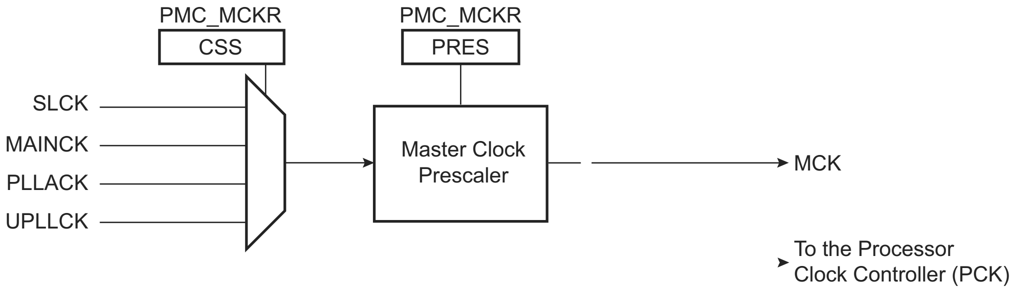
If secure boot is used, refer to the device “Secure Boot Strategy” application note included in the Secure Package.

The Master clock selection is made by writing the CSS (Clock Source Selection) field in the Master Clock register (PMC_MCKR). The prescaler supports the division by a power of 2 of the selected clock between 1 and 64, and the division by 6. PMC_MCKR.PRES programs the prescaler.

Note: It is forbidden to modify MDIV and CSS at the same access. Each field must be modified separately with a wait for MCKRDY flag between the first field modification and the second field modification.

Each time PMC_MCKR is written to define a new Master clock, PMC_SR.MCKRDY is cleared. It reads 0 until the Master clock is established. Then, the MCKRDY bit is set and can trigger an interrupt to the processor. This feature is useful when switching from a high-speed clock to a lower one to inform the software when the change is actually done.

Figure 27-2: Master Clock Controller



27.5 Processor Clock Controller

The PMC features a Processor Clock (PCK) Controller that implements the processor Idle mode.

The Processor clock can be disabled by executing the WFI (WaitForInterrupt) processor instruction.

The Processor clock can be disabled by writing the PMC System Clock Disable Register (PMC_SCDR). The status of this clock (at least for debug purposes) can be read in the PMC System Clock Status Register (PMC_SCSR).

The Processor clock is enabled after a reset and is automatically re-enabled by any enabled interrupt. The processor Idle mode is entered by disabling the Processor clock, which is automatically re-enabled by any enabled fast or normal interrupt, or by the reset of the product.

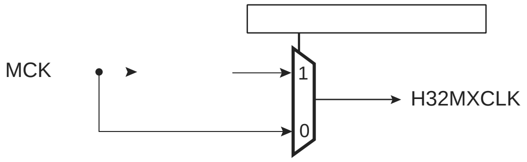
When processor Idle mode is entered, the current instruction is finished before the clock is stopped, but this does not prevent data transfers from other masters of the system bus.

27.6 Matrix Clock Controller

The AXI Matrix and H64MX 64-bit Matrix clocks are MCK.

The H32MX 32-bit matrix clock is to be configured as MCK if MCK does not exceed 100 MHz (refer to "Section 56.4.2 "Master Clock Characteristics"); otherwise, this clock is to be configured as MCK/2. Selection is done with the H32MXDIV bit in PMC Master Clock Register.

Figure 27-3: H32MX 32-bit Matrix Clock Configuration



27.7 Peripheral Clock Controller

The PMC controls the clocks of each embedded peripheral by means of the Peripheral Clock Controller. The user can individually enable and disable the clock on the peripherals and select a division factor from MCK. This is done in the Peripheral Control register (PMC_PCR).

In order to reduce power consumption, the division factor can be 1, 2, 4 or 8.

The divisor is defined in PMC_PCR. To apply a division factor, PID, CMD and DIV must be written in a single operation. The target peripheral clock is defined by the PID field. The divisor value is defined by DIV and the bit CMD must be set. To read the current division factor associated with a peripheral clock, two separate operations must be performed:

1. Write a zero to the CMD bit and configure PID for the target peripheral clock. DIV is not significant for this operation.

28.5.4 Output Control

When the I/O line is assigned to a peripheral function, i.e., the corresponding bit in PIO_PSR is at zero, the drive of the I/O line is controlled by the peripheral. Peripheral A or B or C or D depending on the value in PIO_ABCDSR1 and PIO_ABCDSR2 determines whether the pin is driven or not.

When the I/O line is controlled by the PIO Controller, the pin can be configured to be driven. This is done by writing the Output Enable Register (PIO_OER) and Output Disable Register (PIO_ODR). The results of these write operations are detected in the Output Status Register (PIO_OSR). When a bit in this register is at zero, the corresponding I/O line is used as an input only. When the bit is at one, the corresponding I/O line is driven by the PIO Controller.

The level driven on an I/O line can be determined by writing in the Set Output Data Register (PIO_SODR) and the Clear Output Data Register (PIO_CODR). These write operations, respectively, set and clear the Output Data Status Register (PIO_ODSR), which represents the data driven on the I/O lines. Writing in PIO_OER and PIO_ODR manages PIO_OSR whether the pin is configured to be controlled by the PIO Controller or assigned to a peripheral function. This enables configuration of the I/O line prior to setting it to be managed by the PIO Controller.

Similarly, writing in PIO_SODR and PIO_CODR affects PIO_ODSR. This is important as it defines the first level driven on the I/O line.

28.5.5 Synchronous Data Output

Clearing one or more PIO line(s) and setting another one or more PIO line(s) synchronously cannot be done by using PIO_SODR and PIO_CODR. It requires two successive write operations into two different registers. To overcome this, the PIO Controller offers a direct control of PIO outputs by single write access to PIO_ODSR. Only bits unmasked by the Output Write Status Register (PIO_OWSR) are written. The mask bits in PIO_OWSR are set by writing to the Output Write Enable Register (PIO_OWER) and cleared by writing to the Output Write Disable Register (PIO_OWDR).

After reset, the synchronous data output is disabled on all the I/O lines as PIO_OWSR resets at 0x0.

28.5.6 Multi-Drive Control (Open Drain)

Each I/O can be independently programmed in open drain by using the multi-drive feature. This feature permits several drivers to be connected on the I/O line which is driven low only by each device. An external pullup resistor (or enabling of the internal one) is generally required to guarantee a high level on the line.

The multi-drive feature is controlled by the Multi-driver Enable Register (PIO_MDER) and the Multi-driver Disable Register (PIO_MDDR). The multi-drive can be selected whether the I/O line is controlled by the PIO Controller or assigned to a peripheral function. The Multi-driver Status Register (PIO_MDSR) indicates the pins that are configured to support external drivers.

After reset, the multi-drive feature is disabled on all pins, i.e., PIO_MDSR resets at value 0x0.

28.5.7 Output Line Timings

Figure 28-3 shows how the outputs are driven either by writing PIO_SODR or PIO_CODR, or by directly writing PIO_ODSR. This last case is valid only if the corresponding bit in PIO_OWSR is set. Figure 28-3 also shows when the feedback in the Pin Data Status Register (PIO_PDSR) is available.

Figure 28-3: Output Line Timings

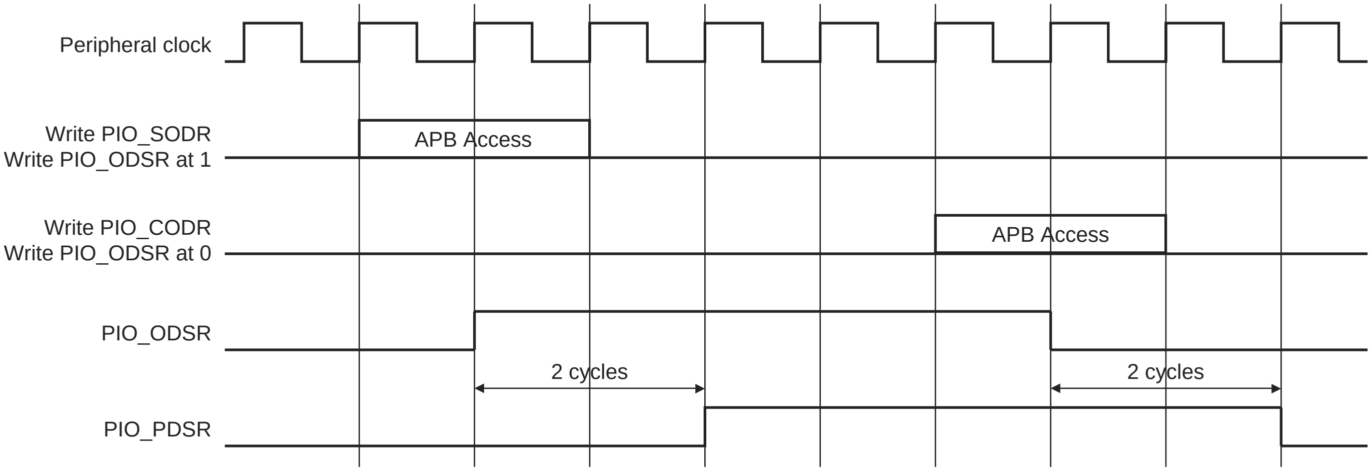


Table 29-20: Interleaved Mapping DDR-SDRAM Configuration Mapping: 8K Rows /512/1024/2048 Columns, 4 banks

CPU Address Line																												
28	**27**	**26**	**25**	**24**	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
			Row[13:0]														Bk[1:0]		Column[8:0]								M[1:0]	
		Row[13:0]														Bk[1:0]		Column[9:0]									M[1:0]	
	Row[13:0]														Bk[1:0]		Column[10:0]										M[1:0]	

Table 29-21: Sequential Mapping DDR-SDRAM Configuration Mapping: 8K Rows /1024 Columns, 8 banks

CPU Address Line																												
28	**27**	**26**	**25**	**24**	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
	Bk[2:0]			Row[12:0]													Column[9:0]										M[1:0]	

Table 29-22: Interleaved Mapping DDR-SDRAM Configuration Mapping: 8K Rows /1024 Columns, 8 banks

CPU Address Line																												
28	**27**	**26**	**25**	**24**	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
	Row[12:0]													Bk[2:0]			Column[9:0]										M[1:0]	

Table 29-23: Sequential Mapping DDR-SDRAM Configuration Mapping: 16K Rows /1024 Columns, 4 banks

CPU Address Line																												
28	**27**	**26**	**25**	**24**	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
	Bk[1:0]		Row[13:0]														Column[9:0]										M[1:0]	

Table 29-24: Interleaved Mapping DDR-SDRAM Configuration Mapping: 16K Rows /1024 Columns, 4 banks

CPU Address Line																												
28	**27**	**26**	**25**	**24**	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
	Row[13:0]														Bk[1:0]		Column[9:0]										M[1:0]	

Table 29-25: Sequential Mapping DDR-SDRAM Configuration Mapping: 16K Rows /1024 Columns, 8 banks

CPU Address Line																												
28	**27**	**26**	**25**	**24**	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bk[2:0]			Row[13:0]														Column[9:0]										M[1:0]	

Table 29-26: Interleaved Mapping DDR-SDRAM Configuration Mapping: 16K Rows /1024 Columns, 8 banks

CPU Address Line																												
28	**27**	**26**	**25**	**24**	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Row[13:0]														Bk[2:0]			Column[9:0]										M[1:0]	

Table 30-20: Register Mapping (Continued)

Offset	Register	Name	Access	Reset
0x14*CS_number+0x604	Pulse Register	HSMC_PULSE	Read/Write	0x0101_0101
0x14*CS_number+0x608	Cycle Register	HSMC_CYCLE	Read/Write	0x0003_0003
0x14*CS_number+0x60C	Timings Register	HSMC_TIMINGS	Read/Write	0x0000_0000
0x14*CS_number+0x610	Mode Register	HSMC_MODE	Read/Write	0x0000_1003
0x6A0	Off Chip Memory Scrambling Register	HSMC_OCMS	Read/Write	0x0
0x6A4	Off Chip Memory Scrambling KEY1 Register	HSMC_KEY1	Write-once	0x0
0x6A8	Off Chip Memory Scrambling KEY2 Register	HSMC_KEY2	Write-once	0x0
0x6AC–0x6E0	Reserved	–	–	–
0x6E4	Write Protection Mode Register	HSMC_WPMR	Read/Write	0x0
0x6E8	Write Protection Status Register	HSMC_WPSR	Read-only	0x0
0x6EC–0x6FC	Reserved	–	–	–

Table 32-55: Register Mapping (Continued)

Offset	Register	Name	Access	Reset
0x00000140	Overlay 1 Channel Enable Register	LCDC_OVR1CHER	Write-only	–
0x00000144	Overlay 1 Channel Disable Register	LCDC_OVR1CHDR	Write-only	–
0x00000148	Overlay 1 Channel Status Register	LCDC_OVR1CHSR	Read-only	0x00000000
0x0000014C	Overlay 1 Interrupt Enable Register	LCDC_OVR1IER	Write-only	–
0x00000150	Overlay 1 Interrupt Disable Register	LCDC_OVR1IDR	Write-only	–
0x00000154	Overlay 1 Interrupt Mask Register	LCDC_OVR1IMR	Read-only	0x00000000
0x00000158	Overlay 1 Interrupt Status Register	LCDC_OVR1ISR	Read-only	0x00000000
0x0000015C	Overlay 1 DMA Head Register	LCDC_OVR1HEAD	Read/Write	0x00000000
0x00000160	Overlay 1 DMA Address Register	LCDC_OVR1ADDR	Read/Write	0x00000000
0x00000164	Overlay 1 DMA Control Register	LCDC_OVR1CTRL	Read/Write	0x00000000
0x00000168	Overlay 1 DMA Next Register	LCDC_OVR1NEXT	Read/Write	0x00000000
0x0000016C	Overlay 1 Configuration Register 0	LCDC_OVR1CFG0	Read/Write	0x00000000
0x00000170	Overlay 1 Configuration Register 1	LCDC_OVR1CFG1	Read/Write	0x00000000
0x00000174	Overlay 1 Configuration Register 2	LCDC_OVR1CFG2	Read/Write	0x00000000
0x00000178	Overlay 1 Configuration Register 3	LCDC_OVR1CFG3	Read/Write	0x00000000
0x0000017C	Overlay 1 Configuration Register 4	LCDC_OVR1CFG4	Read/Write	0x00000000
0x00000180	Overlay 1 Configuration Register 5	LCDC_OVR1CFG5	Read/Write	0x00000000
0x00000184	Overlay 1 Configuration Register 6	LCDC_OVR1CFG6	Read/Write	0x00000000
0x00000188	Overlay 1 Configuration Register 7	LCDC_OVR1CFG7	Read/Write	0x00000000
0x0000018C	Overlay 1 Configuration Register 8	LCDC_OVR1CFG8	Read/Write	0x00000000
0x00000190	Overlay 1 Configuration Register 9	LCDC_OVR1CFG9	Read/Write	0x00000000
0x00000194–0x0000023C	Reserved	–	–	–
0x00000240	Overlay 2 Channel Enable Register	LCDC_OVR2CHER	Write-only	–
0x00000244	Overlay 2 Channel Disable Register	LCDC_OVR2CHDR	Write-only	–
0x00000248	Overlay 2 Channel Status Register	LCDC_OVR2CHSR	Read-only	0x00000000
0x0000024C	Overlay 2 Interrupt Enable Register	LCDC_OVR2IER	Write-only	–
0x00000250	Overlay 2 Interrupt Disable Register	LCDC_OVR2IDR	Write-only	–
0x00000254	Overlay 2 Interrupt Mask Register	LCDC_OVR2IMR	Read-only	0x00000000
0x00000258	Overlay 2 Interrupt Status Register	LCDC_OVR2ISR	Read-only	0x00000000
0x0000025C	Overlay 2 DMA Head Register	LCDC_OVR2HEAD	Read/Write	0x00000000
0x00000260	Overlay 2 DMA Address Register	LCDC_OVR2ADDR	Read/Write	0x00000000
0x00000264	Overlay 2 DMA Control Register	LCDC_OVR2CTRL	Read/Write	0x00000000
0x00000268	Overlay 2 DMA Next Register	LCDC_OVR2NEXT	Read/Write	0x00000000
0x0000026C	Overlay 2 Configuration Register 0	LCDC_OVR2CFG0	Read/Write	0x00000000
0x00000270	Overlay 2 Configuration Register 1	LCDC_OVR2CFG1	Read/Write	0x00000000
0x00000274	Overlay 2 Configuration Register 2	LCDC_OVR2CFG2	Read/Write	0x00000000

32.7.10 LCD Controller Status Register

Name: LCDC_LCDSR

Address:0xF0000028

Access: Read-only

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
–	–	–	–	–	–	–	–
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
–	–	–	–	–	–	–	–
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
–	–	–	SIPSTS	PWMSTS	DISPSTS	LCDSTS	CLKSTS

CLKSTS: Clock Status

0: Pixel clock is disabled.

1: Pixel clock is running.

LCDSTS: LCD Controller Synchronization status

0: Timing engine is disabled.

1: Timing engine is running.

DISPSTS: LCD Controller DISP Signal Status

0: DISP is disabled.

1: DISP signal is activated.

PWMSTS: LCD Controller PWM Signal Status

0: PWM is disabled.

1: PWM signal is activated.

SIPSTS: Synchronization In Progress

0: Clock domain synchronization is terminated.

1: Synchronization is in progress. Access to the registers LCDC_LCDCCFG[0..6], LCDC_LCDEN and LCDC_LCDDIS has no effect.

34.5.5 Codec Path

34.5.5.1 Color Space Conversion

Depending on user selection, this module can be bypassed so that input YCrCb stream is directly connected to the format converter module. If the RGB input stream is selected, this module converts RGB to YCrCb color space with the formulas given below:

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

An example of coefficients is given below:

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

34.5.5.2 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.

34.5.5.3 DMA Features

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

Figure 35-14: Data OUT Transfer for Endpoint with One Bank

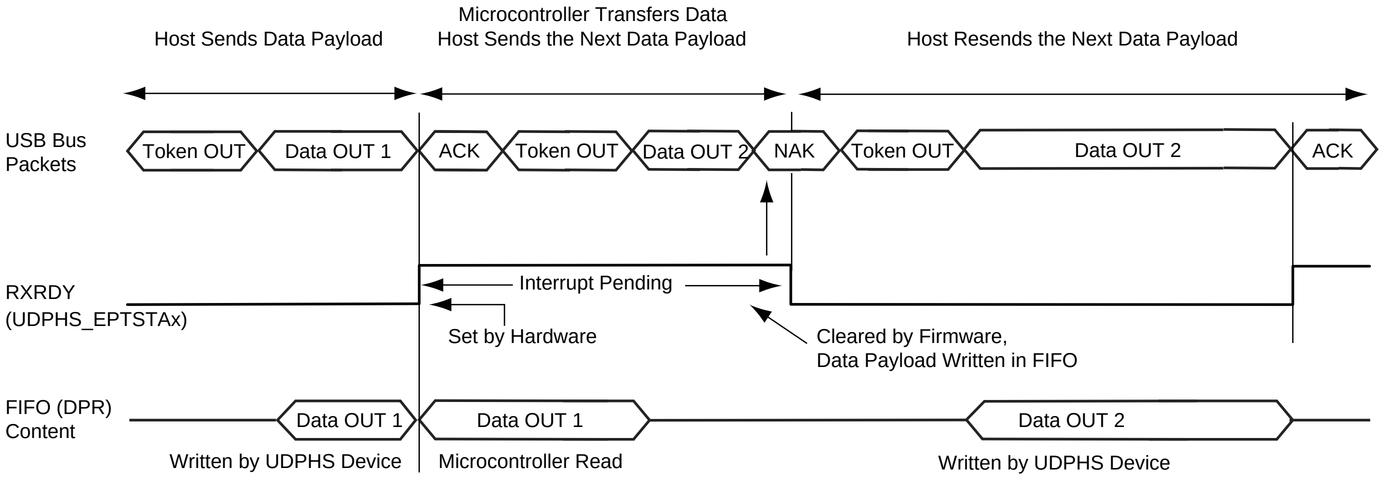
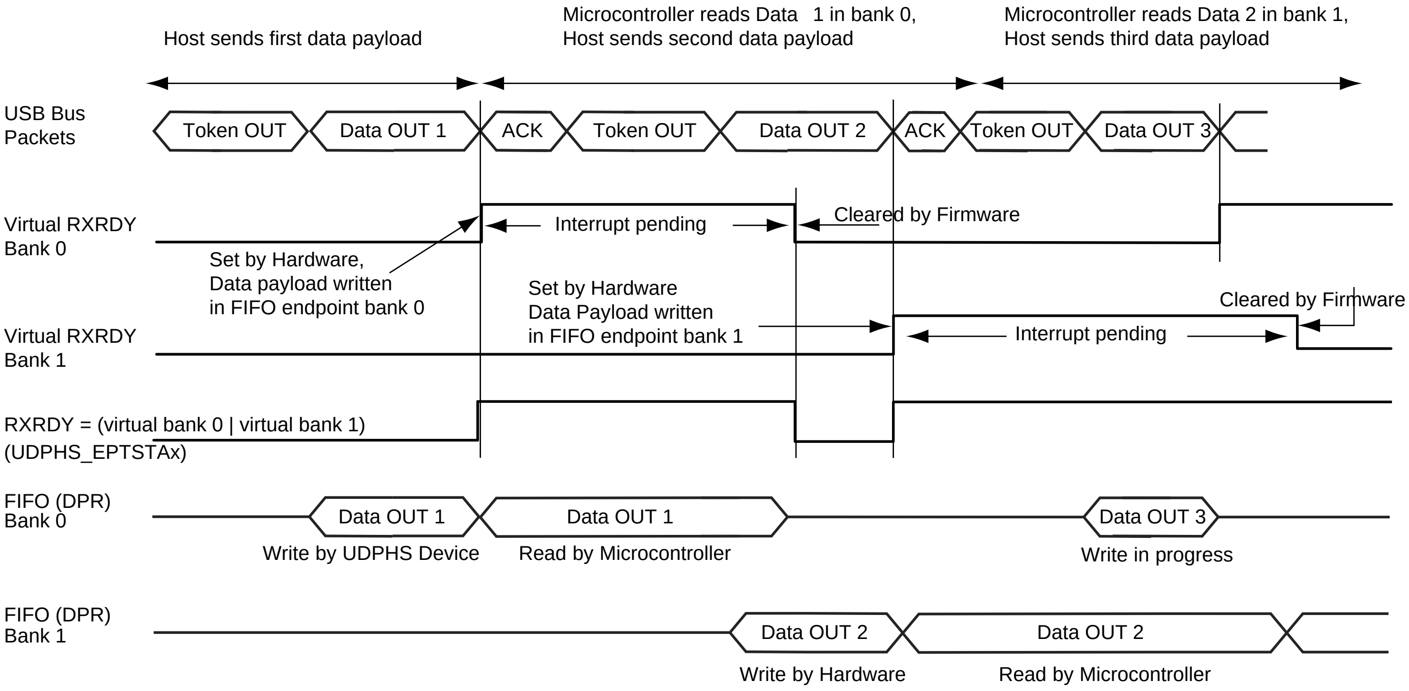


Figure 35-15: Data OUT Transfer for an Endpoint with Two Banks



High Bandwidth Isochronous Endpoint OUT

USB 2.0 supports individual High Speed isochronous endpoints that require data rates up to 192 Mb/s (24 MB/s): 3x1024 data bytes per microframe.

To support such a rate, two or three banks may be used to buffer the three consecutive data packets. The microcontroller (or the DMA) should be able to empty the banks very rapidly (at least 24 MB/s on average).

NB_TRANS field in UDPHS_EPTCFGx register = Number Of Transactions per Microframe.

If NB_TRANS > 1 then it is High Bandwidth.

36.7.12 UHPHS Port Status and Control Register

Name:UHPHS_PORTSC_x[x = 0..2]

Access: Read/Write

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–

23	22	21	20	19	18	17	16
–	WKOC_E	WKDSCNNT_E	WKCNNT_E	PTC			

15	14	13	12	11	10	9	8
PIC		PO	PP	LS		–	PR

7	6	5	4	3	2	1	0
SUS	FPR	OCC	OCA	PEDC	PED	CSC	CCS

A host controller must implement one or more port registers. The number of port registers implemented by a particular instantiation of a host controller is documented in the UHPHS_HCSPARAMS register (**Section 36.7.2 "UHPHS Host Controller Structural Parameters Register"**). Software uses this information as an input parameter to determine how many ports need to be serviced. All ports have the structure defined below.

This register is in the auxiliary power well. It is only reset by hardware when the auxiliary power is initially applied or in response to a host controller reset. The initial conditions of a port are:

- No device connected
- Port disabled

If the port has port power control, software cannot change the state of the port until after it applies power to the port by setting port power to a 1. Software must not attempt to change the state of the port until after power is stable on the port. The host is required to have power stable to the port within 20 milliseconds of the 0 to 1 transition.

Note 1: When a device is attached, the port state transitions to the connected state and system software will process this as with any status change notification.

2: If a port is being used as the Debug Port, then the port may report device connected and enabled when the Configured Flag is set to 0.

CCS: Current Connect Status (read-only)

0: No device is present (default value).

1: Device is present on port.

This value reflects the current state of the port, and may not correspond directly to the event that caused the Connect Status Change bit (Bit 1) to be set.

This field is 0 if Port Power is 0.

CSC: Connect Status Change (read/write clear)

0: No change (default value).

1: Change in Current Connect Status.

Indicates a change has occurred in the port's Current Connect Status. The host controller sets this bit for all changes to the port device connect status, even if system software has not cleared an existing connect status change. For example, the insertion status changes twice before system software has cleared the changed condition, hub hardware will be "setting" an already-set bit (i.e., the bit will remain set). Software sets this bit to 0 by writing a 1 to it.

This field is 0 if Port Power is 0.

PED: Port Enabled/Disabled (read/write)

0: Disable (default value).

1: Enable.

37.6.16.1 802.3 Pause Frame Reception

Bit 13 of the Network Configuration register is the pause enable control for reception. If this bit is set, transmission pauses if a non zero pause quantum frame is received.

If a valid pause frame is received, then the Pause Time register is updated with the new frame's pause time, regardless of whether a previous pause frame is active or not. An interrupt (either bit 12 or bit 13 of the Interrupt Status register) is triggered when a pause frame is received, but only if the interrupt has been enabled (bit 12 and bit 13 of the Interrupt Mask register). Pause frames received with non zero quantum are indicated through the interrupt bit 12 of the Interrupt Status register. Pause frames received with zero quantum are indicated on bit 13 of the Interrupt Status register.

Once the Pause Time register is loaded and the frame currently being transmitted has been sent, no new frames are transmitted until the pause time reaches zero. The loading of a new pause time, and hence the pausing of transmission, only occurs when the GMAC is configured for full duplex operation. If the GMAC is configured for half duplex there will be no transmission pause, but the pause frame received interrupt will still be triggered. A valid pause frame is defined as having a destination address that matches either the address stored in Specific Address 1 register or if it matches the reserved address of 0x0180C2000001. It must also have the MAC control frame type ID of 0x8808 and have the pause opcode of 0x0001.

Pause frames that have frame check sequence (FCS) or other errors will be treated as invalid and will be discarded. 802.3 Pause frames that are received after Priority-based Flow Control (PFC) has been negotiated will also be discarded. Valid pause frames received will increment the Pause Frames Received statistic register.

The Pause Time register decrements every 512 bit times once transmission has stopped. For test purposes, the retry test bit can be set (bit 12 in the Network Configuration register) which causes the Pause Time register to decrement every GTXCK cycle once transmission has stopped.

The interrupt (bit 13 in the Interrupt Status register) is asserted whenever the Pause Time register decrements to zero (assuming it has been enabled by bit 13 in the Interrupt Mask register). This interrupt is also set when a zero quantum pause frame is received.

37.6.16.2 802.3 Pause Frame Transmission

Automatic transmission of pause frames is supported through the transmit pause frame bits of the Network Control register. If either bit 11 or bit 12 of the Network Control register is written with logic 1, an 802.3 pause frame will be transmitted, providing full duplex is selected in the Network Configuration register and the transmit block is enabled in the Network Control register.

Pause frame transmission will happen immediately if transmit is inactive or if transmit is active between the current frame and the next frame due to be transmitted.

Transmitted pause frames comprise the following:

- A destination address of 01-80-C2-00-00-01
- A source address taken from Specific Address 1 register
- A type ID of 88-08 (MAC control frame)
- A pause opcode of 00-01
- A Pause Quantum register
- Fill of 00 to take the frame to minimum frame length
- Valid FCS

The pause quantum used in the generated frame will depend on the trigger source for the frame as follows:

- If bit 11 is written with a one, the pause quantum will be taken from the Transmit Pause Quantum register. The Transmit Pause Quantum register resets to a value of 0xFFFF giving maximum pause quantum as default.
- If bit 12 is written with a one, the pause quantum will be zero.

After transmission, a pause frame transmitted interrupt will be generated (bit 14 of the Interrupt Status register) and the only the statistics register Pause Frames Transmitted is incremented.

Pause frames can also be transmitted by the MAC using normal frame transmission methods.

37.6.17 MAC PFC Priority-based Pause Frame Support

Note: Refer to the 802.1Qbb standard for a full description of priority-based pause operation.

The following table shows the start of a Priority-based Flow Control (PFC) pause frame.

37.8.53 GMAC 128 to 255 Byte Frames Transmitted Register

Name:GMAC_TBFT255

Address:0xF8020120 (0), 0xFC028120 (1)

Access: Read-only

31	30	29	28	27	26	25	24
NFTX							

23	22	21	20	19	18	17	16
NFTX							

15	14	13	12	11	10	9	8
NFTX							

7	6	5	4	3	2	1	0
NFTX							

NFTX: 128 to 255 Byte Frames Transmitted without Error

This register counts the number of 128 to 255 byte frames successfully transmitted without error, i.e., no underrun and not too many retries.

37.8.82 GMAC Length Field Frame Errors Register

Name:GMAC_LFFE

Address:0xF8020194 (0), 0xFC028194 (1)

Access: Read-only

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–

23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–

15	14	13	12	11	10	9	8
–	–	–	–	–	–	LFER	

7	6	5	4	3	2	1	0
LFER							

LFER: Length Field Frame Errors

This register counts the number of frames received that have a measured length shorter than that extracted from the length field (bytes 13 and 14). This condition is only counted if the value of the length field is less than 0x0600, the frame is not of excessive length and checking is enabled through bit 16 of the Network Configuration Register. Refer to Section 37.8.2 “GMAC Network Configuration Register”.

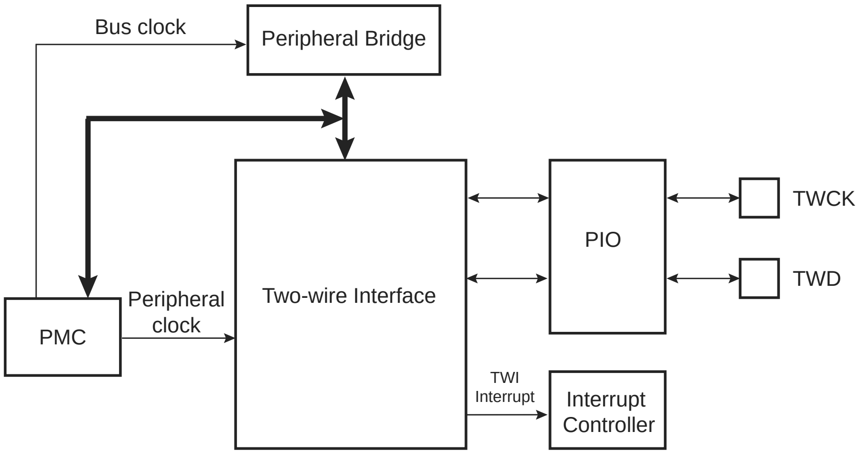
40.3 List of Abbreviations

Table 40-2: Abbreviations

Abbreviation	Description
TWI	Two-wire Interface
A	Acknowledge
NA	Non Acknowledge
P	Stop
S	Start
Sr	Repeated Start
SADR	Slave Address
ADR	Any address except SADR
R	Read
W	Write

40.4 Block Diagram

Figure 40-1: Block Diagram



40.5 I/O Lines Description

Table 40-3: I/O Lines Description

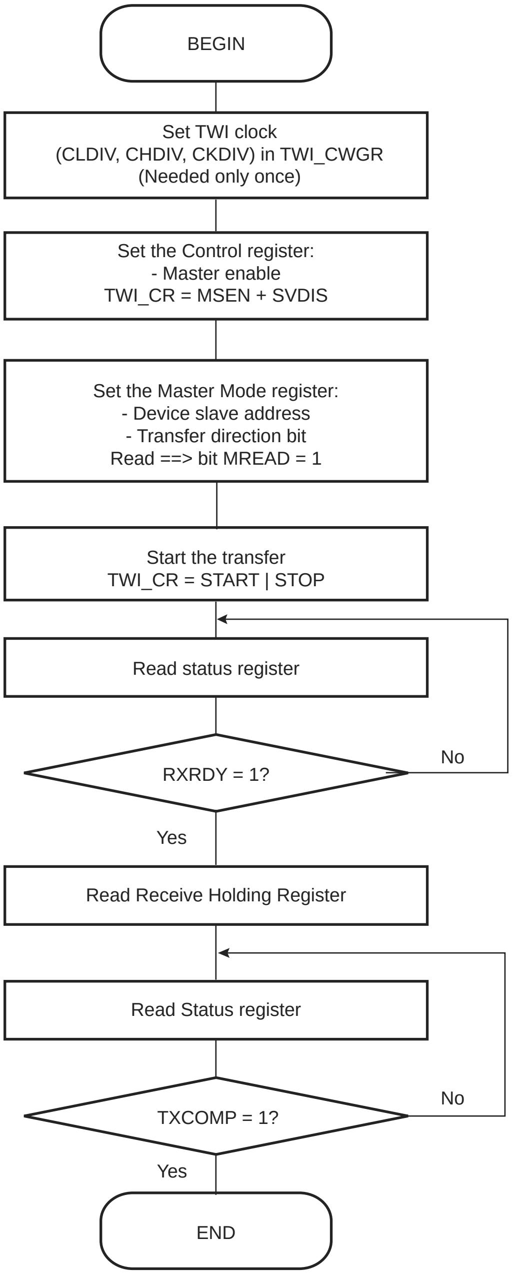
Name	Description	Type
TWD	Two-wire Serial Data (drives external serial data line – SDA)	Input/Output
TWCK	Two-wire Serial Clock (drives external serial clock line – SCL)	Input/Output

40.6 Product Dependencies

40.6.1 I/O Lines

Both TWD and TWCK are bidirectional lines, connected to a positive supply voltage via a current source or pull-up resistor. When the bus is free, both lines are high. The output stages of devices connected to the bus must have an open-drain or open-collector to perform the wired-AND function.

Figure 40-16: TWI Read Operation with Single Data Byte without Internal Address



Value	Name	Description
15	–	Reserved

NVPSIZ2: Second Nonvolatile Program Memory Size

Value	Name	Description
0	NONE	None
1	8K	8 Kbytes
2	16K	16 Kbytes
3	32K	32 Kbytes
4	–	Reserved
5	64K	64 Kbytes
6		Reserved
7	128K	128 Kbytes
8	–	Reserved
9	256K	256 Kbytes
10	512K	512 Kbytes
11	–	Reserved
12	1024K	1024 Kbytes
13	–	Reserved
14	2048K	2048 Kbytes
15	–	Reserved

SRAMSIZ: Internal SRAM Size

Value	Name	Description
0	–	Reserved
1	1K	1 Kbytes
2	2K	2 Kbytes
3	6K	6 Kbytes
4	112K	112 Kbytes
5	4K	4 Kbytes
6	80K	80 Kbytes
7	160K	160 Kbytes
8	8K	8 Kbytes
9	16K	16 Kbytes
10	32K	32 Kbytes
11	64K	64 Kbytes
12	128K	128 Kbytes

48.7.22 ADC Write Protection Mode Register

Name:ADC_WPMR

Address:0xFC0340E4

Access: Read/Write

31	30	29	28	27	26	25	24
WPKEY							
23	22	21	20	19	18	17	16
WPKEY							
15	14	13	12	11	10	9	8
WPKEY							
7	6	5	4	3	2	1	0
–	–	–	–	–	–	–	WPEN

WPEN: Write Protection Enable

0: Disables the write protection if WPKEY value corresponds to 0x414443 (“ADC” in ASCII).

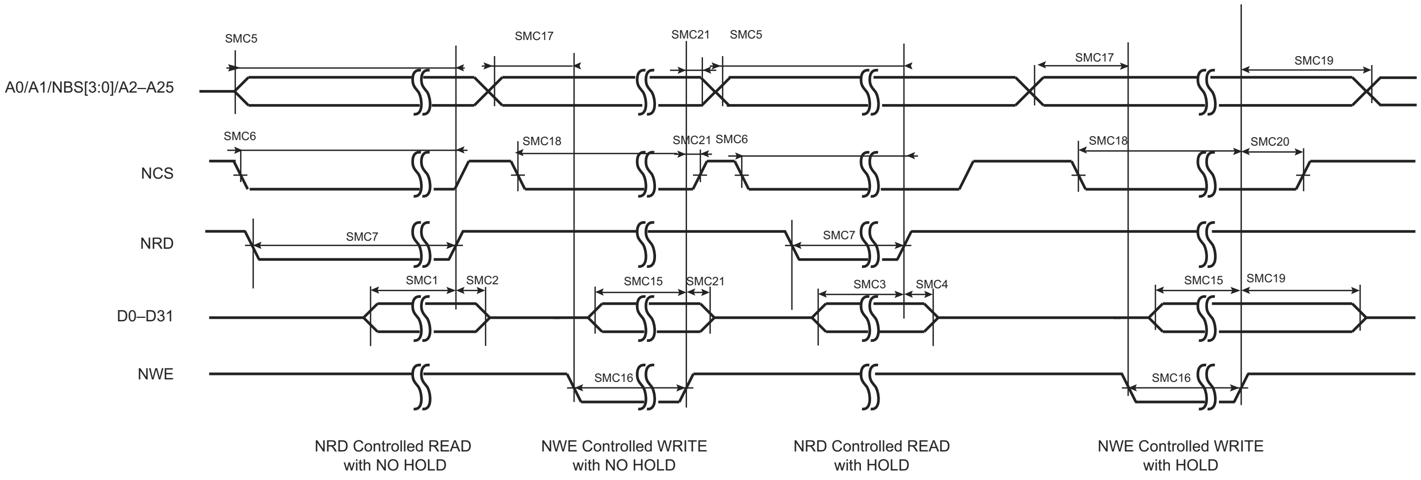
1: Enables the write protection if WPKEY value corresponds to 0x414443 (“ADC” in ASCII).

See Section 48.6.14 “Register Write Protection” for the list of write-protected registers.

WPKEY: Write Protection Key

Value	Name	Description
0x414443	PASSWD	Writing any other value in this field aborts the write operation of the WPEN bit. Always reads as 0

Figure 56-5: SMC Timings - NRD Controlled Read and NWE Controlled Write



56.14 SPI Timings

56.14.1 Maximum SPI Frequency

The following formulas give maximum SPI frequency in Master read and write modes and in Slave read and write modes.

- Master Write Mode

  The SPI is only sending data to a slave device such as an LCD, for example. The limit is given by SPI_2 (or SPI_5) timing.

- Master Read Mode

  $$f_{SPCK}Max = \frac{1}{SPI_0(\text{or } SPI_3) + t_{valid}}$$

  t_{valid} is the slave time response to output data after deleting an SPCK edge. For a non-volatile memory with t_{valid} (or t_v) = 12 ns, f_{SPCK}max = 45 MHz at V_{DDIO} = 3.3V.

- Slave Read Mode

  In slave mode, SPCK is the input clock for the SPI. The max SPCK frequency is given by setup and hold timings SPI_7/SPI_8 (or SPI_{10}/SPI_{11}). Since this gives a frequency well above the pad limit, the limit in slave read mode is given by SPCK pad.

- Slave Write Mode

  $$f_{SPCK}Max = \frac{1}{SPI_6(\text{or } SPI_9) + t_{setup}}$$

  t_{setup} is the setup time from the master before sampling data (12 ns).

  This gives f_{SPCK}Max = 45 MHz @ V_{DDIO} = 3.3V.