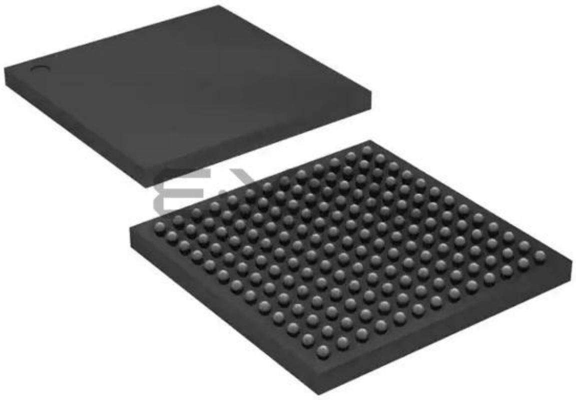
NXP USA Inc. - MCIMX233DJM4C Datasheet



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	ARM926EJ-S
Number of Cores/Bus Width	1 Core, 32-Bit
Speed	454MHz
Co-Processors/DSP	Data; DCP
RAM Controllers	DRAM
Graphics Acceleration	No
Display & Interface Controllers	LCD, Touchscreen
Ethernet	-
SATA	-
USB	USB 2.0 + PHY (1)
Voltage - I/O	2.0V, 2.5V, 2.7V, 3.0V, 3.3V
Operating Temperature	-10°C ~ 70°C (TA)
Security Features	Cryptography, Hardware ID
Package / Case	169-LFBGA
Supplier Device Package	169-MAPBGA (11x11)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/mcimx233djm4c

Contents

Paragraph Number | Title | Page Number

Chapter 27
Debug UART

Chapter 28
AUDIOIN/ADC

Chapter 29
AUDIOOUT/DAC

Contents

Paragraph Number | **Title** | **Page Number**

Chapter 38 Digital Video Encoder Programmers' Manual

Chapter 39
Register Macro Usage

Figures

Figure Number | Title | Page Number

Table 5-29. HW_ICOLL_INTERRUPT4 Bit Field Descriptions

BITS	LABEL	RW	RESET	DEFINITION
31:5	**RSRVD1**	RO	0x0	Always write zeroes to this bitfield.

4	**ENFIQ**	RW	0x0	Set this to 1 to steer this interrupt to the non-vectored FIQ line. When set to 0 the interrupt will pass through the main IRQ FSM and priority logic.
				DISABLE = 0x0 Disable
				ENABLE = 0x1 Enable

3	**SOFTIRQ**	RW	0x0	Set this bit to one to force a software interrupt. NO_INTERRUPT = 0x0 turn off the software interrupt request. FORCE_INTERRUPT = 0x1 force a software interrupt

2	**ENABLE**	RW	0x0	Enable the interrupt bit through the collector.
				DISABLE = 0x0 Disable
				ENABLE = 0x1 Enable

1:0	**PRIORITY**	RW	0x0	Set the priority level for this interrupt, 0x3 is highest, 0x0 is lowest (weakest). LEVEL0 = 0x0 level 0, lowest or weakest priority LEVEL1 = 0x1 level 1 LEVEL2 = 0x2 level 2 LEVEL3 = 0x3 level 3, highest or strongest priority

DESCRIPTION:

This register provides a mechanism to specify the priority associated with an interrupt bit. In addition, this register controls the enable and software generated interrupt. WARNING: Modifying the priority of an enabled interrupt may result in undefined behavior. You should always disable an interrupt prior to changing its priority.

EXAMPLE:

```
HW_ICOLL_INTERRUPT4_SET(0,0x00000001);
```

5.4.15 Interrupt Collector Interrupt Register 5 Description

This register provides a mechanism to specify the priority level for an interrupt source. It also provides an enable and software interrupt for each one, as well as security designation.

HW_ICOLL_INTERRUPT5	0x170
HW_ICOLL_INTERRUPT5_SET	0x174
HW_ICOLL_INTERRUPT5_CLR	0x178
HW_ICOLL_INTERRUPT5_TOG	0x17C

Table 5-30. HW_ICOLL_INTERRUPT5

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSRVD1																											ENFIQ	SOFTIRQ	ENABLE	PRIORITY	

Table 8-58. HW_USBCTRL_ULPI Bit Field Descriptions

BITS	LABEL	RW	RESET	DEFINITION
15:8	**ULPIDATRD**	RO	0x0	Not used. Read as 0.
7:0	**ULPIDATWR**	RW	0x0	Not used. Read as 0.

DESCRIPTION:

ULPI control

EXAMPLE:

```
Empty Example.
```

8.6.30 Endpoint NAK Register Description

HW_USBCTRL_ENDPTNAK 0x178

Table 8-59. HW_USBCTRL_ENDPTNAK

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD1											EPTN					RSVD0											EPRN				

Table 8-60. HW_USBCTRL_ENDPTNAK Bit Field Descriptions

BITS	LABEL	RW	RESET	DEFINITION
31:21	**RSVD1**	RO	0x0	Reserved.

20:16	**EPTN**	RW	0x0	TX Endpoint NAK.
				Each TX endpoint has one bit in this field. The bit is set when the device sends a NAK handshake on a received IN token for the corresponding endpoint.
				EPTN[4] = Endpoint 4
				EPTN[3] = Endpoint 3
				EPTN[2] = Endpoint 2
				EPTN[1] = Endpoint 1
				EPTN[0] = Endpoint 0

15:5	**RSVD0**	RO	0x0	Reserved.

4:0	**EPRN**	RW	0x0	RX Endpoint NAK.
				Each RX endpoint has 1 bit in this field. The bit is set when the device sends a NAK handshake on a received OUT or PING token for the corresponding endpoint.
				EPRN[4] = Endpoint 4
				EPRN[3] = Endpoint 3
				EPRN[2] = Endpoint 2
				EPRN[1] = Endpoint 1
				EPRN[0] = Endpoint 0

DESCRIPTION:

NAK-sent indicator

EXAMPLE:

```
Empty Example.
```

9.4.4 USB PHY General Control Register Description

The USB PHY General Control Register handles Host controls. This register also includes interrupt enables and connectivity detect enables and results.

HW_USBPHY_CTRL	0x030
HW_USBPHY_CTRL_SET	0x034
HW_USBPHY_CTRL_CLR	0x038
HW_USBPHY_CTRL_TOG	0x03c

Table 9-8. HW_USBPHY_CTRL

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
SFTRST	CLKGATE	UTMI_SUSPENDM	HOST_FORCE_LS_SE0	RSVD3														DATA_ON_LRADC	DEVPLUGIN_IRQ	ENIRQDEVPLUGIN	RESUME_IRQ	ENIRQRESUMEDETECT	RSVD2	ENOTGIDDETECT	RSVD1	DEVPLUGIN_POLARITY	ENDEVPLUGINDETECT	HOSTDISCONDETECT_IRQ	ENIRQHOSTDISCON	ENHOSTDISCONDETECT	RSVD0

Table 9-9. HW_USBPHY_CTRL Bit Field Descriptions

BITS	LABEL	RW	RESET	DEFINITION
31	**SFTRST**	RW	0x1	Writing a 1 to this bit will soft-reset the HW_USBPHY_PWD, HW_USBPHY_TX, HW_USBPHY_RX, and HW_USBPHY_CTRL registers.
30	**CLKGATE**	RW	0x1	Gate UTMI Clocks. Clear to 0 to run clocks. Set to 1 to gate clocks. Set this to save power while the USB is not actively being used. Configuration state is kept while the clock is gated.
29	**UTMI_SUSPENDM**	RO	0x0	Used by the PHY to indicate a powered-down state. If all the power-down bits in the HW_USBPHY_PWD are enabled, UTMI_SUSPENDM will be 0, otherwise 1. UTMI_SUSPENDM is negative logic, as required by the UTMI specification.
28	**HOST_FORCE_LS_SE0**	RW	0x0	Forces the next FS packet that is transmitted to have a EOP with low-speed timing. This bit is used in host mode for the resume sequence. After the packet is transferred, this bit is cleared. The design can use this function to force the LS SE0 or use the HW_USBPHY_CTRL_UTMI_SUSPENDM to trigger this event when leaving suspend. This bit is used in conjunction with HW_USBPHY_DEBUG_HOST_RESUME_DEBUG.
27:14	**RSVD3**	RO	0x0	Reserved.

DESCRIPTION:

This register holds a pointer to the data buffer in system memory. After the command values have been read into the DMA controller and the device controlled by this channel, then the DMA transfer will begin, to or from the buffer pointed to by this register.

EXAMPLE:

```
Empty example.
```

10.5.37 APBH DMA Channel 4 Semaphore Register Description

The APBH DMA Channel 4 semaphore register is used to synchronize between the CPU instruction stream and the DMA chain processing state.

HW_APBH_CH4_SEMA 0x240

Table 10-76. HW_APBH_CH4_SEMA

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD2								PHORE								RSVD1								INCREMENT_SEMA							

Table 10-77. HW_APBH_CH4_SEMA Bit Field Descriptions

BITS	LABEL	RW	RESET	DEFINITION
31:24	**RSVD2**	RO	0x0	Reserved, always set to zero.
23:16	**PHORE**	RO	0x0	This read-only field shows the current (instantaneous) value of the semaphore counter.
15:8	**RSVD1**	RO	0x0	Reserved, always set to zero.
7:0	**INCREMENT_SEMA**	RW	0x00	The value written to this field is added to the semaphore count in an atomic way such that simultaneous software adds and DMA hardware substracts happening on the same clock are protected. This bit field reads back a value of 0x00. Writing a value of 0x02 increments the semaphore count by two, unless the DMA channel decrements the count on the same clock, then the count is incremented by a net one.

DESCRIPTION:

Each DMA channel has an 8 bit counting semaphore that is used to synchronize between the program stream and and the DMA chain processing. DMA processing continues until the DMA attempts to decrement a semaphore that has already reached a value of zero. When the attempt is made, the DMA channel is stalled until software increments the semaphore count.

non-zero and the channel is in its IDLE state, then it uses the value in the HW_APBX_CHn_NXTCMDAR (next command address register) to fetch a pointer to the next command to process. NOTE: this is a double indirect case. This method allows software to append to a running command list under the protection of the counting semaphore.

Receiving an IRQ for HALTONTERMINATE (HOT) is a new feature in the APBH/X DMA descriptor that allows certain peripheral block (e.g. GPMI, SSP, I2C) to signal to the DMA engine that an error has occurred. In prior chips, if a block stalled due to an error, the only practical way to discover this in s/w was via a timer of some sort, or to poll the block. Now, an HOT signal is sent from the peripheral to the DMA engine and causes an IRQ after terminating the DMA descriptor being executed. Note not all peripheral block support this termination feature.

Therefore, it is recommended that s/w use this signal as follows:

- Always set HALTONTERMINATE to 1 in a DMA descriptor. That way, if a peripheral signals HOT, the transfer will end, leaving the peripheral block and the DMA engine synchronized (but at the end of a command).
- When an IRQ from an APBH/X channel is received, and the IRQ is determined to be due to an error (as opposed to an IRQONCOMPLETE interrupt) the software should:

1. reset the channel, and
2. determine the error from error reporting in the peripheral block, then manage the error in the peripheral that is attached to that channel in whatever appropriate way exists for that device (software recovery, device reset, block reset, etc).

To start processing the first time, software creates the command list to be processed. It writes the address of the first command into the HW_APBX_CHn_NXTCMDAR register, and then writes a 1 to the counting semaphore in HW_APBX_CHn_SEMA. The DMA channel loads HW_APBX_CHn_CURCMDAR register and then enters the normal state machine processing for the next command. When software writes a value to the counting semaphore, it is added to the semaphore count by hardware, protecting the case where both hardware and software are trying to change the semaphore on the same clock edge.

Software can examine the value of HW_APBX_CHn_CURCMDAR at any time to determine the location of the command structure that is currently being processed.

11.3 DMA Chain Example

The example in Figure 11-3 shows how to bring the basic items together to make a simple DMA chain to read PCM samples and send them out the Audio Output (DAC) using one DMA channel. This example shows three command structures linked together using their normal command list pointers. The first command writes a single PIO word to the HW_AUDIOOUT_CTRL0 register with a new word count for the DAC. This first command also performs a 512 byte DMA_READ operation to read the data block bytes into the DAC. A second and a third DMA command structure also performs a DMA_READ operation to handle circular buffer style outputs. The completion of each command structure generates an interrupt request. In addition, each command structure decrements the semaphore. If the decompression software

DESCRIPTION:

The GPMI payload address register specifies the location of the data buffers in system memory. This value must be word aligned.

EXAMPLE:

```
No Example.
```

13.4.6 GPMI Auxiliary Address Register Description

The GPMI auxiliary address register specifies the location of the auxiliary buffers in system memory. This value must be word aligned.

HW_GPMI_AUXILIARY	0x050

Table 13-12. HW_GPMI_AUXILIARY

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
ADDRESS																														RSVD0	

Table 13-13. HW_GPMI_AUXILIARY Bit Field Descriptions

BITS	LABEL	RW	RESET	DEFINITION
31:2	**ADDRESS**	RW	0x00000000	Pointer to ECC control structure and meta-data storage.
1:0	**RSVD0**	RO	0x0	Always write zeroes to this bit field.

DESCRIPTION:

The GPMI auxiliary address register specifies the location of the auxiliary buffers in system memory. This value must be word aligned.

EXAMPLE:

```
No Example.
```

13.4.7 GPMI Control Register 1 Description

The GPMI control register 1 specifies additional control fields that are not used on a per-transaction basis.

HW_GPMI_CTRL1	0x060
HW_GPMI_CTRL1_SET	0x064
HW_GPMI_CTRL1_CLR	0x068
HW_GPMI_CTRL1_TOG	0x06C

The PXP is organized as having a background image (S0) and one or more overlay images that can be blended with the background. Each overlay image must be a multiple of eight pixels in both height and width and the offset of the overlay into the background image must be a multiple of eight pixels. As the PXP processes data, it reads each 8x8 block from the background image and finds the highest priority (lowest numbered) overlay that is co-located at that block coordinate. The PXP then fetches the overlay and performs the alpha blending and color key operations on the two blocks. The resulting 8x8 pixel block is then written to the corresponding block in the output buffer.

For the S0 plane, the PXP supports RGB images (unscaled) or colorspace conversion (YUV->RGB) and scaling of YUV images. The S1 plane consists of up to eight overlay regions consisting of 16 or 32-bit RGB data. The S0 and S1 planes may then be combined by alpha blending, color key substitution, or raster operations (ROPs) to form the output image. Finally the resulting image may be clock-wise rotated in 90 degree increments or flipped horizontally or vertically. The PXP also supports letterboxing and interlacing of progressive content (by writing alternate lines to different frame buffers).

The flow of data through the PXP is shown in Figure 17-2.

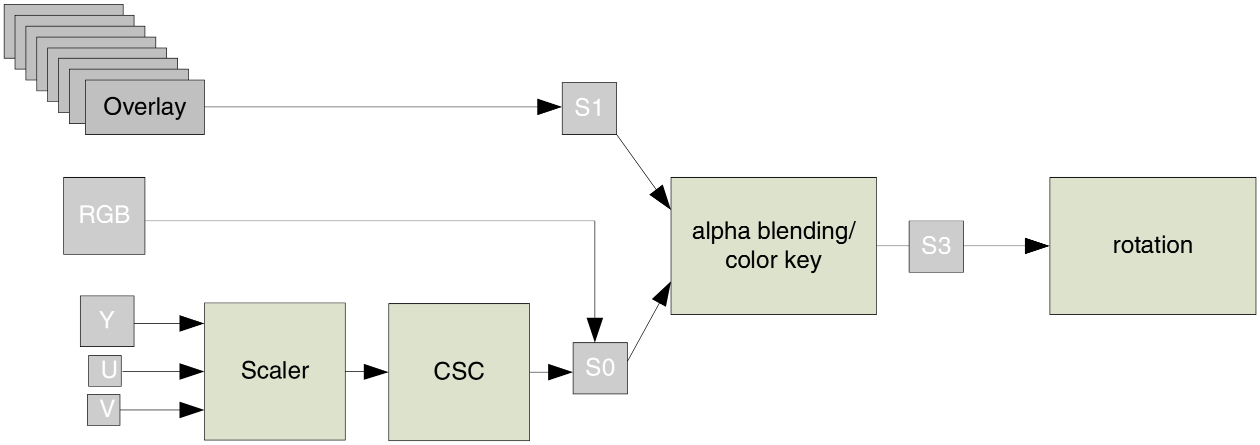


Figure 17-2. Pixel Pipeline (PXP) Data Flow

17.1.1 Image Support

The PXP's S0 buffer supports the following image formats:

- 24-bit unpacked RGB (32bpp)
- 16-bit RGB in either 555 or 565 format
- 3-plane YUV/YCbCr in 4:2:0 or 4:2:2 format

The PXP's S1 buffer supports the following image formats:

- 32-bit RGB (with or without alpha)
- 16-bit RGB in either 555, 565, or 1555 (alpha)

Table 17-94. HW_PXP_OL4PARAM

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD1												ROP				ALPHA								FORMAT				ENABLE_COLORKEY	ALPHA_CNTL		ENABLE

Table 17-95. HW_PXP_OL4PARAM Bit Field Descriptions

BITS	LABEL	RW	RESET	DEFINITION
31:20	**RSVD1**	RO	0x0000	Reserved, always set to zero.

19:16	**ROP**	RW	0x0	Indicates a raster operation to perform when enabled. Raster operations are enabled through the ALPHA_CNTL field.
				MASKOL = 0x0 OL AND S0
				MASKNOTOL = 0x1 nOL AND S0
				MASKOLNOT = 0x2 OL AND nS0
				MERGEOL = 0x3 OL OR S0
				MERGENOTOL = 0x4 nOL OR S0
				MERGEOLNOT = 0x5 OL OR nS0
				NOTCOPYOL = 0x6 nOL
				NOT = 0x7 nS0
				NOTMASKOL = 0x8 OL NAND S0
				NOTMERGEOL = 0x9 OL NOR S0
				XOROL = 0xA OL XOR S0
				NOTXOROL = 0xB OL XNOR S0

15:8	**ALPHA**	RW	0x0	Alpha modifier used when the ALPHA_MULTIPLY or ALPHA_OVERRIDE bits are set. The output alpha value will either be replaced (ALPHA_OVERRIDE) or scaled (ALPHA_MULTIPLY) when enabled in the ALPHA_CNTL field.
7:4	**FORMAT**	RW	0x0	Indicates the input buffer format for overlay 0. ARGB8888 = 0x0 32-bit pixels with alpha RGB888 = 0x1 32-bit pixels without alpha (unpacked 24-bit format) ARGB1555 = 0x3 16-bit pixels with alpha RGB565 = 0x4 16-bit pixels without alpha RGB555 = 0x5 16-bit pixels without alpha
3	**ENABLE_COLORKEY**	RW	0x0	Indicates that colorkey functionality is enabled for this overlay. Pixels found in the overlay colorkey range will be displayed as transparent (the S0 pixel will be used).
2:1	**ALPHA_CNTL**	RW	0x0	Determines how the alpha value is constructed for this overlay. Indicates that the value in the ALPHA field should be used instead of the alpha values present in the input pixels. Embedded = 0x0 Indicates that the OL pixel alpha value will be used to blend the OL with S0. The ALPHA field is ignored. Override = 0x1 Indicates that the value in the ALPHA field should be used instead of the alpha values present in the input pixels. Multiply = 0x2 Indicates that the value in the ALPHA field should be used to scale all pixel alpha values. Each pixel alpha is multiplied by the value in the ALPHA field. ROPs = 0x3 Enable ROPs. The ROP field indicates an operation to be performed on the overlay and S0 pixels.
0	**ENABLE**	RW	0x0	Indicates that the overlay is active for this operation.

- Section 18.2.7, “DOTCLK Interface,”"DOTCLK Interface"
- Section 18.2.8, “ITU-R BT.656 Digital Video Interface (DVI),” "ITU-R BT.656 Digital Video Interface (DVI)"

LCDIF pin usage by interface mode is described in Section 18.2.9, “LCDIF Pin Usage by Interface Mode.”

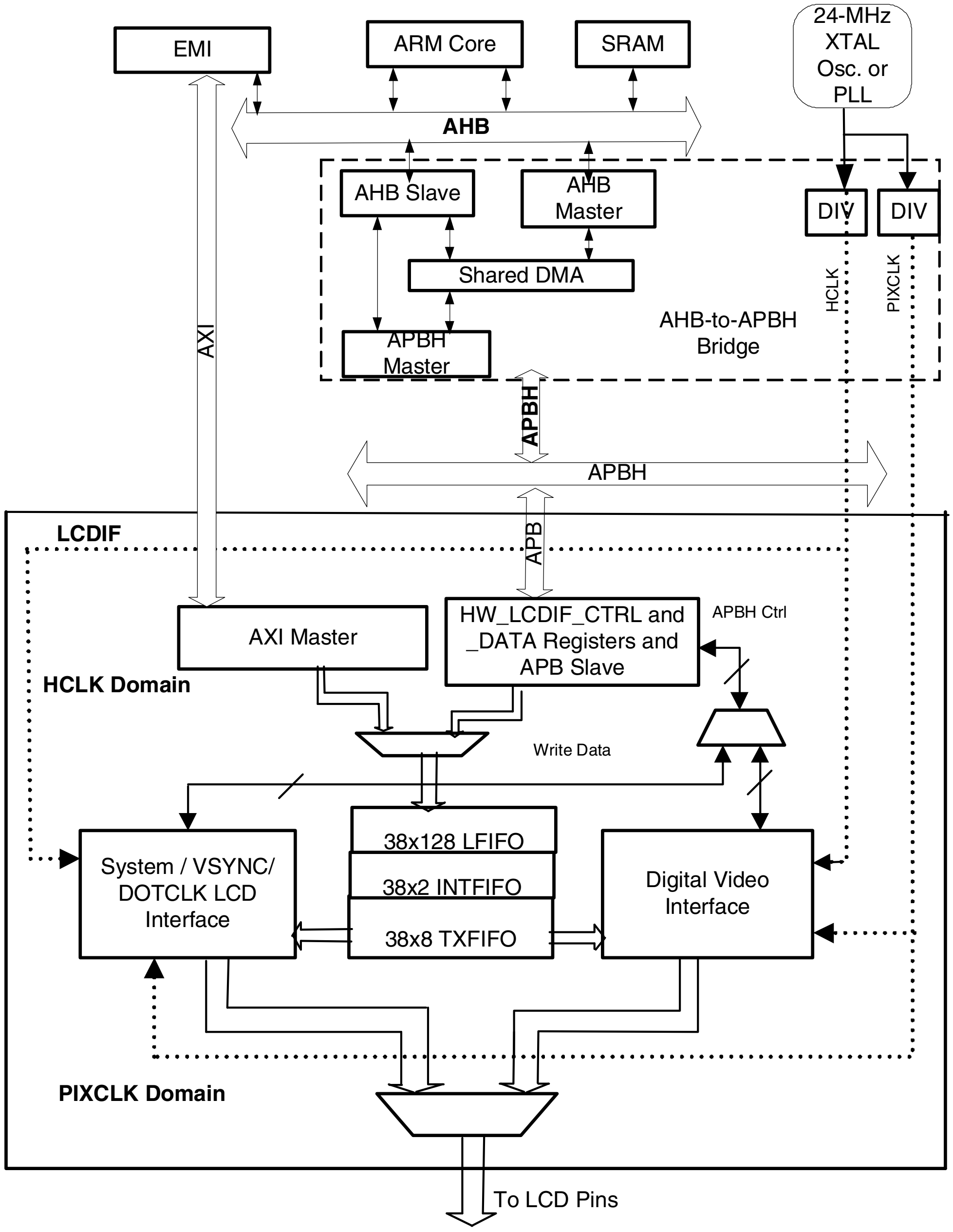


Figure 18-1. LCDIF Top Level Diagram

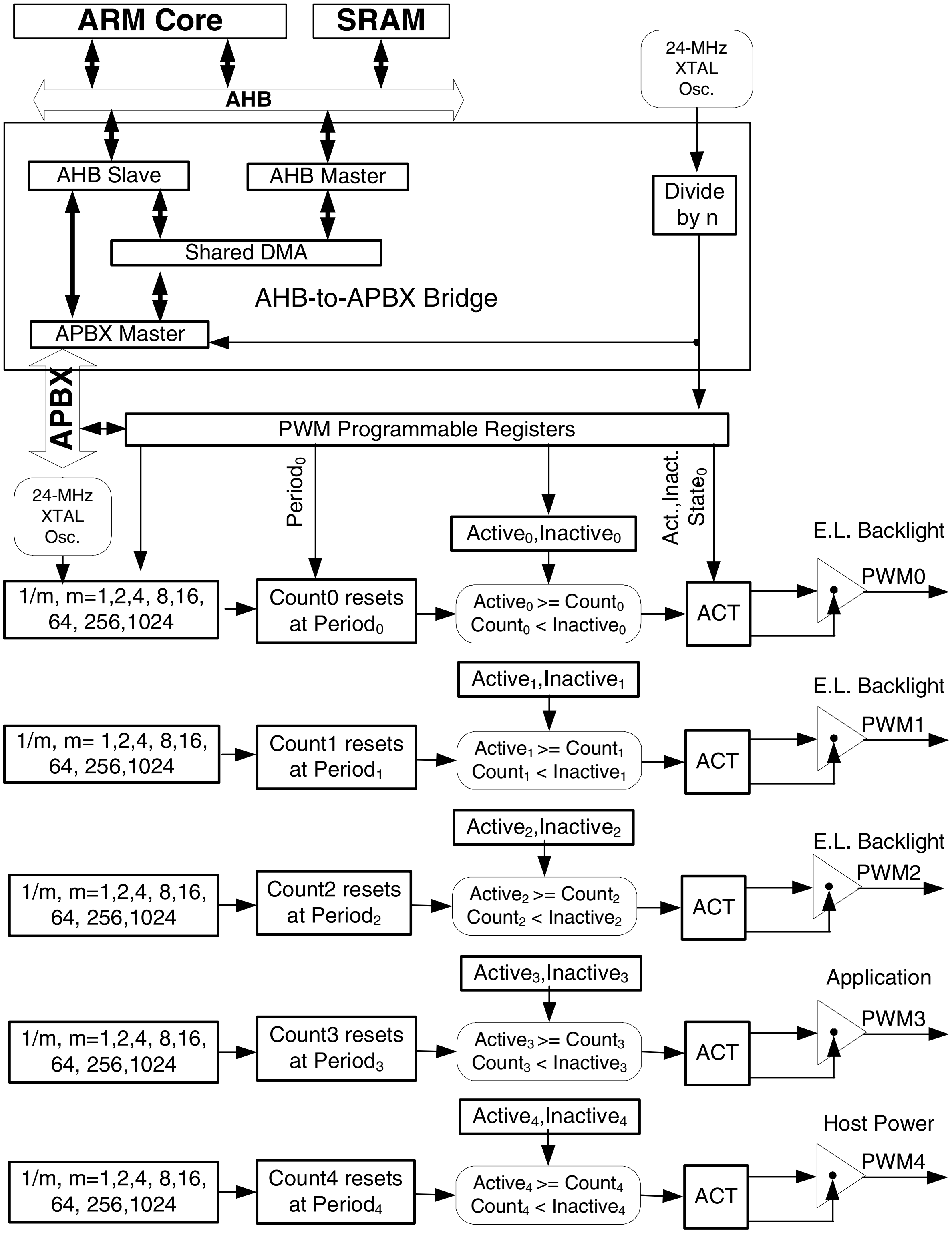


Figure 24-1. Pulse-Width Modulation Controller (PWM) Block Diagram

EXAMPLE:

```
No Example.
```

UARTAPP Block v3.0, Revision 1.42

The SPDIF module receives data by one of two methods:

- Software-directed PIO writes to the HW_SPDIF_DATA register
- Appropriate programming of the DMA-engine. (See Chapter 11, “AHB-to-APBX Bridge with DMA,” for a detailed description of the DMA module and how to perform DMA data transfers to/from modules and memory.)

Once provided by the DMA, the received data is placed in a 2x24 word FIFO for each channel, left and right. At initialization, the FIFO is filled before SPDIF data transfer occurs. After this, data is requested whenever this FIFO has an empty entry or at a nominal rate corresponding to the programmed sample-rate in HW_SPDIF_SRR.

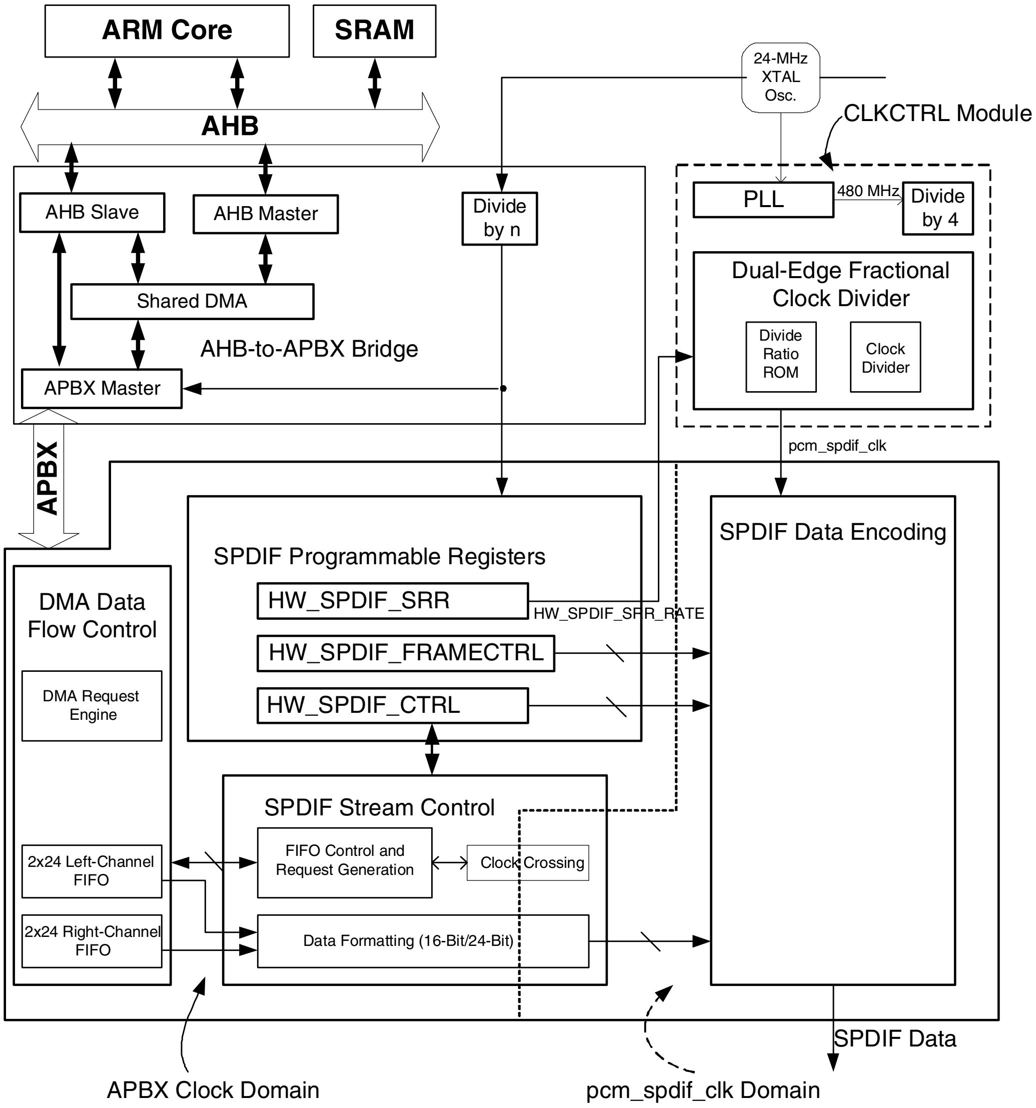


Figure 30-1. SPDIF Transmitter Block Diagram

NOTE: If the data supplied actually represents a lower resolution analog-to-digital conversion, this information is not captured by the SPDIF transmitter, which always reports a 24-bit sample-size.

30.2.1 Interrupts

The SPDIF module contains a single interrupt source that is asserted on FIFO overflows and/or FIFO underflows. This interrupt is enabled by setting HW_SPDIF_CTRL_FIFO_ERROR_IRQ_EN. On interrupt detection, the HW_SPDIF_CTRL_FIFO_UNDERFLOW_IRQ and HW_SPDIF_CTRL_FIFO_OVERFLOW_IRQ fields can be polled for the exact cause of the interrupt and appropriate action taken.

Note: These bits remain valid for polling, regardless of the state of the interrupt enable.

30.2.2 Clocking

The IEC-60958 specification outlines the requirements for SPDIF clocking. The SPDIF module is designed according to the Consumer Audio requirements. These dictate that:

- Average Sample-Rate Error must not exceed 1000 ppm
- Maximum Instantaneous Jitter must not exceed ~4.4 ns.

The jitter requirement implies either a single-phase of a >240-MHz clock or both phases of a 120-MHz clock. It also implies the use of a fractional divider for which the divisors are maintained to sufficient significant digits to yield the required ppm tolerance. The SPDIF module in the i.MX23 uses nine-bit fractional coefficients that yield an average frequency error of 52 ppm. These coefficients are determined according to the required clock-rates that are dictated by the sample rates implemented. The required clock frequencies provided by the CLKCTRL module for the implemented sample-rates are:

F(48 kHz)	≥ 6.144 MHz
F(44.1 kHz)	≥ 5.6448 MHz
F(32 kHz)	≥ 4.096 MHz
F(96 kHz)	≥ 12.288 MHz
F(88.2 kHz)	≥ 11.2896 MHz
F(64 kHz)	≥ 8.192 MHz

All clocks within the SPDIF module are gated according to the state of HW_SPDIF_CTRL_CLKGATE. When set, all clocks derived from the apb_clk are gated. Gating of the pcm_spdif_clk is accomplished through HW_CLKCTRL_SPDIF_CLKGATE. A module-level reset is also provided in HW_SPDIF_CTRL_SFTRST. Setting this bit performs a module-wide reset and subsequent assertion of the HW_SPDIF_CTRL_CLKGATE.

Table 31-4. HW_SAIF_CTRL Bit Field Descriptions

BITS	LABEL	RW	RESET	DEFINITION
10	**JUSTIFY**	RW	0x0	SAIF Data Justification. This bit selects whether serial PCM data is left- or right-justified within each sample's LRCLK frame. 0 = Data is left-justified (start or MSB of serial sample transmission/reception coincides with LRCLK transition) 1 = Data is right-justified (end or LSB of serial sample transmission/reception coincides with LRCLK transition).
9	**LRCLK_POLARITY**	RW	0x0	SAIF LRCLK Polarity Select. This bit selects which LRCLK levels (high/low) correspond to left and right PCM samples. 0 = Left low/right high 1 = Left high/right low.
8	**BITCLK_EDGE**	RW	0x0	SAIF BITCLK Edge Select. For both transmit and receive, this bit selects the BITCLK edge upon which serial PCM data changes. For receive, data is sampled and stored to the receive FIFO on the opposite edge as selected by BITCLK_EDGE that corresponds to the midpoint of the data. 0 = TX: data is driven (changes) on falling-edges of BITCLK; RX: data is sampled on rising-edges of BITCLK 1 = TX: data is driven (changes) on rising-edges of BITCLK; RX: data is sampled on falling-edges of BITCLK.
7:4	**WORD_LENGTH**	RW	0x0	SAIF data size. Selects one of nine PCM data widths from 16-bit to 24-bit to serially input or output from/to a codec. 17-bit to 24-bit PCM data should be right-justified (LSB in bit 0) when it is DMAed or written to the HW_SAIF_DATA register. These samples should be interleaved starting with a left sample first, followed by a right, then left and so on. For 16-bit PCM data, stereo pairs should be constructed with the right sample in the upper half-word (bits 31-16) and the left sample in the lower half word (bits 15-0). 0000 = 16-bit 0001 = 17-bit 0010 = 18-bit 0011 = 19-bit 0100 = 20-bit 0101 = 21-bit 0110 = 22-bit 0111 = 23-bit 1000 = 24-bit 1001-1111 = Reserved but defaults to 24-bit.
3	**BITCLK_48XFS_ENABLE**	RW	0x0	BITCLK 48x Sample Rate Enable. For 384x base frequency multiples, this bit enables generation of 48 BITCLKs per sample pair (24 BITCLKs per channel or LRCLK transition) when the SAIF is BITCLK master. This bit is ignored for the following cases: BITCLK_BASE_RATE=0, or READ_MODE=1, or READ_MODE=0 and SLAVE_MODE=1.

Table 32-5. HW_POWER_5VCTRL Bit Field Descriptions

BITS	LABEL	RW	RESET	DEFINITION
1	**PWRUP_VBUS_CMPS**	RW	0x0	Powers up comparators for 5v
0	**ENABLE_DCDC**	RW	0x0	Enables the switching DC-DC converter when 5V is present. It is recommended to set ILIMIT_EQ_ZERO, ENABLE_ILIMIT, and all LINREG_OFFSET bits when enabling this functionality.

DESCRIPTION:

Empty Description.

EXAMPLE:

```
Empty Example.
```

32.11.3 DC-DC Minimum Power and Miscellaneous Control Register Description

This register controls options to drop the power used by the switching DC-DC converter. These bits should only be modified with guidance from Freescale.

HW_POWER_MINPWR	0x020
HW_POWER_MINPWR_SET	0x024
HW_POWER_MINPWR_CLR	0x028
HW_POWER_MINPWR_TOG	0x02C

Table 32-6. HW_POWER_MINPWR

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSRVD1																	LOWPWR_4P2	VDAC_DUMP_CTRL	PWD_BO	USE_VDDXTAL_VBG	PWD_ANA_CMPS	ENABLE_OSC	SELECT_OSC	VBG_OFF	DOUBLE_FETS	HALF_FETS	LESSANA_I	PWD_XTAL24	DC_STOPCLK	EN_DC_PFM	DC_HALFCLK

Table 32-7. HW_POWER_MINPWR Bit Field Descriptions

BITS	LABEL	RW	RESET	DEFINITION
31:15	**RSRVD1**	RO	0x0	Empty Description.
14	**LOWPWR_4P2**	RW	0x0	Enable low power regulation of DCDC_4P2 limited to 2.5mA. This bit should always be set to 0.
13	**VDAC_DUMP_CTRL**	RW	0x0	Dumps extra Video DAC current into the VDDD supply rail only when the VDDD supply voltage is below it's target value and XX bit is set. Default is to always dump the current to VDDD when XX bit is set.
12	**PWD_BO**	RW	0x0	Powers down supply brownout comparators. This should only be used when monitoring supply brownouts is not needed.

Chapter 37
Pin Control and GPIO

This chapter describes the pin control and general-purpose input/output (GPIO) pin interface implemented on the i.MX23. It includes sections on pin multiplexing and configuration, including color-coded pin multiplexing tables (see Tables 37-1–37-3), followed by a description of the GPIO interface operation. Programmable registers are described in Section 37.4.

37.1 Overview

The pin control interface on the i.MX23 has the following features:

- The i.MX23 has four banks of pins, three of which can serve as GPIOs. The last bank contains the EMI high speed pins which are not muxed with other functions due to tight timing requirements to memory and the need to skew match the timing of the pins.
- All digital pins have selectable output drive strengths as described in Section 37.2.2.1.
- All EMI pins have 1.8/2.5-V and 3.3-V selects as described in Section 37.2.2.1.1.
- All digital pins have weak internal keepers to minimize power loss due to undriven pins.
- The following pin interfaces have selectable pull up resistors:
  - — SSP data - 47 kΩ
  - — SSP command - 10 kΩ
  - — GPMI chip enable - 47 kΩ
  - — GPMI ready/busy - 10 kΩ
- All EMI data and DQM pins' internal keepers can be disabled to allow them to change to a high-impedance state (as required by some DRAM manufacturers).
- Slow transitioning pin interfaces contain internal Schmidt triggers for noise immunity.

NOTE

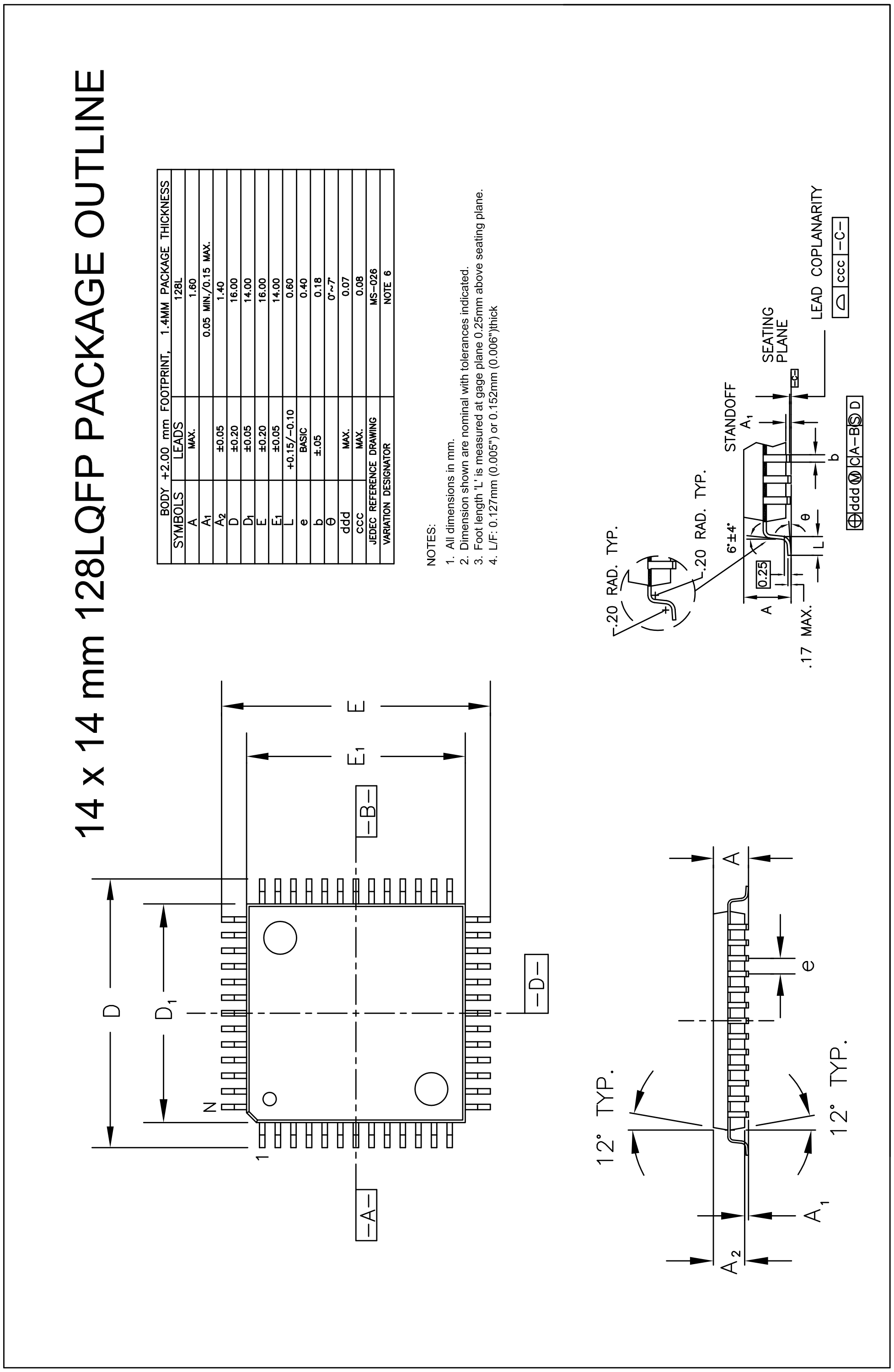
> In the context of this chapter, "digital pin" means the standard digital interface pins. This does *not* include the DEBUG pin.

37.2 Operation

Each individual digital pin supporting GPIO operation may be dynamically programmed at any time to be in one of the following states:

- High-impedance (for input, three-state, or open-drain applications)
- Low
- High

42.2 128-Pin Low-Profile Quad Flat Package (LQFP)



14 x 14 mm 128LQFP PACKAGE OUTLINE

BODY +2.00 mm FOOTPRINT, 1.4MM PACKAGE THICKNESS		
SYMBOLS	LEADS	128L
A	MAX.	1.60
A_1		0.05 MIN./0.15 MAX.
A_2	±0.05	1.40
D	±0.20	16.00
D_1	±0.05	14.00
E	±0.20	16.00
E_1	±0.05	14.00
L	+0.15/−0.10	0.60
e	BASIC	0.40
b	±.05	0.18
θ		0°~7°
ddd	MAX.	0.07
ccc	MAX.	0.08
JEDEC REFERENCE DRAWING		MS–026
VARIATION DESIGNATOR		NOTE 6

NOTES:

1. All dimensions in mm.
2. Dimension shown are nominal with tolerances indicated.
3. Foot length 'L' is measured at gage plane 0.25mm above seating plane.
4. L/F: 0.127mm (0.005") or 0.152mm (0.006")thick

Figure 42-2. 128-Pin Low-Profile Quad Flat Pack (LQFP) Package Drawing