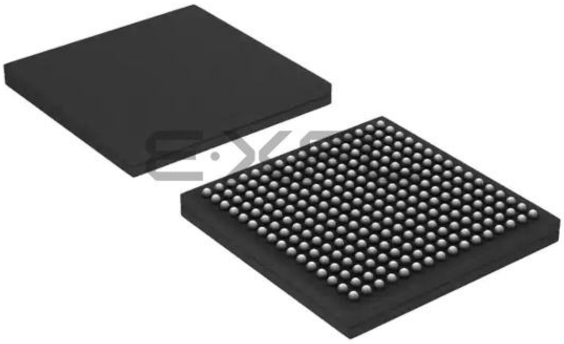
NXP USA Inc. - MK70FN1M0VMJ15 Datasheet



What is "Embedded - Microcontrollers"?

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

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Active
Core Processor	ARM® Cortex®-M4
Core Size	32-Bit Single-Core
Speed	150MHz
Connectivity	CANbus, EBI/EMI, Ethernet, I²C, IrDA, SD, SPI, UART/USART, USB, USB OTG
Peripherals	DMA, I²S, LCD, LVD, POR, PWM, WDT
Number of I/O	128
Program Memory Size	1MB (1M x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	128K x 8
Voltage - Supply (Vcc/Vdd)	1.71V ~ 3.6V
Data Converters	A/D 71x16b; D/A 2x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 105°C (TA)
Mounting Type	Surface Mount
Package / Case	256-LBGA
Supplier Device Package	256-MAPPBGA (17x17)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/mk70fn1m0vmj15

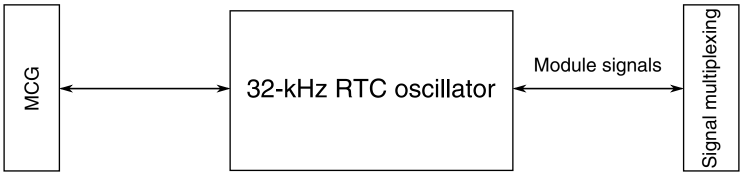


Figure 3-20. RTC OSC configuration

Table 3-36. Reference links to related information

Topic	Related module	Reference
Full description	RTC OSC	RTC OSC
Signal multiplexing	Port control	Signal multiplexing
Full description	MCG	MCG

3.5 Memories and Memory Interfaces

3.5.1 Flash Memory Configuration

This section summarizes how the module has been configured in the chip. For a comprehensive description of the module itself, see the module's dedicated chapter.

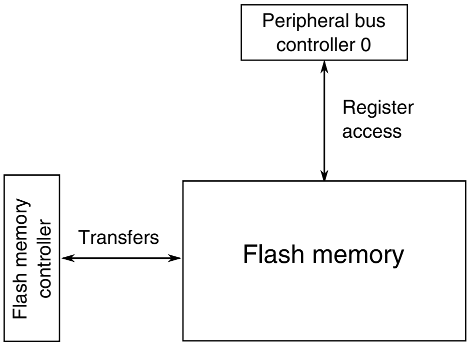


Figure 3-21. Flash memory configuration

Table 3-37. Reference links to related information

Topic	Related module	Reference
Full description	Flash memory	Flash memory
System memory map		System memory map
Clocking		Clock Distribution
Transfers	Flash memory controller	Flash memory controller
Register access	Peripheral bridge	Peripheral bridge

3.7.2 CMP Configuration

This section summarizes how the module has been configured in the chip. For a comprehensive description of the module itself, see the module's dedicated chapter.

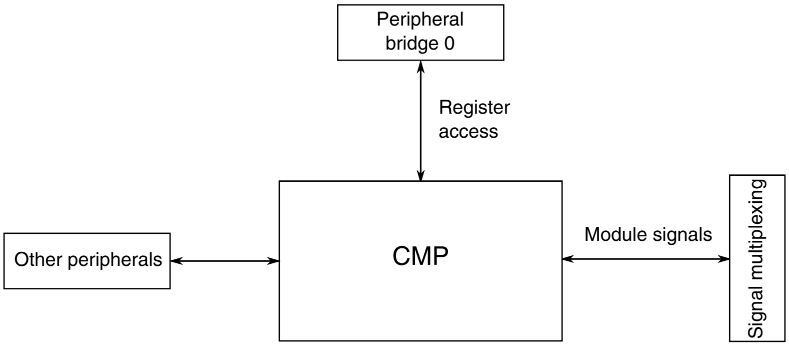


Figure 3-44. CMP configuration

Table 3-54. Reference links to related information

Topic	Related module	Reference
Full description	Comparator (CMP)	Comparator
System memory map		System memory map
Clocking		Clock distribution
Power management		Power management
Signal multiplexing	Port control	Signal multiplexing

3.7.2.1 CMP input connections

The following table shows the fixed internal connections to the CMP.

Table 3-55. CMP input connections

CMP Inputs	CMP0	CMP1	CMP2	CMP3
IN0	CMP0_IN0	CMP1_IN0	CMP2_IN0	CMP3_IN0
IN1	CMP0_IN1	CMP1_IN1	CMP2_IN1	CMP3_IN1
IN2	CMP0_IN2	ADC0SE16/CMP1_IN2	ADC1SE16/CMP2_IN2	CMP3_IN2
IN3	CMP0_IN3	12b DAC0 Reference/ CMP1_IN3	12b DAC1 Reference/ CMP2_IN3	12b DAC0Reference/ CMP2_IN3
IN4	12b DAC1 Reference	—	CMP2_IN4	CMP3_IN4
IN5	VREF Output/ CMP0_IN5	VREF Output/ CMP1_IN5	CMP2_IN5	CMP3_IN5

Table continues on the next page...

- PDB channel 0 pre-trigger 1 acknowledgement input: ADC0SC1A_COCO
- PDB channel 1 pre-trigger 0 acknowledgement input: ADC0SC1B_COCO
- PDB channel 1 pre-trigger 1 acknowledgement input: ADC1SC1A_COCO

- PDB channel 2 pre-trigger 0 acknowledgement input: ADC3SC1B_COCO
- PDB channel 2 pre-trigger 1 acknowledgement input: ADC2SC1A_COCO
- PDB channel 3 pre-trigger 0 acknowledgement input: ADC2SC1B_COCO
- PDB channel 3 pre-trigger 1 acknowledgement input: ADC3SC1A_COCO

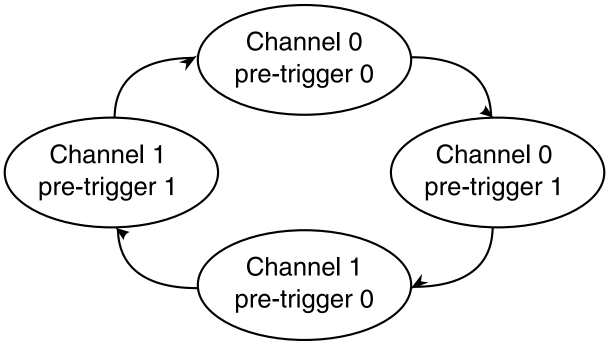


Figure 3-48. PDB back-to-back chain 1

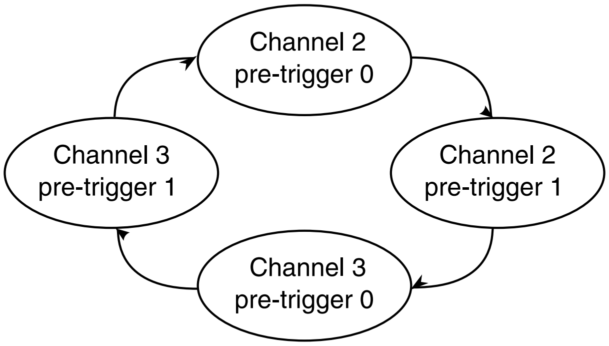


Figure 3-49. PDB back-to-back chain 2

When SIM_MCR[PDBLOOP]=1:

- PDB channel 0 pre-trigger 0 acknowledgement input: ADC3SC1B_COCO
- PDB channel 0 pre-trigger 1 acknowledgement input: ADC0SC1A_COCO
- PDB channel 1 pre-trigger 0 acknowledgement input: ADC0SC1B_COCO
- PDB channel 1 pre-trigger 1 acknowledgement input: ADC1SC1A_COCO
- PDB channel 2 pre-trigger 0 acknowledgement input: ADC1SC1B_COCO
- PDB channel 2 pre-trigger 1 acknowledgement input: ADC2SC1A_COCO
- PDB channel 3 pre-trigger 0 acknowledgement input: ADC2SC1B_COCO
- PDB channel 3 pre-trigger 1 acknowledgement input: ADC3SC1A_COCO

3.9.2 Universal Serial Bus (USB) FS Subsystem

The USB FS subsystem includes these components:

- Dual-role USB OTG-capable (On-The-Go) controller that supports a full-speed (FS) device or FS/LS host. The module complies with the USB 2.0 specification.
- USB transceiver that includes internal 15 kΩ pulldowns on the D+ and D- lines for host mode functionality.
- A 3.3 V regulator.
- USB device charger detection module.
- VBUS detect signal: To detect a valid VBUS in device mode, use a GPIO signal that can wake the chip in all power modes.

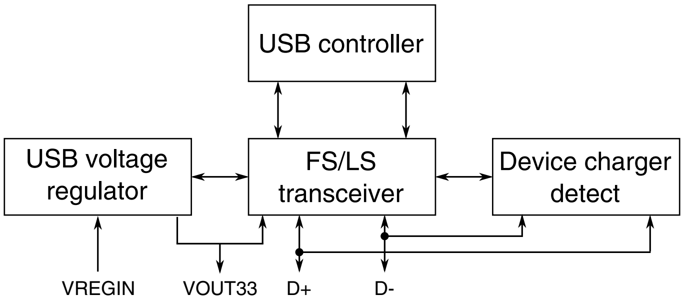


Figure 3-59. USB Subsystem Overview

3.9.2.1 USB Wakeup

When the USB detects that there is no activity on the USB bus for more than 3 ms, the INT_STAT[SLEEP] bit is set. This bit can cause an interrupt and software decides the appropriate action.

Waking from a low power mode (except in LLS/VLLS mode where USB is not powered) occurs through an asynchronous interrupt triggered by activity on the USB bus. Setting the USBTRC0[USBRESMEN] bit enables this function.

3.9.2.2 USB Power Distribution

This chip includes an internal 5 V to 3.3 V USB regulator that powers the USB transceiver or the MCU (depending on the application).

Table 32-19. Virtual-to-Physical Mappings of Different Flash, (continued)

Flash page size (main +spare) bytes	NFC_CFG [ECC MODE]	ECC bits	NFC_CFG [PAGE CNT]	Sector size (bytes)	Virtual page user size (bytes)	Mapping
4096 + 208	001	4	8	538	530	VirtualPage_0[529:0] = Physical[529:0] VirtualPage_1[529:0] = Physical[1067:538] VirtualPage_2[529:0] = Physical[1605:1076] VirtualPage_3[529:0] = Physical[2143:1614] VirtualPage_4[529:0] = Physical[2681:2152] VirtualPage_5[529:0] = Physical[3219:2690] VirtualPage_6[529:0] = Physical[3757:3228] VirtualPage_7[529:0] = Physical[4295:3766]
4096 + 208	010	6	8	538	526	VirtualPage_0[525:0] = Physical[525:0] VirtualPage_1[525:0] = Physical[1063:538] VirtualPage_2[525:0] = Physical[1601:1076] VirtualPage_3[525:0] = Physical[2139:1614] VirtualPage_4[525:0] = Physical[2677:2152] VirtualPage_5[525:0] = Physical[3215:2690] VirtualPage_6[525:0] = Physical[3753:3228] VirtualPage_7[525:0] = Physical[4291:3766]
4096 + 208	011	8	8	538	523	VirtualPage_0[522:0] = Physical[523:0] VirtualPage_1[522:0] = Physical[1060:538] VirtualPage_2[522:0] = Physical[1598:1076] VirtualPage_3[522:0] = Physical[2136:1614] VirtualPage_4[522:0] = Physical[2674:2152] VirtualPage_5[522:0] = Physical[3212:2690] VirtualPage_6[522:0] = Physical[3750:3228] VirtualPage_7[522:0] = Physical[4288:3766]

1. In most applications, this mode is of no use because user size is too small.
2. When 4KB page is split into eight virtual pages, if page program/read using DMA, set NFC_CFG[PAGECNT] to 8. if not using DMA, set NFC_CFG[PAGECNT] to 4. See Page Read and Page Program for details.

If flash devices with a physical page size of 4K or more are used, the bad block marker appears as the first byte of the spare area. But, because of the physical-to-virtual mapping, it does not appear in byte 2048 of the virtual page, where its logical place would be. The DMA engine contains the option to swap some bytes, and to make the bad block marker appear in the requested place.

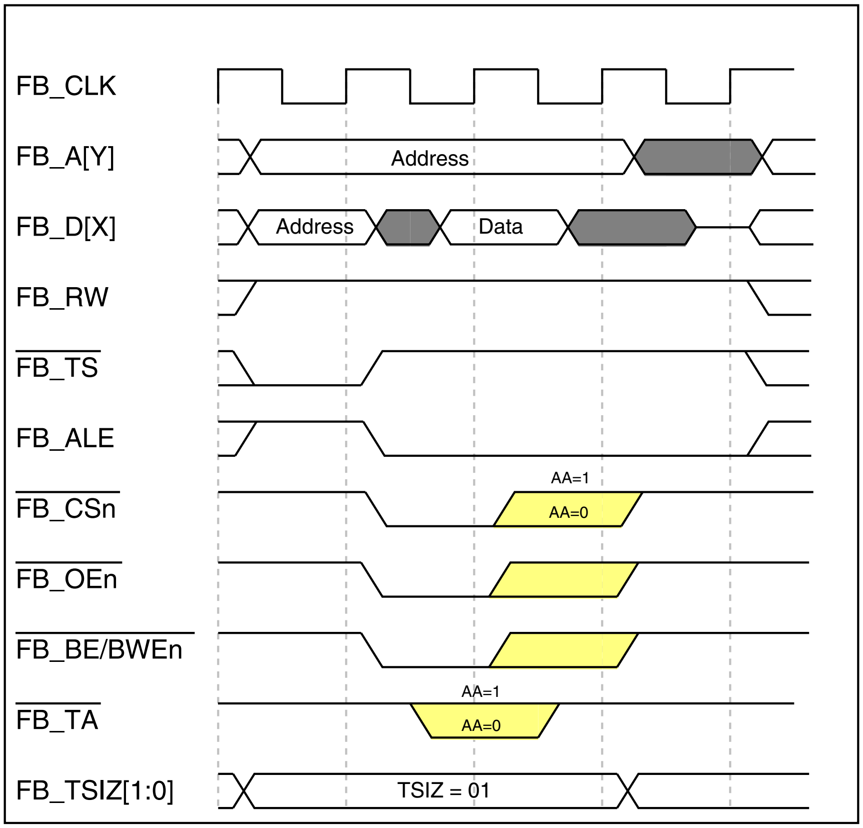


Figure 33-30. Single Byte-Read Transfer

The following figure shows the similar configuration for a write transfer. The data is driven from the second clock on FB_AD[31:24].

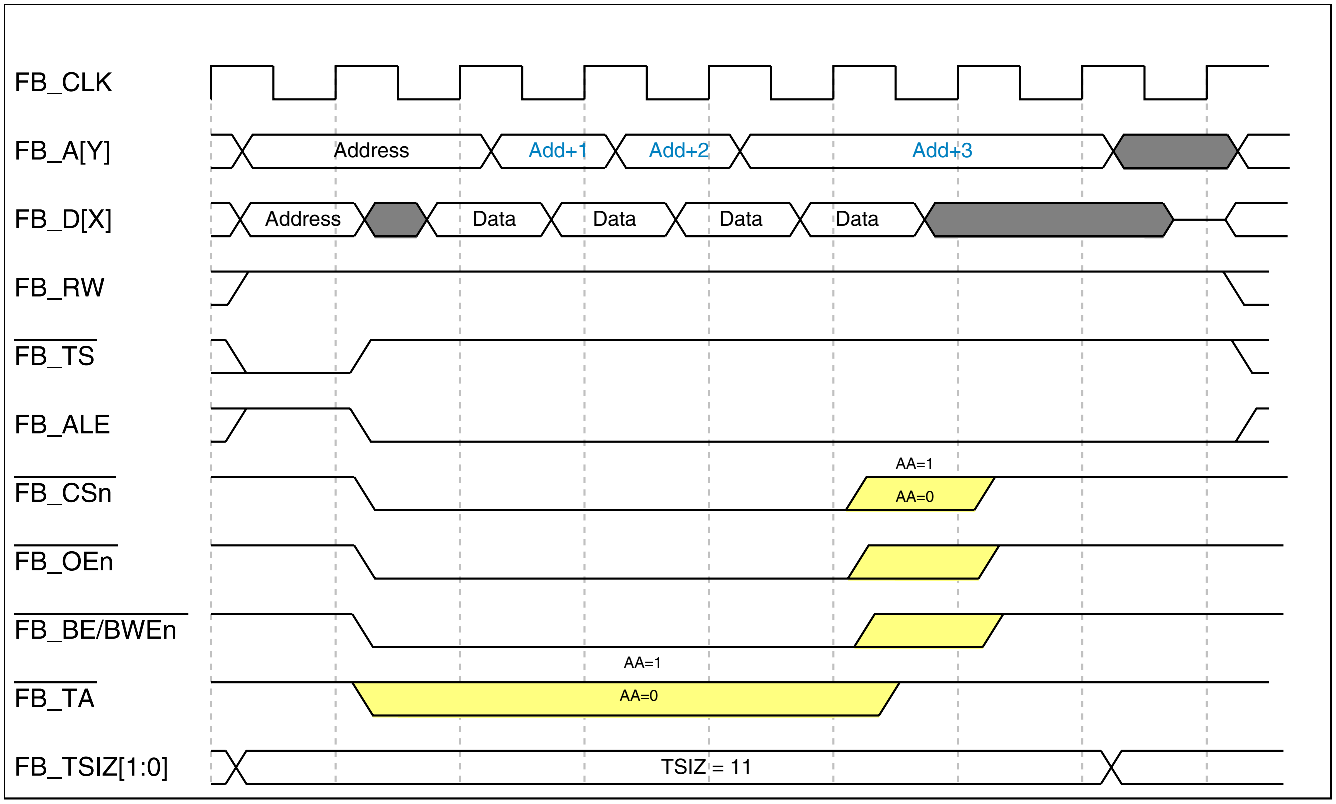


Figure 33-45. 32-bit-Read Burst from 8-Bit Port 2-1-1-1 (No Wait States)

The following figure shows a 32-bit write to an 8-bit device with burst enabled. The transfer results in a 4-beat burst and the data is driven on FB_AD[31:24]. The transfer size is driven at 32-bit (00) throughout the bus cycle.

Note

The first beat of any write burst cycle has at least one wait state. If the bus cycle is programmed for zero wait states (CSCR*n*[WS] = 0), one wait state is added. Otherwise, the programmed number of wait states are used.

DDR_CR20 field descriptions (continued)

Field	Description
	Clock stable delay on self refresh exit. Sets the number of cycles to hold the clock stable before exiting self-refresh mode. The clock will run for a minimum of cksrx cycles before CLK rises.
15–12 Reserved	Reserved This read-only field is reserved and always has the value zero. Must be cleared.
11–8 CKSRE	Clock hold delay on self refresh entry. Sets the number of cycles to hold the clock stable after entering self-refresh mode. The clock will run for a minimum of cksre cycles after CLK falls.
7–2 Reserved	Reserved This read-only field is reserved and always has the value zero. Must be cleared.
1–0 LPRE	Low Power Refresh enable Sets whether refreshes will occur while the memory controller is in one of the power-down modes. **NOTE:** The refreshes will not occur while in any of the self-refresh modes. **NOTE:** This parameter is active low. 00 Refreshes occur 01 Refreshes do not occur 10 Reserved 11 Reserved

34.4.22 DDR Control Register 21 (DDR_CR21)

Address: DDR_CR21 is 400A_E000h base + 54h offset = 400A_E054h

Bit	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
R W	MR1DAT0																MR0DAT0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0

DDR_CR21 field descriptions

Field	Description
31–16 MR1DAT0	Data to program into memory mode register 1 for chip select .
15–0 MR0DAT0	Data to program into memory mode register 0 for chip select .

DDR_CR50 field descriptions

Field	Description
31–17 Reserved	Reserved This read-only field is reserved and always has the value zero. Must be cleared.
16 CLKSTATUS	Clock Status Register access to clkstatus signal. 0 Disabled 1 Enabled
15–10 Reserved	Reserved This read-only field is reserved and always has the value zero. Must be cleared.
9–0 P2PRIRLX	Port 2 Priority Relax Counter value to trigger priority relax on port 2.

34.4.52 DDR Control Register 51 (DDR_CR51)

Address: DDR_CR51 is 400A_E000h base + CCh offset = 400A_E0CCh

<table>
<tr><td>Bit</td><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>R</td><td colspan="4">0</td><td colspan="4">PHYWRLAT</td><td colspan="8" rowspan="2">DLLRADLY</td><td colspan="16" rowspan="2">DLLRSTDLY</td></tr>
<tr><td>W</td><td colspan="4"></td><td colspan="4"></td></tr>
<tr><td>Reset</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
</table>

DDR_CR51 field descriptions

Field	Description
31–28 Reserved	Reserved This read-only field is reserved and always has the value zero. Must be cleared.
27–24 PHYWRLAT	PHY Write Latency Holds the calculated value of the t $_{phy\_wrlat}$ timing parameter, the number of DFI PHY clock cycles between when a write command is sent on the DFI control interface and when the *dfi_wrdata_en* signal is asserted. This parameter is used to adjust the dfi_wrdata_en signal timing. $t_{dfi\_phy\_wrlat} = t_{dfi\_phy\_wrlat\_base} + wrlat\_adj + reg\_dimm\_enable$ minus WRLAT_WIDTH'h3 **NOTE:** Values of $t_{dfi\_phy\_wrlat\_base} + wrlat\_adj$ that are less than 3 are not supported. All DFI timing parameters must be programmed relative to the DFI clock.
23–16 DLLRADLY	DLL Reset Adjust Delay Minimum number of cycles after setting master delay in DLL until reset is released.
15–0 DLLRSTDLY	DLL Reset Delay Minimum number of cycles required for DLL reset.

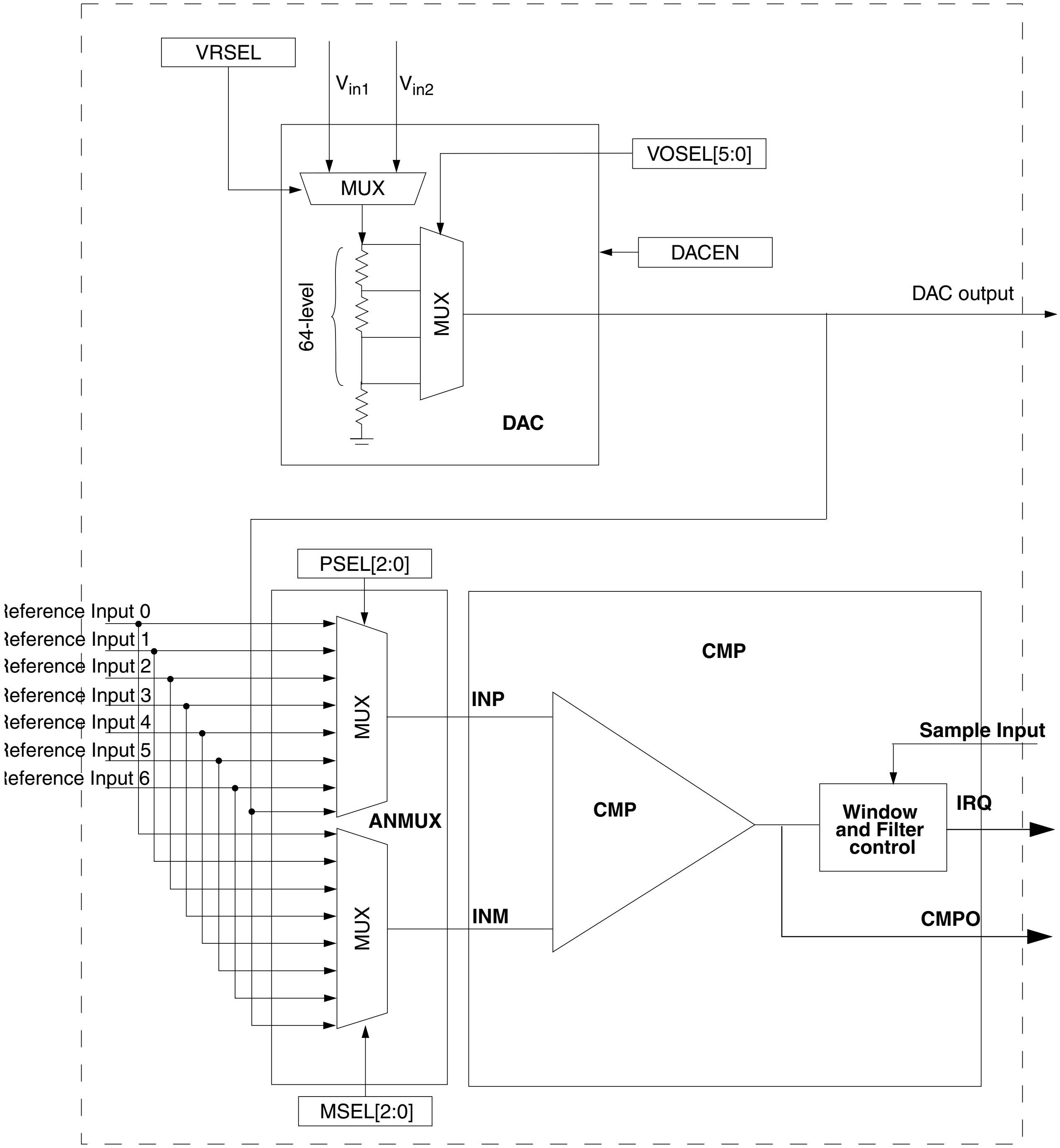
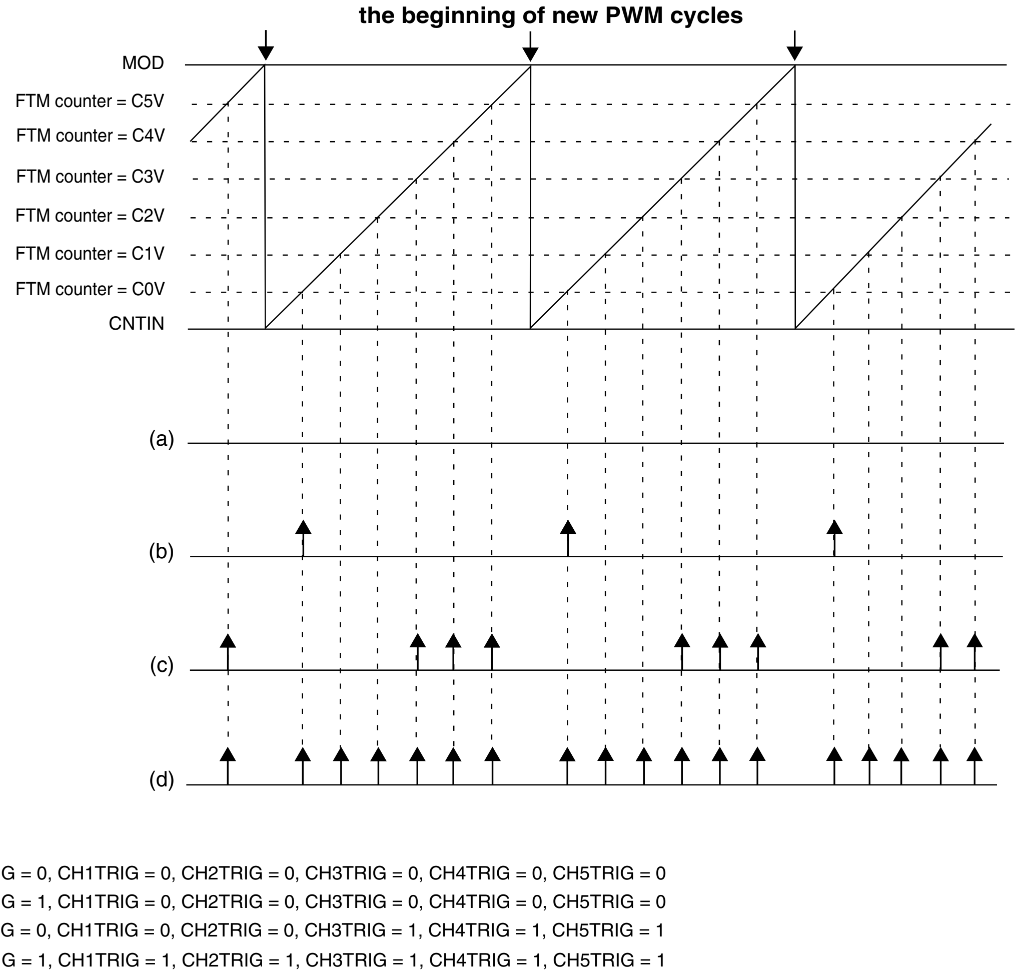


Figure 40-1. CMP, DAC and ANMUX Blocks Diagram

40.6 CMP Block Diagram

The following figure shows the block diagram for the Comparator module.

The FTM is able to generate multiple triggers in one PWM period. Since each trigger is generated for a specific channel, several channels are required to implement this functionality. This behavior is described in the following figure.



NOTE

(a) CH0TRIG = 0, CH1TRIG = 0, CH2TRIG = 0, CH3TRIG = 0, CH4TRIG = 0, CH5TRIG = 0
(b) CH0TRIG = 1, CH1TRIG = 0, CH2TRIG = 0, CH3TRIG = 0, CH4TRIG = 0, CH5TRIG = 0
(c) CH0TRIG = 0, CH1TRIG = 0, CH2TRIG = 0, CH3TRIG = 1, CH4TRIG = 1, CH5TRIG = 1
(d) CH0TRIG = 1, CH1TRIG = 1, CH2TRIG = 1, CH3TRIG = 1, CH4TRIG = 1, CH5TRIG = 1

Figure 44-279. Channel Match Trigger

Note

It is expected that the channel match trigger be used only in combine mode.

44.4.21 Initialization Trigger

If INITTRIGEN = 1, then the FTM generates a trigger when the FTM counter is updated with the CNTIN register value in the following cases.

- The FTM counter is automatically updated with the CNTIN register value by the selected counting mode.

- Stops sending data for the amount of time specified by the pause quanta in 512 bit time increments
- Sets ENET*n*_TCR[RFC_PAUSE]

Frame transfer resumes when the time specified by the quanta expireds and if no new quanta value is received or if a new pause frame with a quanta value set to 0x0000 is received. The MAC also resets RFC_PAUSE to zero.

If ENET*n*_RCR[FCE] cleared, the MAC ignores received pause frames.

Optionally and independent of ENET*n*_RCR[FCE], pause frames are forwarded to the client interface if PAUFWD is set.

49.4.6.2 Local Device/FIFO Congestion

The MAC transmit engine generates pause frames when the local receive FIFO is not able to receive more than a pre-defined number of words (FIFO programmable threshold) or when pause frame generation is requested by the local host processor.

- To generate a pause frame, the host processor sets ENET*n*_TCR[TFC_PAUSE]. A single pause frame is generated when the current frame transfer is completed and TFC_PAUSE is automatically cleared. Optionally, an interrupt is generated.
- A XOFF pause frame is generated when the receive FIFO asserts its section empty flag (internal). A XOFF pause frame is generated automatically, when the current frame transfer completes.
- A XON pause frame is generated when the receive FIFO deasserts its section empty flag (internal). A XON pause frame is generated automatically, when the current frame transfer completes.

When a XOFF pause frame is generated, the pause quanta (payload byte P1 and P2) is filled with the value programmed in ENET*n*_OPD[PAUSE_DUR].

USBHS_USB_SBUSCFG field descriptions (continued)

Field	Description
	101 INCR4 + INCR4 + INCR4 + INCR4 + INCR4 + INCR unspec. length. 110 INCR8 + INCR8 + INCR4 + INCR unspec. length. 111 INCR16 + INCR4 + INCR unspec. length. 001 INCR4 + INCR4 + INCR4 + INCR4 + INCR4 + SINGLE + SINGLE. 010 INCR8 + INCR8 + INCR4 + SINGLE + SINGLE. 011 INCR16 + INCR4 + SINGLE + SINGLE. When this field is different from zero, the values in the TXBURST/RXBURST bitfields in the USB_BURSTSIZE register are ignored by the controller. Internally the BURSTMODE is set to the value of the INCRx burst. Since this has a direct relation with the burst sizes you must be careful with AHB burst selected. Although the TXBURST/RXBURST are bypassed, this register can be written/read with no effect while the BURSTMODE field is non-zero. **NOTE:** Setting the BURSTMODE value to 000 might cause bus allocation during BULK or ISO transfers. **NOTE:** Changing this BURSTMODE field while a transaction is in progress yields undefined results. One possible way to prevent undefined results is to clear the Run/Stop (RS) bit in the USB_USBCMD register, after the HCHALTED is detected in USB_USBSTS. 000 INCR burst of unspecified length 001 INCR4, non-multiple transfers of INCR4 is decomposed into singles. 010 INCR8, non-multiple transfers of INCR8, is decomposed into INCR4 or singles. 011 INCR16, non-multiple transfers of INCR16, is decomposed into INCR8, INCR4 or singles. 100 Reserved, do not use. 101 INCR4, non-multiple transfers of INCR4 is decomposed into smaller unspecified length bursts. 110 INCR8, non-multiple transfers of INCR8 is decomposed into smaller unspecified length bursts. 111 INCR16, non-multiple transfers of INCR16 is decomposed into smaller unspecified length bursts.

53.3.10 Host Controller Interface Version and Capability Registers Length Register (USBHS_HCIVERSION)

Contains the CAPLENGTH field used as an offset to add to the register base address to find the beginning of the operational register space, the location of the USBCMD register. Also contains a BCD encoding of the EHCI revision number supported by this OTG controller. The most-significant byte of the HCIVERSION field represents a major revision; the least-significant byte is the minor revision.

Address: USBHS_HCIVERSION is 4003_4000h base + 100h offset = 4003_4100h

Bit	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
R	HCIVERSION																0								CAPLENGTH							
W																																
Reset	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0

Clearing one of those three flags does not affect the state of the other two.

An interrupt is generated if an IFLAG bit is asserted and the corresponding mask bit is asserted too.

A powerful filtering scheme is provided to accept only frames intended for the target application, thus reducing the interrupt servicing work load. The filtering criteria is specified by programming a table of up to 128 32-bit registers, according to CTRL2[RFFN] setting, that can be configured to one of the following formats (see also Rx FIFO Structure):

- Format A: 128 IDAFs (extended or standard IDs including IDE and RTR)
- Format B: 256 IDAFs (standard IDs or extended 14-bit ID slices including IDE and RTR)
- Format C: 512 IDAFs (standard or extended 8-bit ID slices)

Note

A chosen format is applied to all entries of the filter table. It is not possible to mix formats within the table.

Every frame available in the FIFO has a corresponding IDHIT (Identifier Acceptance Filter Hit Indicator) that can be read by accessing the RXFIR register. The RXFIR[IDHIT] field refers to the message at the output of the FIFO and is valid while the IFLAG[BUF5I] flag is asserted. The RXFIR register must be read only before clearing the flag, which guarantees that the information refers to the correct frame within the FIFO.

Up to thirty two elements of the filter table are individually affected by the Individual Mask Registers (RXIMRx), according to the setting of CTRL2[RFFN], allowing very powerful filtering criteria to be defined. If the IRMQ bit is negated, then the FIFO filter table is affected by RXFGMASK.

54.4.8 CAN Protocol Related Features

This section describes the CAN protocol related features.

57.4.2.2 Transmission bit order

When the S2[MSBF] bit is set, the UART automatically transmits the MSB of the data word as the first bit after the start bit. Likewise the LSB of the data word is transmitted immediately preceding the parity bit (or the stop bit if parity is not enabled). All necessary bit ordering is handled automatically by the module hence the format of the data written to the D register for transmission is completely independent of the S2[MSBF] setting.

57.4.2.3 Character transmission

To transmit data, the MCU writes the data bits to the UART transmit buffer using UART data registers (C3[T8]/D). Data in the transmit buffer is then in turn transferred to the transmitter shift register as needed. The transmit shift register then shifts a frame out through the transmit data output signal after it has prefaced it with any required start and stop bits. The UART data registers (C3[T8] and D) provide access to the transmit buffer structure.

The UART also sets a flag, the transmit data register empty flag (S1[TDRE]) and generates interrupt or DMA request (C5[TDMAS]), whenever the number of datawords in the transmit buffer is equal to or less than the value indicated by the TWFIFO[TXWATER]. The transmit driver routine may respond to this flag by writing additional datawords to the transmit buffer using (C3[T8]/D) as space permits.

See Application information for specific programing sequences.

Setting the C2[TE] bit automatically loads the transmit shift register with a preamble of 10 logic 1s (if C1[M] = 0), 11 logic 1s (if C1[M] = 1 and C4[M10] = 0), or 12 logic 1s (if C1[M] = 1, C4[M10] = 1, C1[PE] = 1). After the preamble shifts out, control logic transfers the data from the UART data register into the transmit shift register. The transmitter automatically transmits the correct start bit and stop bit before and after the dataword.

When C7816[ISO_7816E] = 1 setting the C2[TE] bit does not result in a preamble being generated. The transmitter starts transmitting as soon as the corresponding guard time expires. When C7816[TTYPE] = 0 the value in GT is used, when C7816[TTYPE] = 1 the value BGT is used since it is assumed that the C2[TE] will remain asserted until the end of the block transfer. The C2[TE] bit is automatically cleared when in C7816[TTYPE] = 1 and the block being transmitted has been completed. When C7816[TTYPE] = 0, the transmitter listens for a NACK indication. If no NACK is received it is assumed that character was correctly received. If a NACK is received the transmitter will resend the data, assuming that the number of retries for that character (number of NACKs received) is less than or equal to the value in ET7816[TXTHRESHOLD].

57.4.8.5 Baud rate generation

The value in WF7816[GTFD] does not impact the clock frequency. The SBR and BRFD are used to generate the clock frequency. This clock frequency is used by the UART only and is not seen by the device (smartcard). The transmitter clocks operates at 1/16 the frequency of the receive clock so that the receiver is able to sample the received value 16 times during the ETU.

57.4.8.6 UART restrictions in ISO-7816 operation

Due to the flexibility of the UART module, there are several features and interrupts that are not supported while running in ISO-7816 mode. These restrictions are documented within the register bit definitions.

57.4.9 Infrared interface

The UART provides the capability of transmitting narrow pulses to an IR LED and receiving narrow pulses and transforming them to serial bits, which are sent to the UART. The IrDA physical layer specification defines a half-duplex infrared communication link for exchanging data. The full standard includes data rates up to 16 Mbits/s. This design covers data rates only between 2.4 kbits/s and 115.2 kbits/s.

The UART has an infrared transmit encoder and receive decoder. The UART transmits serial bits of data which are encoded by the infrared submodule to transmit a narrow pulse for every zero bit. No pulse is transmitted for every one bit. When receiving data, the IR pulses are detected using an IR photo diode and transformed to CMOS levels by the IR receive decoder (external from the MCU). The narrow pulses are then stretched by the infrared receive decoder to get back to a serial bit stream to be received by the UART. The polarity of transmitted pulses and expected receive pulses can be inverted so that a direct connection can be made to external IrDA transceiver modules that use active low pulses.

The infrared submodule receives its clock sources from the UART. One of these two clocks are selected in the infrared submodule in order to generate either 3/16, 1/16, 1/32 or 1/4 narrow pulses during transmission.

Table 57-357. UART interrupt sources (continued)

Interrupt Source	Flag	Local enable	DMA select
Receiver	LBKDIF	LBKDIE	-
Receiver	RXEDGIF	RXEDGIE	-
Receiver	OR	ORIE	-
Receiver	NF	NEIE	-
Receiver	FE	FEIE	-
Receiver	PF	PEIE	-
Receiver	RXUF	RXUFE	-
Transmitter	TXOF	TXOFE	-
Receiver	WT	WTWE	-
Receiver	CWT	CWTE	-
Receiver	BWT	BWTE	-
Receiver	INITD	INITDE	-
Receiver	TXT	TXTE	-
Receiver	RXT	RXTE	-
Receiver	GTV	GTVE	-

57.6.1 RXEDGIF description

The S2[RXEDGIF] is set when an active edge is detected on the RxD pin. Hence, the active edge can only be detected when in two wire mode. A RXEDGIF interrupt is only generated when S2[RXEDGIF] is set. If RXEDGIE is not enabled prior to S2[RXEDGIF] getting set, an interrupt is not generated until S2[RXEDGIF] bit gets set.

57.6.1.1 RxD edge detect sensitivity

Edge sensitivity can be software programmed to be either falling or rising. The polarity of the edge sensitivity is selected using the S2[RXINV] bit. To detect falling edge S2[RXINV] is programmed to zero and to detect rising edge S2[RXINV] is programmed to one.

Synchronizing logic is used prior to detect edges. Prior to detecting an edge, the receive data on RxD input must be at the de-asserted logic level. A falling edge is detected when the RxD input signal is seen as a logic 1 (the deasserted level) during one module clock cycle and then a logic 0 (the asserted level) during the next cycle. A rising edge is detected when the input is seen as a logic 0 during one module clock cycle and then a logic 1 during the next cycle.

When the write operation is paused, the data transfer inside the host system is not stopped, and the transfer is active until the data buffer is full. Because of this (if not needed), it is recommended to avoid using the suspend command for the SDIO card. This is because when such a command is sent, the SDHC thinks the system will switch to another function on the SDIO card, and flush the data buffer. The SDHC takes the resume command as a normal command with data transfer, and it is left for the driver to set all the relevant registers before the transfer is resumed. If there is only one block to send when the transfer is resumed, the XFERTYP[MSBSEL] and XFERTYP[BCEN] bits are set as well as the XFERTYP[AC12EN] bit. However, the SDHC will automatically send a CMD12 to mark the end of the multi-block transfer.

58.6.3.2 Block read

This section discusses the block read access methods.

58.6.3.2.1 Normal read

For block reads, the basic unit of data transfer is a block whose maximum size is stored in areas defined by the corresponding card specification. A CRC is appended to the end of each block, ensuring data transfer integrity. The CMD17, CMD18, CMD53, CMD60, CMD61, and so on, can initiate a block read. After completing the transfer, the card returns to the transfer state. For multi blocks read, data blocks will be continuously transferred until a stop command is issued.

The software flow to read from a card incorporates the internal DMA and the read operation is a multi-block read with the Auto CMD12 enabled. For the other two methods (by means of external DMA or CPU polling status) with different transfer methods, the internal DMA parts should be removed and the alternative steps should be straightforward.

The software flow to read from a card is described below:

1. Check the card status, wait until card is ready for data.
2. Set the card block length/size:
   a. For SD/MMC, use SET_BLOCKLEN (CMD16)
   b. For SDIO cards or the I/O portion of SDCombo cards, use IO_RW_DIRECT(CMD52) to set the I/O block size bit field in the CCCR register (for function 0) or FBR register (for functions 1~7)
   c. For CE-ATA cards, configure bits 1~0 in the scrControl register

I2Sx_RFR*n* field descriptions (continued)

Field	Description
19–16 WFP	Write FIFO pointer FIFO write pointer for receive data channel.
15–4 Reserved	This read-only field is reserved and always has the value zero.
3–0 RFP	Read FIFO pointer FIFO read pointer for receive data channel.

59.3.18 SAI Receive Mask Register (I2Sx_RMR)

This register is double-buffered and updates when the receive enable bit is first set and then at the end of each frame. This allows the masked words in each frame to change from frame to frame.

Addresses: I2S0_RMR is 4002_F000h base + E0h offset = 4002_F0E0h
I2S1_RMR is 400A_F000h base + E0h offset = 400A_F0E0h

Bit	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
R	RWM																															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

I2Sx_RMR field descriptions

Field	Description
31–0 RWM	Receive word mask For each word in the frame, configures if the receive word is masked. 0 Word N is enabled. 1 Word N is masked.

Updated Time Alarm section with IER[TAIE].

A.49 ENET changes

- In MAC Features: replaced "Supports" with "Compliant with the" in AMD magic packet bullet.

A.50 USB changes

- No substantial content changes

A.51 USBDCD changes

- No substantial content changes

A.52 USB VREG changes

- No substantial content changes

A.53 USB high speed OTG controller changes

Updated these registers:

- Identification Register (USBHS_ID)
- General Hardware Parameters Register (USBHS_HWGENERAL)
- Host Hardware Parameters Register (USBHS_HWHOST)
- Transmit Buffer Hardware Parameters Register (USBHS_HWTXBUF)
- Receive Buffer Hardware Parameters Register (USBHS_HWRXBUF)
- System Bus Interface Configuration Register (USBHS_USB_SBUSCFG): new
- Host Controller Interface Version Register (USBHS_HCIVERSION)
- Capability Registers Length Register (USBHS_CAPLENGTH): moved-- The CAPLENGTH field is now included within the USBHS_HCIVERSION register.
- Host Controller Structural Parameters Register (USBHS_HCSPARAMS)
- USB Status Register (USBHS_USBSTS)
- USB Interrupt Enable Register (USBHS_USBINTR)
- Device Address Register (USBHS_DEVICEADDR)
- Host TT Asynchronous Buffer Control (USBHS_TTCTRL)
- Master Interface Data Burst Size Register (USBHS_BURSTSIZE)
- Endpoint NAK Register (USBHS_ENDPTNAK) new
- Endpoint NAK Enable Register (USBHS_ENDPTNAKEN) new
- Port Status and Control Registers (USBHS_PORTSC1)
- On-the-Go Status and Control Register (USBHS_OTGSC)
- USB Mode Register (USBHS_USBMODE)
- Endpoint Setup Status Register (USBHS_EPSETUPSR)