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	-40°C ~ 85°C (TA)
Security Features	Cryptography, Hardware ID
Package / Case	128-LQFP
Supplier Device Package	128-LQFP (14x14)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/mcimx233cag4c

Chapter 2
Characteristics and Specifications

This chapter describes the characteristics and specifications of the i.MX23 and includes sections on absolute maximum ratings, recommended operating conditions, and DC characteristics.

2.1 Absolute Maximum Ratings

Table 2-1. Absolute Maximum Ratings

Parameter	Min	Max	Units
Storage Temperature	–40	125	°C
Battery Pin - BATT, VDD4P2V	–0.3	4.242	V
5-Volt Source Pin - VDD5V (transient, t < 30ms, duty cycle < 0.05%)	-0.3	7.00	V
5-Volt Source Pin - VDD5V (static)	–0.3	6.00	V
PSWITCH (Note 1)	–0.3	VDDXTAL + 1.575	V
Analog Supply Voltage—VDDA	–0.3	2.10	V
Speaker Amplifier Supply Voltage—VDDS	–0.3	4.242	V
Digital Core Supply Voltage —VDDD	–0.3	1.575	V
Non-EMI Digital I/O Supply—VDDIO	–0.3	3.63	V
EMI Digital I/O Supply—VDDIO.EMI	–0.3	3.63	V
DC-DC Converter—DCDC_BATT (Note 2)	–0.3	BATT	V
Input Voltage on Any Digital I/O Pin Relative to Ground	–0.3	VDDIO+0.3	V
Input Voltage on USB_DP and USB_DN Pins Relative to Ground (Note 3)	–0.3	3.63	V
Input Voltage on Any Analog I/O Pin Relative to Ground	–0.3	VDDA+0.3	V

1 VDDIO can be applied to PSWITCH through a 10 kΩ resistor. This is necessary in order to enter the chip's firmware recovery mode. (The on-chip circuitry prevents the actual voltage on the pin from exceeding acceptable levels.)

2 Application should include a Schottky diode between BATT and VDD4P2.

3 USB_DN and USB_DP can tolerate 5V for up to 24 hours. Note that while 5V is applied to USB_DN or USB_DP, LRADC readings can be corrupted.

Table 2-2. Electro-Static Discharge Immunity

169-Pin BGA & 128-Pin LQFP Packages	Tested Level
Human Body Model (HBM)	2 kV
Charge Device Model (CDM)	500 V

Table 8-45. HW_USBCTRL_ASYNCLISTADDR

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
ASYBASE																												RSVD			

Table 8-46. HW_USBCTRL_ASYNCLISTADDR Bit Field Descriptions

BITS	LABEL	RW	RESET	DEFINITION
31:5	**ASYBASE**	RW	0x0	Link Pointer Low (LPL). These bits correspond to memory address signals [31:5], respectively. This field may only reference a Queue Head (OH). Only used by the host controller.
4:0	**RSVD**	RO	0x0	Reserved. These bits are reserved and their value has no effect on operation.

DESCRIPTION:

ASYNC-LIST-ADDR (host-controller)

EXAMPLE:

```
Empty Example.
```

8.6.24 Endpoint List Address Register (Device Controller mode) Description

In Device Controller mode, this register contains the address of the top of the endpoint list in system memory. Bits [10:0] of this register cannot be modified by the system software and will always return a 0 when read. The memory structure referenced by this physical memory pointer is assumed 64-byte. This is a read/write register. Writes must be DWORD writes.

HW_USBCTRL_ENDPOINTLISTADDR 0x158

Table 8-47. HW_USBCTRL_ENDPOINTLISTADDR

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
EPBASE																					RSVD										

11.5 Programmable Registers

This section describes the programmable registers of the AHB-to-APBX bridge block.

11.5.1 AHB to APBX Bridge Control Register 0 Description

The APBX CTRL 0 provides overall control and IRQ status of the AHB to APBX bridge and DMA.

HW_APBX_CTRL0	0x000
HW_APBX_CTRL0_SET	0x004
HW_APBX_CTRL0_CLR	0x008
HW_APBX_CTRL0_TOG	0x00C

Table 11-4. HW_APBX_CTRL0

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	RSVD0																													

Table 11-5. HW_APBX_CTRL0 Bit Field Descriptions

BITS	LABEL	RW	RESET	DEFINITION
31	**SFTRST**	RW	0x1	Set this bit to zero to enable normal APBX DMA operation. Set this bit to one (default) to disable clocking with the APBX DMA and hold it in its reset (lowest power) state. This bit can be turned on and then off to reset the APBX DMA block to its default state.
30	**CLKGATE**	RW	0x1	This bit must be set to zero for normal operation. When set to one it gates off the clocks to the block.
29:0	**RSVD0**	RO	0x000000	Reserved, always set to zero.

DESCRIPTION:

This register contains softreset, clock gating bits.

EXAMPLE:

```
No Example.
```

11.5.2 AHB to APBX Bridge Control Register 1 Description

The APBX CTRL 1 provides channel complete IRQ status of the AHB to APBX bridge and DMA.

HW_APBX_CTRL1	0x010
HW_APBX_CTRL1_SET	0x014
HW_APBX_CTRL1_CLR	0x018
HW_APBX_CTRL1_TOG	0x01C

Table 11-166. HW_APBX_CH10_DEBUG2

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
APB_BYTES																AHB_BYTES															

Table 11-167. HW_APBX_CH10_DEBUG2 Bit Field Descriptions

BITS	LABEL	RW	RESET	DEFINITION
31:16	**APB_BYTES**	RO	0x0	This value reflects the current number of APB bytes remaining to be transfered in the current transfer.
15:0	**AHB_BYTES**	RO	0x0	This value reflects the current number of AHB bytes remaining to be transfered in the current transfer.

DESCRIPTION:

This register allows debug visibility of the APBX DMA Channel 10.

EXAMPLE:

```
Empty example.
```

11.5.83 APBX DMA Channel 11 Current Command Address Register Description

The APBX DMA Channel 11 current command address register points to the multiword command that is currently being executed. Commands are threaded on the command address.

HW_APBX_CH11_CURCMDAR 0x5D0

Table 11-168. HW_APBX_CH11_CURCMDAR

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
CMD_ADDR																															

Table 11-169. HW_APBX_CH11_CURCMDAR Bit Field Descriptions

BITS	LABEL	RW	RESET	DEFINITION
31:0	**CMD_ADDR**	RO	0x00000000	Pointer to command structure currently being processed for Channel 11.

DESCRIPTION:

APBX DMA Channel 11 is controlled by a variable sized command structure. This register points to the command structure currently being executed.

Chapter 12
External Memory Interface (EMI)

This chapter describes the external memory interface (EMI) on the i.MX23. It describes the DRAM controller and EMI power management. Programmable registers for both the DRAM controller are described in Section 12.5, “Programmable Registers.”

12.1 Overview

The i.MX23 supports off-chip DRAM storage via the EMI controller, which is connected to the four internal AHB/AXI busses.

The EMI supports multiple external memory types, including:

- 1.8-V Mobile DDR
- Standard 2.5V DDR1

The DRAM controller supports up to two external chip-select signals for the i.MX23 platform. Programmable registers within the DRAM controller allow great flexibility for device timings, low-power operation, and performance tuning. Note the differences between the two package options:

- The 128-pin LQFP has 1 chip enable. Maximum DRAM supported is 64MB.
- The 169-pin BGA has 2 chip enables. Maximum DRAM supported is 128MB.

The EMI uses two primary clocks: the AHB bus HCLK and the DRAM source clock EMI_CLK. The maximum specified frequencies for these two clocks can be found in Chapter 2, “Characteristics and Specifications. The memory controller operates at frequencies that are asynchronous to the rest of the i.MX23.

The EMI consists of two major components:

- DRAM controller
- Delay compensation circuitry (DCC)

Table 12-24. HW_DRAM_CTL08 Bit Field Descriptions

BITS	LABEL	RW	RESET	DEFINITION
31:25	**RSVD4**	RO	0x0	Reserved.
24	**TRAS_LOCKOUT**	RW	0x0	Allow the controller to execute auto pre-charge commands before TRAS_MIN expires. Defines the tRAS lockout setting for the DRAM device. tRAS lockout allows the memory controller to execute auto pre-charge commands before the TRAS_MIN parameter has expired. 0 = tRAS lockout not supported by memory device. 1 = tRAS lockout supported by memory device.
23:17	**RSVD3**	RO	0x0	Reserved.
16	**START**	RW	0x0	Initiate command processing in the controller. With this bit field cleared to 0, the memory controller will not issue any commands to the DRAM devices or respond to any signal activity except for reading and writing bit fields. Once this bit field is set to 1, the memory controller will respond to inputs from the ASIC. When set, the memory controller begins its initialization routine. When the interrupt bit in the INT_STATUS bit field associated with completed initialization is set, the user may begin to submit transactions. 0 = Controller is not in active mode. 1 = Initiate active mode for the memory controller.
15:9	**RSVD2**	RO	0x0	Reserved.
8	**SREFRESH**	RW	0x0	Place DRAMs in self-refresh mode. When this bit field is written with a 1, the DRAM device(s) will be placed in self-refresh mode. For this, the current burst for the current transaction (if any) will complete, all banks will be closed, the self-refresh command will be issued to the DRAM, and the clock enable signal will be de-asserted. The system will remain in self-refresh mode until this bit field is written with a 0. The DRAM devices will return to normal operating mode after the self-refresh exit time (txsr) of the device and any DLL initialization time for the DRAM is reached. The memory controller will resume processing of the commands from the interruption point. This bit field will be updated with an assertion of the srefresh_enter pin, regardless of the behavior on the register interface. To disable self-refresh again after a srefresh_enter pin assertion, the user will need to clear the bit field to 0. 0 = Disable self-refresh mode. 1 = Initiate self-refresh of the DRAM devices.
7:1	**RSVD1**	RO	0x0	Reserved.
0	**SDR_MODE**	RW	0x0	Select SDR or DDR mode of the controller. Selects between SDR (single data rate) and DDR (dual data rate) modes. 0 = DDR mode 1 = SDR mode

DESCRIPTION:

DRAM Control registers. Individual fields control various aspects of the DRAM interface. See bit fields for descriptions

Table 12-43. HW_DRAM_CTL18

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
RSVD4	DLL_DQS_DELAY_1							RSVD3	DLL_DQS_DELAY_0							RSVD2			INT_STATUS					RSVD1			INT_MASK				

Table 12-44. HW_DRAM_CTL18 Bit Field Descriptions

BITS	LABEL	RW	RESET	DEFINITION
31	**RSVD4**	RO	0x0	Reserved.
30:24	**DLL_DQS_DELAY_1**	RW	0x00	Fraction of a cycle to delay the dqs signal from the DRAMs for dll_rd_dqs_slice 1 during reads. Sets the delay for the read_dqs signal from the DDR SDRAM devices for dll_rd_dqs_slice 1. This delay is used center the edges of the read_dqs signal so that the read data will be captured in the middle of the valid window in the I/O logic.Each increment of this bit field adds a delay of 1/128 of the system clock.
23	**RSVD3**	RO	0x0	Reserved.
22:16	**DLL_DQS_DELAY_0**	RW	0x00	Fraction of a cycle to delay the dqs signal from the DRAMs for dll_rd_dqs_slice 0 during reads. Sets the delay for the read_dqs signal from the DDR SDRAM devices for dll_rd_dqs_slice 0. This delay is used center the edges of the read_dqs signal so that the read data will be captured in the middle of the valid window in the I/O logic.Each increment of this bit field adds a delay of 1/128 of the system clock.
15:13	**RSVD2**	RO	0x0	Reserved.
12:8	**INT_STATUS**	RO	0x00	Status of interrupt features in the controller. Shows the status of all possible interrupts generated by the memory controller. The MSB is the result of a logical OR of all the lower bits. The INT_STATUS bits correspond to these interrupts: Bit 0 = A single access outside the defined PHYSICAL memory space detected. Bit 1 = Multiple accesses outside the defined PHYSICAL memory space detected. Bit 2 = DRAM initialization complete. Bit 3 = DLL unlock condition detected. Bit 4 = Logical OR of all lower bits.
7:5	**RSVD1**	RO	0x0	Reserved.
4:0	**INT_MASK**	RW	0x00	Mask for controller_int signals from the INT_STATUS bit field. Active-high mask bits that control the value of the memory controller_int signal on the ASIC interface. This mask is inverted and then logically AND'ed with the outputs of the INT_STATUS bit field.

Table 12-62. HW_DRAM_CTL27 Bit Field Descriptions

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

DESCRIPTION:

DRAM Control registers. Individual fields control various aspects of the DRAM interface. See bit fields for descriptions

EXAMPLE:

```
Empty Example.
```

12.5.31 DRAM Control Register 28 Description

DRAM control register. See bit fields for detailed descriptions.

HW_DRAM_CTL28 0x070

Table 12-63. HW_DRAM_CTL28

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
OBSOLETE																															

Table 12-64. HW_DRAM_CTL28 Bit Field Descriptions

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

DESCRIPTION:

DRAM Control registers. Individual fields control various aspects of the DRAM interface. See bit fields for descriptions

EXAMPLE:

```
Empty Example.
```

12.5.32 DRAM Control Register 29 Description

DRAM control register. See bit fields for detailed descriptions.

HW_DRAM_CTL29 0x074

Table 14-8. HW_ECC8_DEBUG0 Bit Field Descriptions

BITS	LABEL	RW	RESET	DEFINITION
11	**KES_STANDALONE**	RW	0x0	Set to 1 to cause the KES engine to suppress toggling the KES_BM_DONE signal to the bus master and to suppress toggling the CF_BM_DONE signal by the CF engine. NORMAL = 0x0 Bus master address generator for synd_gen writes operates normally. TEST_MODE = 0x1 Bus master address generator always addresses last four bytes in auxiliary block.
10	**KES_DEBUG_STEP**	RW	0x0	Toggling this bit causes the KES FSM to skip past the stall state if it is in DEBUG_STALL mode and it has completed processing a block.
9	**KES_DEBUG_STALL**	RW	0x0	Set to 1 to cause KES FSM to stall after notifying the Chien search engine to start processing its block but before notifying the bus master that the KES computation is complete. This allows a diagnostic to stall the FSM after each block's key equations are solved. This also has the effect of stalling the CSFE search engine so its state can be examined after it finishes processing the KES stalled block. NORMAL = 0x0 KES FSM proceeds to next block supplied by bus master. WAIT = 0x1 KES FSM waits after current equations are solved and the search engine is started.
8	**BM_KES_TEST_BYPASS**	RW	0x0	1 = Point all synd_gen writes to dummy area at the end of the auxiliary block so that diagnostics can preload all payload, parity bytes, and computed syndrome bytes for test the KES engine. NORMAL = 0x0 Bus master address generator for synd_gen writes operates normally. TEST_MODE = 0x1 Bus master address generator always addresses last four bytes in auxiliary block.
7:6	**RSRVD0**	RO	0x0	Reserved.
5:0	**DEBUG_REG_SELECT**	RW	0x0	The value loaded in this bit field is used to select the internal register state view of KES engine or the Chien search engine.

DESCRIPTION:

The Hardware ECC Accelerator Debug Register 0 provides access to various internal state information which might prove useful during hardware debug and validation.

EXAMPLE:

```
Value = HW_ECC8_DEBUG0.U; // diagnostic programs can read and act upon various bit fields.
```

14.4.5 KES Debug Read Register Description

The hardware ECC accelerator key equation solver internal state machines and signals can be seen in the KES Debug Read Register.

HW_ECC8_DBGKESREAD 0x040

The hardware BCH accelerator is illustrated in Figure 15-1.

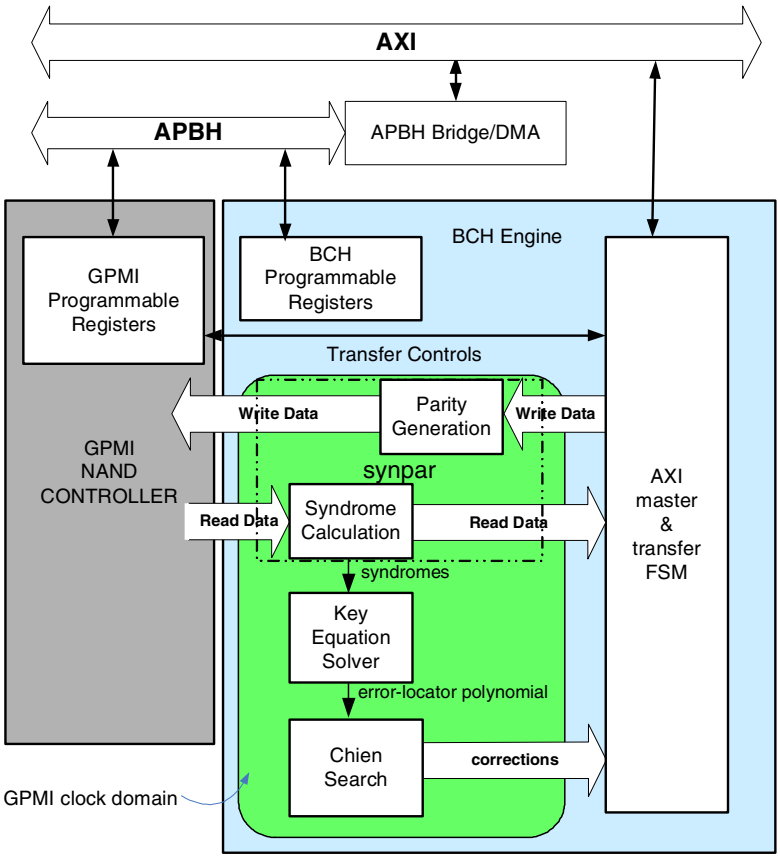

Figure 15-1. Hardware BCH Accelerator

15.2 Operation

Before performing any NAND flash read or write operations, software should first program the BCH's flash layout registers (see Section 15.2.2, "Flash Page Layout") to specify how data is to be formatted on the flash device. The BCH hardware allows full programmability over the flash page layout to enable users flexibility in balancing ECC correction levels and ever-changing flash page sizes.

To initiate a NAND flash write, software will program a GPMI DMA operation. The DMA need only program the GPMI control registers (and handle the requisite flash addressing handshakes) since the BCH will handle all data operations using its AXI bus interface. The BCH will then send the data to the GPMI controller to be written to flash as it computes the parity symbols. At the end of each data block the BCH will insert the parity symbols into the data stream so that the GPMI sees only a continuous stream of data to be written.

NAND flash read operations operate in a similar manner. As the GPMI controller reads the device, all data is sent to the BCH hardware for error detection/correction. The BCH controller writes all incoming read data to system memory and in parallel computes the syndromes used to detect bit errors. If errors are

Table 15-35. HW_BCH_DEBUG0 Bit Field Descriptions

BITS	LABEL	RW	RESET	DEFINITION
9	**KES_DEBUG_STALL**	RW	0x0	Set to one to cause KES FSM to stall after notifying Chien search engine to start processing its block but before notifying the bus master that the KES computation is complete. This allows a diagnostic to stall the FSM after each block's key equations are solved. This also has the effect of stalling the CSFE search engine so it's state can be examined after it finishes processing the KES stalled block. NORMAL = 0x0 KES FSM proceeds to next block supplied by bus master. WAIT = 0x1 KES FSM waits after current equations are solved and the search engine is started.
8	**BM_KES_TEST_BYPASS**	RW	0x0	1 = Point all synd_gen writes to dummy area at the end of the AUXILIARY block. With this data diagnostics can preload all payload, parity bytes and computed syndrome bytes for testing the KES engine. NORMAL = 0x0 Bus master address generator for synd_gen writes operates normally. TEST_MODE = 0x1 Bus master address generator always addresses last four bytes in Auxilliary block.
7:6	**RSVD0**	RO	0x0	Reserved, always set these bits to zero.
5:0	**DEBUG_REG_SELECT**	RW	0x0	The value loaded in this bit field is used to select the internal register state view of KES engine or the Chien search engine.

DESCRIPTION:

The HW_BCH_DEBUG0 register provides access to various internal state information which might prove useful during hardware debug and validation.

EXAMPLE:

```
// perform BIST operation
HW_BCH_DEBUG0_SET(BM_BCH_DEBUG0_ROM_BIST_ENABLE); // enable BIST operation
// poll until BIST_DONE
while( (HW_BCH_DEBUG0_RD() & BM_BCH_DEBUG0_ROM_BIST_COMPLETE) == 0 );
i=HW_BCH_DBGKESREAD();
if(HW_BCH_DBGKESREAD_RD() != 0x7AA3792F) {
// BIST FAILED
err++;
}
HW_BCH_DEBUG0_CLR(BM_BCH_DEBUG0_ROM_BIST_ENABLE | BM_BCH_DEBUG0_ROM_BIST_COMPLETE); // clear bist
status
```

15.6.17 Hardware BCH ECC KES Debug Read Register Description

The hardware ECC 8 accelerator key equation solver internal state machines and signals can be seen in the ECC debug registers.

HW_BCH_DBGKESREAD 0x110

Table 15-36. HW_BCH_DBGKESREAD

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
VALUES																															

Table 18-4. HW_LCDIF_CTRL Bit Field Descriptions

BITS	LABEL	RW	RESET	DEFINITION
15:14	**INPUT_DATA_SWIZZLE**	RW	0x0	This field specifies how to swap the bytes either in the HW_LCDIF_DATA register or those fetched by the AXI master part of LCDIF.The swizzle function is independent of the WORD_LENGTH bit. See the explanation of the HW_LCDIF_DATA below for names and definitions of data register fields. The supported swizzle configurations are: NO_SWAP = 0x0 No byte swapping.(Little endian) LITTLE_ENDIAN = 0x0 Little Endian byte ordering (same as NO_SWAP). BIG_ENDIAN_SWAP = 0x1 Big Endian swap (swap bytes 0,3 and 1,2). SWAP_ALL_BYTES = 0x1 Swizzle all bytes, swap bytes 0,3 and 1,2 (aka Big Endian). HWD_SWAP = 0x2 Swap half-words. HWD_BYTE_SWAP = 0x3 Swap bytes within each half-word.
13:12	**CSC_DATA_SWIZZLE**	RW	0x0	This field specifies how to swap the bytes after the data has been converted into an internal representation of 24 bits per pixel and before it is transmitted over the LCD interface bus. The data is always transmitted with the least significant byte/hword (half word) first after the swizzle takes place. So, INPUT_DATA_SWIZZLE takes place first on the incoming data, and then CSC_DATA_SWIZZLE is applied. The swizzle function is independent of the WORD_LENGTH or the LCD_DATABUS_WIDTH fields. If RGB_TO_YCRCB422_CSC bit is set, the swizzle occurs on the Y, Cb, Cr values. The supported swizzle configurations are: NO_SWAP = 0x0 No byte swapping.(Little endian) LITTLE_ENDIAN = 0x0 Little Endian byte ordering (same as NO_SWAP). BIG_ENDIAN_SWAP = 0x1 Big Endian swap (swap bytes 0,3 and 1,2). SWAP_ALL_BYTES = 0x1 Swizzle all bytes, swap bytes 0,3 and 1,2 (aka Big Endian). HWD_SWAP = 0x2 Swap half-words. HWD_BYTE_SWAP = 0x3 Swap bytes within each half-word.
11:10	**LCD_DATABUS_WIDTH**	RW	0x0	LCD Data bus transfer width. 16_BIT = 0x0 16-bit data bus mode. 8_BIT = 0x1 8-bit data bus mode. 18_BIT = 0x2 18-bit data bus mode. 24_BIT = 0x3 24-bit data bus mode.
9:8	**WORD_LENGTH**	RW	0x0	Input data format. 16_BIT = 0x0 Input data is 16 bits per pixel. Valid BYTE_PACKING_FORMAT settings are 0x3, 0xC and 0xF. H_COUNT must be a multiple of 2 pixels if BYTE_PACKING_FORMAT = 0xF. 8_BIT = 0x1 Input data is 8 bits wide. Any setting in BYTE_PACKING_FORMAT is valid as long as it is non-zero. H_COUNT must be a multiple of sum of the bits of BYTE_PACKING_FORMAT[3:0]. 18_BIT = 0x2 Input data is 18 bits per pixel. Valid BYTE_PACKING_FORMAT setting is 0xF. There are no restrictions on H_COUNT. 24_BIT = 0x3 Input data is 24 bits per pixel. Valid BYTE_PACKING_FORMAT settings are 0x7, 0xE and 0xF. If BYTE_PACKING_FORMAT = 0xF, H_COUNT must be a multiple of 4 pixels, otherwise there are no restrictions.
7	**RGB_TO_YCBCR422_CSC**	RW	0x0	Set this bit to 1 to enable conversion from RGB to YCbCr colorspace. See the HW_LCDIF_CSC_ registers for further details.
6	**RSVD2**	RW	0x0	Program this field to 0x0.

18.4.12 Digital Video Interface Control0 Register Description

The Digital Video interface Control0 register provides the overall control of the Digital Video interface.

HW_LCDIF_DVICTRL0 0x0c0

Table 18-25. HW_LCDIF_DVICTRL0

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	H_ACTIVE_CNT											H_BLANKING_CNT										V_LINES_CNT									

Table 18-26. HW_LCDIF_DVICTRL0 Bit Field Descriptions

BITS	LABEL	RW	RESET	DEFINITION
31	**RSVD1**	RW	0x0	Program this field to 0x0.
30:20	**H_ACTIVE_CNT**	RW	0x000	Number of active video samples to be transmitted. (Mostly will be 1440 for both PAL and NTSC). Must always be a multiple of 4.
19:10	**H_BLANKING_CNT**	RW	0x000	Number of blanking samples to be inserted between EAV and SAV during horizontal blanking interval.
9:0	**V_LINES_CNT**	RW	0x000	Total number of vertical lines per frame (generally 525 or 625)

DESCRIPTION:

This register gives information about the horizontal active, horizontal blanking and total number of lines in the ITU-R BT.656 interface.

EXAMPLE:

```
//525/60 video system
HW_LCDIF_DVICTRL0_H_ACTIVE_CNT_WR(0x5A0);//1440
HW_LCDIF_DVICTRL0_H_BLANKING_CNT_WR(0x106);//262
HW_LCDIF_DVICTRL0_V_LINES_CNT_WR(0x20D);//525
//625/50 video system
HW_LCDIF_DVICTRL0_H_ACTIVE_CNT_WR(0x5A0);//1440
HW_LCDIF_DVICTRL0_H_BLANKING_CNT_WR(0x112);//274
HW_LCDIF_DVICTRL0_V_LINES_CNT_WR(0x271);//625
```

18.4.13 Digital Video Interface Control1 Register Description

The Digital Video interface Control1 register provides the overall control of the Digital Video interface.

HW_LCDIF_DVICTRL1 0x0d0

Table 23-25. HW_RTC_DEBUG

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
RSVD0																														WATCHDOG_RESET_MASK	WATCHDOG_RESET

Table 23-26. HW_RTC_DEBUG Bit Field Descriptions

BITS	LABEL	RW	RESET	DEFINITION
31:2	**RSVD0**	RO	0x0	Debug read-only view of various state machine bits.
1	**WATCHDOG_RESET_MASK**	RW	0x0	When set, mask the reset generation by the watchdog timer for testing purposes.
0	**WATCHDOG_RESET**	RO	0x0	Reflects the state of the watchdog reset. Used for testing purposes so the watchdog can be tested without resetting part. When set, Watchdog reset is asserted.

DESCRIPTION:

Read-only view into the internals of the digital side of the RTC for diagnostic purposes.

EXAMPLE:

```
DebugValue = HW_RTC_DEBUG_RD(); // read debug register value
```

23.8.14 Real-Time Clock Version Register Description

Version register.

HW_RTC_VERSION 0x0D0

Table 23-27. HW_RTC_VERSION

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
MAJOR								MINOR								STEP															

Table 25-16. HW_I2C_CTRL0 Bit Field Descriptions

BITS	LABEL	RW	RESET	DEFINITION
24	**PIO_MODE**	RW	0x0	Set this bit to one to enable PIO mode of operation for the I2C master. One can preload up to four bytes into HW_I2C_DATA register before setting the RUN bit. The state machine will not attempt to use the DMA for master transmit operation. The normal start and stop conditions can be sent and the clock can be held at the end of the transfer, if desired. NOTE: all receive operations must use the DMA mode, not the PIO mode.
23	**RSVD1**	RW	0x0	Program this field to 0x0.
22	**CLOCK_HELD**	RW	0x0	This bit is set to one by the I2C controller state machines. It holds the I2C clock line low until cleared. It must be cleared by firmware, either by CPU instructions or DMA PIO transactions. It is set high when a slave address is matched by the slave controller. It is also set high at the end of a master or slave transaction that had the RETAIN_CLOCK bit set high. Software should not set this bit to one. RELEASE = 0x0 Release the clock line. HELD_LOW = 0x1 The clock line is currently being held low.
21	**RETAIN_CLOCK**	RW	0x0	Set this bit to one to retain the clock at the end of this transaction. This has the effect of holding the clock low until the start of the next transaction. RELEASE = 0x0 Release the clock line after this data transfer. HOLD_LOW = 0x1 Hold the clock line low after this data transfer.
20	**POST_SEND_STOP**	RW	0x0	Set this bit to one to send a stop condition after transferring the data associated with this transaction. This bit is automatically cleared by the hardware after the operation has been performed. NO_STOP = 0x0 Do not send a stop condition before this transaction. SEND_STOP = 0x1 Send a stop condition before this transaction.
19	**PRE_SEND_START**	RW	0x0	Set this bit to one to send a start condition before transferring the data associated with this transaction. This bit is automatically cleared by the hardware after the operation has been performed. NO_START = 0x0 Do not send a start condition before this transaction. SEND_START = 0x1 Send a start condition before this transaction.
18	**SLAVE_ADDRESS_ENABLE**	RW	0x0	Set this bit to one to enable the slave address decoder. When an address match occurs, the I2C bus clock is frozen, by setting HW_I2C_CTRL0_CLOCK_HELD, and an interrupt is generated. DISABLED = 0x0 Disable the slave address decoder. ENABLED = 0x1 Enable the slave address decoder.
17	**MASTER_MODE**	RW	0x0	Set this bit to one to select master mode. MASTER_DISABLED = 0x0 Master mode disabled. MASTER_ENABLED = 0x1 Operate in master mode.
16	**DIRECTION**	RW	0x0	Set this bit to one to select an I2C transmit operation. XMIT = write. Set this bit to zero to select an I2C receive operation. RECEIVE = 0x0 I2C receive operation. TRANSMIT = 0x1 I2C transmit operation.
15:0	**XFER_COUNT**	RW	0x0000	Number of bytes to transfer. This field decrements as bytes are transferred.

DESCRIPTION:

This register is either written by the DMA or the CPU depending on the state of an I2C transaction.

Table 26-11. HW_UARTAPP_LINECTRL2 Bit Field Descriptions

BITS	LABEL	RW	RESET	DEFINITION
31:16	**BAUD_DIVINT**	RW	0x0	Baud Rate Integer [15:0]. The integer baud rate divisor.
15:14	**RSVD**	RO	0x0	Reserved, do not modify, read as zero.
13:8	**BAUD_DIVFRAC**	RW	0x0	Baud Rate Fraction [5:0]. The fractional baud rate divisor.
7	**SPS**	RW	0x0	Stick Parity Select. When bits 1, 2, and 7 of this register are set, the parity bit is transmitted and checked as a 0. When bits 1 and 7 are set, and bit 2 is 0, the parity bit is transmitted and checked as a 1. When this bit is cleared stick parity is disabled.
6:5	**WLEN**	RW	0x0	Word length [1:0]. The select bits indicate the number of data bits transmitted or received in a frame as follows: 11 = 8 bits, 10 = 7 bits, 01 = 6 bits, 00 = 5 bits.
4	**FEN**	RW	0x0	Enable FIFOs. If this bit is set to 1, transmit and receive FIFO buffers are enabled (FIFO mode). When cleared to 0, the FIFOs are disabled (character mode); that is, the FIFOs become 1-byte-deep holding registers.
3	**STP2**	RW	0x0	Two Stop Bits Select. If this bit is set to 1, two stop bits are transmitted at the end of the frame. The receive logic does not check for two stop bits being received.
2	**EPS**	RW	0x0	Even Parity Select. If this bit is set to 1, even parity generation and checking is performed during transmission and reception, which checks for an even number of 1s in data and parity bits. When cleared to 0, then odd parity is performed which checks for an odd number of 1s. This bit has no effect when parity is disabled by Parity Enable (PEN, bit 1) being cleared to 0.
1	**PEN**	RW	0x0	Parity Enable. If this bit is set to 1, parity checking and generation is enabled, else parity is disabled and no parity bit added to the data frame.
0	**RSVD1**	RO	0x0	Reserved, do not modify, read as zero.

DESCRIPTION:

The UART Line Control 2 Register contains integer and fractional part of the baud rate divisor value. It also contains the line control bits.

EXAMPLE:

```
No Example.
```

26.4.6 UART Interrupt Register Description

The UART Interrupt Register contains the interrupt enables and the interrupt status. The interrupt status bits report the unmasked state of the interrupts. To clear a particular interrupt status bit, write the bit-clear address with the particular bit set to 1. The enable bits control the UART interrupt output: a 1 will enable a particular interrupt to assert the UART interrupt output, while a 0 will disable the particular interrupt from affecting the interrupt output. All the bits, except for the modem status interrupt bits, are cleared to 0 when reset. The modem status interrupt bits are undefined after reset.

HW_UARTAPP_INTR 0x050

Table 30-1. HW_SPDIF_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	RSRVD1									DMAWAIT_COUNT					RSRVD0										WAIT_END_XFER	WORD_LENGTH	FIFO_UNDERFLOW_IRQ	FIFO_OVERFLOW_IRQ	FIFO_ERROR_IRQ_EN	RUN

Table 30-2. HW_SPDIF_CTRL Bit Field Descriptions

BITS	LABEL	RW	RESET	DEFINITION
31	**SFTRST**	RW	0x1	Setting this bit to one forces a reset to the entire block and then gates the clocks off. This bit must be set to zero for normal operation.
30	**CLKGATE**	RW	0x1	This bit must be set to zero for normal operation. When set to one it gates off the clocks to the block. WARNING: First set the CLKGATE bit in the HW_CLKCTRL_SPDIF register to 1. Only then, set this bit to 1 to prevent any extra samples from being transmitted. When removing clock gating, follow the reverse order: First reset this CLKGATE bit to 0, and then reset the CLKGATE bit in the HW_CLKCTRL_SPDIF register to 0 .
29:21	**RSRVD1**	RO	0x00	Reserved
20:16	**DMAWAIT_COUNT**	RW	0x00	DMA Request Delay Count. This bit field specifies the number of APBX clock cycles (0 to 31) to delay before each DMA request. This field acts as a throttle on the bandwidth consumed by the SPDIF block. This field can be loaded by the DMA.
15:6	**RSRVD0**	RO	0x000	Reserved
5	**WAIT_END_XFER**	RW	0x1	Set this bit to a one if the SPDIF Transmitter should wait until the internal FIFO is empty before halting transmission based on deassertion of RUN. Use in conjuntion with HW_SPDIF_STAT_END_XFER to determine transfer completion
4	**WORD_LENGTH**	RW	0x0	Set this bit to one to enable 16-bit mode. Set this bit to zero for 32-bit mode. In either case, the SPDIF frame allows transmission of only 24 bits. In 16-bit mode, eight zeros will be appended to the LSB's of the input sample; in 32-bit mode, the 24 MSB's of HW_SPDIF_DATA will be transmitted.
3	**FIFO_UNDERFLOW_IRQ**	RW	0x0	This bit is set by hardware if the FIFO underflows during SPDIF transmission. Reset this bit by writing a one to the SCT clear address space and not by a general write.

32.3.2 5V to Battery Power Interaction

The i.MX23 supports several different options related to the interaction of the switching converters with the linear regulators. The two primary options are a reset on 5V insertion/removal or a handoff to the DC-DC converters that is invisible to the end-user of the application. Figure 32-3 includes these two options as the two system architecture decision boxes.

32.3.2.1 Battery Power to 5-V Power

By default, the DC-DC converter turns off when VDD5V becomes valid and the system does not reset. If the system is operating from the DC-DC converter and using more current than the linear regulators can supply, then the VDDD, VDDA, and VDDIO rails will droop when 5V is attached and the system may brownout and shut down. To avoid this issue, set the ENABLE_DCDC bit and set the LINREG_OFFSET fields to 0b2 in anticipation of VDD5V becoming present. The ENABLE_DCDC bit will cause the DC-DC converter to remain on even after 5V is connected and, thus, guarantee a stable supply voltage until the system is configured for removal of 5V. The LINREG_OFFSET fields = 0b2 cause the linear regulators to regulate to a lower target voltage than the switching converter and prevent unwanted interaction between the two power supplies. After the system is configured for removal of 5V, ENABLE_DCDC can be set low and ENABLE_ILIMIT set low in register HW_POWER_5VCTRL to allow the linear regulators to supply the system power, if desired.

32.3.2.2 5-V Power to Battery Power

Configuring the system for a 5-V-to-battery power handoff requires setup code to monitor the battery voltage as well as detect the removal of 5V.

Monitoring the battery voltage is performed by the LRADC. Typically, this involves programming the LRADC registers to periodically monitor the battery voltage as described in Chapter 33, “Low-Resolution ADC and Touch-Screen Interface.” The measured battery voltage should be written into the HW_POWER_BATTMONITOR register field BATT_VAL using the AUTOMATIC field in the HW_LRADC_CONVERSION register. Also, configuring battery brownout should be performed so that the system behaves as desired when 5V is no longer present and the battery is low.

The recommended method to detect removal of 5V requires setting VBUSVALID_5VDETECT and programming the detection threshold VBUSVALID_TRSH to 0x1 in HW_POWER_5VCTRL. Next, in order to minimize linear regulator and DC-DC converter interaction, it is necessary to set the LINREG_OFFSET = 0b2 in the HW_POWER_VDDIOCTRL, HW_POWER_VDDACTRL, and HW_POWER_VDDDCTRL registers. Finally, set DCDC_XFER and clear PWDN_5VBRNOUT in the HW_POWER_5VCTRL register. This sequence is important because it is safe to disable the powerdown-on-unplug functionality of the device only after the system is completely ready for a transition to battery power.

Table 37-30. HW_PINCTRL_DRIVE3 Bit Field Descriptions

BITS	LABEL	RW	RESET	DEFINITION
17:16	**BANK0_PIN28_MA**	RW	0x0	Pin 68, AUART1_RX pin output drive strength selection: 00= 4 mA; 01= 8 mA; 10= 12 mA; 11= reserved.
15:14	**RSRVD3**	RO	0x0	Always write zeroes to this field.
13:12	**BANK0_PIN27_MA**	RW	0x0	Pin 67, AUART1_RTS pin output drive strength selection: 00= 4 mA; 01= 8 mA; 10= 12 mA; 11= reserved.
11:10	**RSRVD2**	RO	0x0	Always write zeroes to this field.
9:8	**BANK0_PIN26_MA**	RW	0x0	Pin 66, AUART1_CTS pin output drive strength selection: 00= 4 mA; 01= 8 mA; 10= 12 mA; 11= reserved.
7:6	**RSRVD1**	RO	0x0	Always write zeroes to this field.
5:4	**BANK0_PIN25_MA**	RW	0x0	Pin 60, GPMI_RDN pin output drive strength selection: 00= 4 mA; 01= 8 mA; 10= 12 mA; 11= reserved.
3:2	**RSRVD0**	RO	0x0	Always write zeroes to this field.
1:0	**BANK0_PIN24_MA**	RW	0x0	Pin 65, GPMI_WRN pin output drive strength selection: 00= 4 mA; 01= 8 mA; 10= 12 mA; 11= reserved.

DESCRIPTION:

The Drive Strength and Voltage Register selects the drive strength and voltage for pins that are configured for output.

EXAMPLE:

```
Empty Example.
```

37.4.14 PINCTRL Drive Strength and Voltage Register 4 Description

The PINCTRL Drive Strength and Voltage Register selects the current drive strength for 8 pins of bank 1.

HW_PINCTRL_DRIVE4	0x240
HW_PINCTRL_DRIVE4_SET	0x244
HW_PINCTRL_DRIVE4_CLR	0x248
HW_PINCTRL_DRIVE4_TOG	0x24C