

### 3.3.8.2 Memory maps

The peripheral bridges are used to access the registers of most of the modules on this device. See [AIPS0 Memory Map](#) and [AIPS1 Memory Map](#) for the memory slot assignment for each module.

### 3.3.8.3 AIPS\_Lite MPRA register reset value

- AIPS<sub>x</sub>\_MPRA reset value is 0x7770\_0000

Therefore, masters 0, 1, and 2 are trusted bus masters after reset.

### 3.3.8.4 AIPS\_Lite PACRA-P and PACRU register reset values

The reset values for the AIPS\_Lite registers are listed in the following table.

Register name	AIPS0 reset value	AIPS1 reset value
PACRA	0x5000_4000	0x5000_0000
PACRB	0x4400_4400	0x0000_0000
PACRC	0x0000_0000	0x0000_0000
PACRD	0x0000_0004	0x0000_0000
PACRE	0x4444_4444	0x4444_4444
PACRF	0x4444_4444	0x4444_4444
PACRG	0x4444_4444	0x4444_4444
PACRH	0x4444_4444	0x4444_4444
PACRI	0x4444_4444	0x4444_4444
PACRJ	0x4444_4444	0x4444_4444
PACRK	0x4444_4444	0x4444_4444
PACRL	0x4444_4444	0x4444_4444
PACRM	0x4444_4444	0x4444_4444
PACRN	0x4444_4444	0x4444_4444
PACRO	0x4444_4444	0x4444_4444
PACRP	0x4444_4444	0x4444_4444
PACRU	0x4400_0000	0x4400_0000

### 3.3.9 DMA request multiplexer 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.

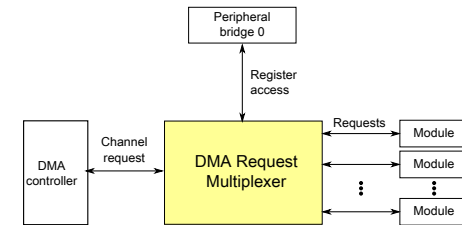


Figure 3-14. DMA request multiplexer configuration

Table 3-25. Reference links to related information

Topic	Related module	Reference
Full description	DMA request multiplexer	<a href="#">DMA Mux</a>
System memory map		<a href="#">System memory map</a>
Clocking		<a href="#">Clock distribution</a>
Power management		<a href="#">Power management</a>
Channel request	DMA controller	<a href="#">DMA Controller</a>
Requests		<a href="#">DMA request sources</a>

### 3.3.9.1 DMA MUX request sources

This device includes a DMA request mux that allows up to 63 DMA request signals to be mapped to any of the 16 DMA channels. Because of the mux there is not a hard correlation between any of the DMA request sources and a specific DMA channel.

Table 3-26. DMA request sources - MUX 0

Source number	Source module	Source description
0	—	Channel disabled <sup>1</sup>
1	Reserved	Not used
2	UART0	Receive
3	UART0	Transmit
4	UART1	Receive
5	UART1	Transmit
6	UART2	Receive
7	UART2	Transmit
8	UART3	Receive
9	UART3	Transmit

Table continues on the next page...

Table 3-26. DMA request sources - MUX 0 (continued)

Source number	Source module	Source description
10	UART4	Transmit or Receive
11	UART5	Transmit or Receive
12	I <sup>2</sup> S0	Receive
13	I <sup>2</sup> S0	Transmit
14	SPI0	Receive
15	SPI0	Transmit
16	SPI1	Transmit or Receive
17	SPI2	Transmit or Receive
18	I <sup>2</sup> C0	—
19	I <sup>2</sup> C1 or I <sup>2</sup> C2	—
20	FTM0	Channel 0
21	FTM0	Channel 1
22	FTM0	Channel 2
23	FTM0	Channel 3
24	FTM0	Channel 4
25	FTM0	Channel 5
26	FTM0	Channel 6
27	FTM0	Channel 7
28	FTM1	Channel 0
29	FTM1	Channel 1
30	FTM2	Channel 0
31	FTM2	Channel 1
32	FTM3	Channel 0
33	FTM3	Channel 1
34	FTM3	Channel 2
35	FTM3	Channel 3
36	FTM3	Channel 4
37	FTM3	Channel 5
38	FTM3	Channel 6
39	FTM3	Channel 7
40	ADC0	—
41	ADC1	—
42	CMP0	—
43	CMP1	—
44	CMP2	—
45	DAC0	—
46	DAC1	—
47	CMT	—
48	PDB	—

Table continues on the next page...

Table 3-26. DMA request sources - MUX 0 (continued)

Source number	Source module	Source description
49	Port control module	Port A
50	Port control module	Port B
51	Port control module	Port C
52	Port control module	Port D
53	Port control module	Port E
54	IEEE 1588 Timers	Timer 0
55	IEEE 1588 Timers	Timer 1
56	IEEE 1588 Timers	Timer 2
57	IEEE 1588 Timers	Timer 3
58	DMA MUX	Always enabled
59	DMA MUX	Always enabled
60	DMA MUX	Always enabled
61	DMA MUX	Always enabled
62	DMA MUX	Always enabled
63	DMA MUX	Always enabled

1. Configuring a DMA channel to select source 0 or any of the reserved sources disables that DMA channel.

### 3.3.9.2 DMA transfers via PIT trigger

The PIT module can trigger a DMA transfer on the first four DMA channels. The assignments are detailed at [PIT/DMA Periodic Trigger Assignments](#).

### 3.3.10 DMA Controller 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.

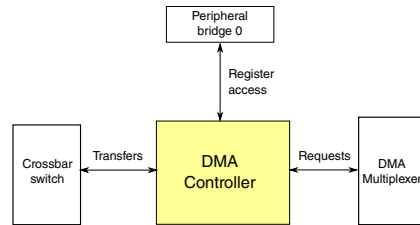


Figure 3-15. DMA Controller configuration

Table 3-27. Reference links to related information

Topic	Related module	Reference
Full description	DMA Controller	<a href="#">DMA Controller</a>
System memory map		<a href="#">System memory map</a>
Register access	Peripheral bridge (AIPS-Lite 0)	<a href="#">AIPS-Lite 0</a>
Clocking		<a href="#">Clock distribution</a>
Power management		<a href="#">Power management</a>
Transfers	Crossbar switch	<a href="#">Crossbar switch</a>

### 3.3.11 External Watchdog Monitor (EWM) 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.

### 3.6.3 RNG 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.

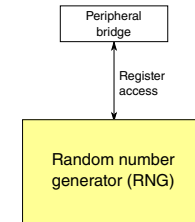


Figure 3-34. RNG configuration

Table 3-47. Reference links to related information

Topic	Related module	Reference
Full description	RNG	<a href="#">RNG</a>
System memory map		<a href="#">System memory map</a>
Clocking		<a href="#">Clock distribution</a>
Power management		<a href="#">Power management</a>

#### 3.6.3.1 RNGA Base Addresses

RNGA can be accessed through both AIPS0 and AIPS1. When accessed through AIPS0, the base address is 4002\_9000h and when accessed through AIPS1, the base address is 400A\_0000h.

## 3.7 Analog

### 3.7.1 16-bit SAR ADC 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.

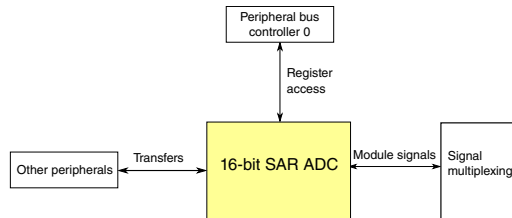


Figure 3-35. 16-bit SAR ADC configuration

Table 3-48. Reference links to related information

Topic	Related module	Reference
Full description	16-bit SAR ADC	<a href="#">16-bit SAR ADC</a>
System memory map		<a href="#">System memory map</a>
Clocking		<a href="#">Clock distribution</a>
Power management		<a href="#">Power management</a>
Signal multiplexing	Port control	<a href="#">Signal multiplexing</a>

### 3.7.1.1 ADC instantiation information

This device contains two ADCs.

#### 3.7.1.1.1 Number of ADC channels

The number of ADC channels present on the device is determined by the pinout of the specific device package. For details regarding the number of ADC channel available on a particular package, refer to the signal multiplexing chapter of this MCU.

### 3.7.1.2 DMA Support on ADC

Applications may require continuous sampling of the ADC (4K samples/sec) that may have considerable load on the CPU. Though using PDB to trigger ADC may reduce some CPU load, the ADC supports DMA request functionality for higher performance when the ADC is sampled at a very high rate or cases where PDB is bypassed. The ADC can trigger the DMA (via DMA req) on conversion completion.

### 3.7.1.3 Connections/channel assignment

#### 3.7.1.3.1 ADC0 Connections/Channel Assignment

##### NOTE

As indicated by the following sections, each ADC<sub>x</sub>\_DP<sub>x</sub> input and certain ADC<sub>x</sub>\_DM<sub>x</sub> inputs may operate as single-ended ADC channels in single-ended mode.

##### 3.7.1.3.1.1 ADC0 Channel Assignment for 144-Pin Package

ADC Channel (SC1n[ADCH])	Channel	Input signal (SC1n[DIFF]= 1)	Input signal (SC1n[DIFF]= 0)
00000	DAD0	ADC0_DP0 and ADC0_DM0 <sup>1</sup>	ADC0_DP0 <sup>2</sup>
00001	DAD1	ADC0_DP1 and ADC0_DM1 <sup>3</sup>	ADC0_DP1
00010	DAD2	ADC0_DP2 and ADC0_DM2	ADC0_DP2
00011	DAD3	ADC0_DP3 and ADC0_DM3 <sup>4</sup>	ADC0_DP3 <sup>5</sup>
00100 <sup>6</sup>	AD4a	Reserved	Reserved
00101 <sup>6</sup>	AD5a	Reserved	Reserved
00110 <sup>6</sup>	AD6a	Reserved	Reserved
00111 <sup>6</sup>	AD7a	Reserved	Reserved
00100 <sup>6</sup>	AD4b	Reserved	ADC0_SE4b
00101 <sup>6</sup>	AD5b	Reserved	ADC0_SE5b
00110 <sup>6</sup>	AD6b	Reserved	ADC0_SE6b
00111 <sup>6</sup>	AD7b	Reserved	ADC0_SE7b
01000	AD8	Reserved	ADC0_SE8 <sup>7</sup>
01001	AD9	Reserved	ADC0_SE9 <sup>8</sup>
01010	AD10	Reserved	ADC0_SE10
01011	AD11	Reserved	ADC0_SE11
01100	AD12	Reserved	ADC0_SE12
01101	AD13	Reserved	ADC0_SE13
01110	AD14	Reserved	ADC0_SE14
01111	AD15	Reserved	ADC0_SE15
10000	AD16	Reserved	ADC0_SE16
10001	AD17	Reserved	ADC0_SE17
10010	AD18	Reserved	ADC0_SE18
10011	AD19	Reserved	ADC0_DM0 <sup>9</sup>
10100	AD20	Reserved	ADC0_DM1
10101	AD21	Reserved	ADC0_SE21
10110	AD22	Reserved	ADC0_SE22
10111	AD23	Reserved	12-bit DAC0 Output/ ADC0_SE23
11000	AD24	Reserved	Reserved

Table continues on the next page...

### Chapter 3 Chip Configuration

ADC Channel (SC1n[ADCH])	Channel	Input signal (SC1n[DIFF]= 1)	Input signal (SC1n[DIFF]= 0)
11001	AD25	Reserved	Reserved
11010	AD26	Temperature Sensor (Diff)	Temperature Sensor (S.E)
11011	AD27	Bandgap (Diff) <sup>10</sup>	Bandgap (S.E) <sup>10</sup>
11100	AD28	Reserved	Reserved
11101	AD29	-VREFH (Diff)	VREFH (S.E)
11110	AD30	Reserved	VREFL
11111	AD31	Module Disabled	Module Disabled

1. Interleaved with ADC1\_DP3 and ADC1\_DM3
2. Interleaved with ADC1\_DP3
3. Interleaved with ADC0\_DP2 and ADC1\_SE6a
4. Interleaved with ADC1\_DP0 and ADC1\_DM0
5. Interleaved with ADC1\_DP0
6. ADCx\_CFG2[MUXSEL] bit selects between ADCx\_SEn channels a and b. Refer to MUXSEL description in ADC chapter for details.
7. Interleaved with ADC1\_SE8
8. Interleaved with ADC1\_SE9
9. Interleaved with ADC1\_DM3
10. This is the PMC bandgap 1V reference voltage not the VREF module 1.2 V reference voltage. Prior to reading from this ADC channel, ensure that you enable the bandgap buffer by setting the PMC\_REGSC[BGBE] bit. Refer to the device data sheet for the bandgap voltage (V<sub>BG</sub>) specification.

#### 3.7.1.3.1.2 ADC0 Channel Assignment for 121-Pin Package

ADC Channel (SC1n[ADCH])	Channel	Input signal (SC1n[DIFF]= 1)	Input signal (SC1n[DIFF]= 0)
00000	DAD0	ADC0_DP0 and ADC0_DM0 <sup>1</sup>	ADC0_DP0 <sup>2</sup>
00001	DAD1	ADC0_DP1 and ADC0_DM1	ADC0_DP1
00010	DAD2	ADC0_DP2 and ADC0_DM2	ADC0_DP2
00011	DAD3	ADC0_DP3 and ADC0_DM3 <sup>3</sup>	ADC0_DP3 <sup>4</sup>
00100 <sup>5</sup>	AD4a	Reserved	Reserved
00101 <sup>5</sup>	AD5a	Reserved	Reserved
00110 <sup>5</sup>	AD6a	Reserved	Reserved
00111 <sup>5</sup>	AD7a	Reserved	Reserved
00100 <sup>5</sup>	AD4b	Reserved	ADC0_SE4b
00101 <sup>5</sup>	AD5b	Reserved	ADC0_SE5b
00110 <sup>5</sup>	AD6b	Reserved	ADC0_SE6b
00111 <sup>5&gt;</sup>	AD7b	Reserved	ADC0_SE7b
01000	AD8	Reserved	ADC0_SE8 <sup>6</sup>
01001	AD9	Reserved	ADC0_SE9 <sup>7</sup>
01010	AD10	Reserved	Reserved
01011	AD11	Reserved	Reserved
01100	AD12	Reserved	ADC0_SE12
01101	AD13	Reserved	ADC0_SE13

Table continues on the next page...

### Analog

ADC Channel (SC1n[ADCH])	Channel	Input signal (SC1n[DIFF]= 1)	Input signal (SC1n[DIFF]= 0)
01110	AD14	Reserved	ADC0_SE14
01111	AD15	Reserved	ADC0_SE15
10000	AD16	Reserved	ADC0_SE16
10001	AD17	Reserved	ADC0_SE17
10010	AD18	Reserved	ADC0_SE18
10011	AD19	Reserved	ADC0_DM0 <sup>8</sup>
10100	AD20	Reserved	ADC0_DM1
10101	AD21	Reserved	ADC0_SE21
10110	AD22	Reserved	ADC0_SE22
10111	AD23	Reserved	12-bit DAC0 Output/ ADC0_SE23
11000	AD24	Reserved	Reserved
11001	AD25	Reserved	Reserved
11010	AD26	Temperature Sensor (Diff)	Temperature Sensor (S.E)
11011	AD27	Bandgap (Diff) <sup>9</sup>	Bandgap (S.E) <sup>9</sup>
11100	AD28	Reserved	Reserved
11101	AD29	-VREFH (Diff)	VREFH (S.E)
11110	AD30	Reserved	VREFL
11111	AD31	Module Disabled	Module Disabled

1. Interleaved with ADC1\_DP3 and ADC1\_DM3
2. Interleaved with ADC1\_DP3
3. Interleaved with ADC1\_DP0 and ADC1\_DM0
4. Interleaved with ADC1\_DP0
5. ADCx\_CFG2[MUXSEL] bit selects between ADCx\_SEn channels a and b. Refer to MUXSEL description in ADC chapter for details.
6. Interleaved with ADC1\_SE8
7. Interleaved with ADC1\_SE9
8. Interleaved with ADC1\_DM3
9. This is the PMC bandgap 1V reference voltage not the VREF module 1.2 V reference voltage. Prior to reading from this ADC channel, ensure that you enable the bandgap buffer by setting the PMC\_REGSC[BGBE] bit. Refer to the device data sheet for the bandgap voltage (V<sub>BG</sub>) specification.

#### 3.7.1.3.1.3 ADC0 Channel Assignment for 100-Pin Package

ADC Channel (SC1n[ADCH])	Channel	Input signal (SC1n[DIFF]= 1)	Input signal (SC1n[DIFF]= 0)
00000	DAD0	ADC0_DP0 and ADC0_DM0 <sup>1</sup>	ADC0_DP0 <sup>2</sup>
00001	DAD1	ADC0_DP1 and ADC0_DM1	ADC0_DP1
00010	DAD2	ADC0_DP2 and ADC0_DM2	ADC0_DP2
00011	DAD3	ADC0_DP3 and ADC0_DM3 <sup>3</sup>	ADC0_DP3 <sup>4</sup>
00100 <sup>5</sup>	AD4a	Reserved	Reserved
00101 <sup>5</sup>	AD5a	Reserved	Reserved
00110 <sup>5</sup>	AD6a	Reserved	Reserved

Table continues on the next page...

ADC Channel (SC1n[ADCH])	Channel	Input signal (SC1n[DIFF]= 1)	Input signal (SC1n[DIFF]= 0)
00111 <sup>5</sup>	AD7a	Reserved	Reserved
00100 <sup>5</sup>	AD4b	Reserved	ADC0_SE4b
00101 <sup>5</sup>	AD5b	Reserved	ADC0_SE5b
00110 <sup>5</sup>	AD6b	Reserved	ADC0_SE6b
00111 <sup>5</sup>	AD7b	Reserved	ADC0_SE7b
01000	AD8	Reserved	ADC0_SE8 <sup>6</sup>
01001	AD9	Reserved	ADC0_SE9 <sup>7</sup>
01010	AD10	Reserved	Reserved
01011	AD11	Reserved	Reserved
01100	AD12	Reserved	ADC0_SE12
01101	AD13	Reserved	ADC0_SE13
01110	AD14	Reserved	ADC0_SE14
01111	AD15	Reserved	ADC0_SE15
10000	AD16	Reserved	Reserved
10001	AD17	Reserved	ADC0_SE17
10010	AD18	Reserved	ADC0_SE18
10011	AD19	Reserved	ADC0_DM0 <sup>8</sup>
10100	AD20	Reserved	ADC0_DM1
10101	AD21	Reserved	Reserved
10110	AD22	Reserved	Reserved
10111	AD23	Reserved	12-bit DAC0 Output/ ADC0_SE23
11000	AD24	Reserved	Reserved
11001	AD25	Reserved	Reserved
11010	AD26	Temperature Sensor (Diff)	Temperature Sensor (S.E)
11011	AD27	Bandgap (Diff) <sup>9</sup>	Bandgap (S.E) <sup>9</sup>
11100	AD28	Reserved	Reserved
11101	AD29	-VREFH (Diff)	VREFH (S.E)
11110	AD30	Reserved	VREFL
11111	AD31	Module Disabled	Module Disabled

1. Interleaved with ADC1\_DP3 and ADC1\_DM3

2. Interleaved with ADC1\_DP3

3. Interleaved with ADC1\_DP0 and ADC1\_DM0

4. Interleaved with ADC1\_DP0

5. ADCx\_CFG2[MUXSEL] bit selects between ADCx\_SEn channels a and b. Refer to MUXSEL description in ADC chapter for details.

6. Interleaved with ADC1\_SE8

7. Interleaved with ADC1\_SE9

8. Interleaved with ADC1\_DM3

9. This is the PMC bandgap 1V reference voltage not the VREF module 1.2 V reference voltage. Prior to reading from this ADC channel, ensure that you enable the bandgap buffer by setting the PMC\_REGSC[BGBE] bit. Refer to the device data sheet for the bandgap voltage (V<sub>BG</sub>) specification.

### 3.7.1.3.2 ADC1 Connections/Channel Assignment

#### NOTE

As indicated in the following tables, each ADCx\_DPx input and certain ADCx\_DMx inputs may operate as single-ended ADC channels in single-ended mode.

#### 3.7.1.3.2.1 ADC1 Channel Assignment for 144-Pin Package

ADC Channel (SC1n[ADCH])	Channel	Input signal (SC1n[DIFF]= 1)	Input signal (SC1n[DIFF]= 0)
00000	DAD0	ADC1_DP0 and ADC1_DM0 <sup>1</sup>	ADC1_DP0 <sup>2</sup>
00001	DAD1	ADC1_DP1 and ADC1_DM1	ADC1_DP1
00010	DAD2	Reserved	Reserved
00011	DAD3	ADC1_DP3 and ADC1_DM3 <sup>3</sup>	ADC1_DP3 <sup>4,5</sup>
00100 <sup>5</sup>	AD4a	Reserved	ADC1_SE4a
00101 <sup>5</sup>	AD5a	Reserved	ADC1_SE5a
00110 <sup>5</sup>	AD6a	Reserved	ADC1_SE6a
00111 <sup>5</sup>	AD7a	Reserved	ADC1_SE7a
00100 <sup>5</sup>	AD4b	Reserved	ADC1_SE4b
00101 <sup>5</sup>	AD5b	Reserved	ADC1_SE5b
00110 <sup>5</sup>	AD6b	Reserved	ADC1_SE6b
00111 <sup>5</sup>	AD7b	Reserved	ADC1_SE7b
01000	AD8	Reserved	ADC1_SE8 <sup>6</sup>
01001	AD9	Reserved	ADC1_SE9 <sup>7</sup>
01010	AD10	Reserved	ADC1_SE10
01011	AD11	Reserved	ADC1_SE11
01100	AD12	Reserved	ADC1_SE12
01101	AD13	Reserved	ADC1_SE13
01110	AD14	Reserved	ADC1_SE14
01111	AD15	Reserved	ADC1_SE15
10000	AD16	Reserved	ADC1_SE16
10001	AD17	Reserved	ADC1_SE17
10010	AD18	Reserved	VREF Output/ADC1_SE18
10011	AD19	Reserved	ADC1_DM0 <sup>8</sup>
10100	AD20	Reserved	ADC1_DM1
10101	AD21	Reserved	Reserved
10110	AD22	Reserved	Reserved
10111	AD23	Reserved	12-bit DAC1 Output/ ADC1_SE23
11000	AD24	Reserved	Reserved
11001	AD25	Reserved	Reserved

Table continues on the next page...

### Chapter 3 Chip Configuration

ADC Channel (SC1n[ADCH])	Channel	Input signal (SC1n[DIFF]= 1)	Input signal (SC1n[DIFF]= 0)
11010	AD26	Temperature Sensor (Diff)	Temperature Sensor (S.E)
11011	AD27	Bandgap (Diff) <sup>9</sup>	Bandgap (S.E) <sup>9</sup>
11100	AD28	Reserved	Reserved
11101	AD29	-VREFH (Diff)	VREFH (S.E)
11110	AD30	Reserved	VREFL
11111	AD31	Module Disabled	Module Disabled

1. Interleaved with ADC0\_DP3 and ADC0\_DM3
2. Interleaved with ADC0\_DP3
3. Interleaved with ADC0\_DP0 and ADC0\_DM0
4. Interleaved with ADC0\_DP0
5. ADCx\_CFG2[MUXSEL] bit selects between ADCx\_SEn channels a and b. Refer to MUXSEL description in ADC chapter for details.
6. Interleaved with ADC0\_SE8
7. Interleaved with ADC0\_SE9
8. Interleaved with ADC0\_DM3
9. This is the PMC bandgap 1V reference voltage not the VREF module 1.2 V reference voltage. Prior to reading from this ADC channel, ensure that you enable the bandgap buffer by setting the PMC\_REGSC[BGBE] bit. Refer to the device data sheet for the bandgap voltage (V<sub>BG</sub>) specification.

#### 3.7.1.3.2.2 ADC1 Channel Assignment for 121-Pin Package

ADC Channel (SC1n[ADCH])	Channel	Input signal (SC1n[DIFF]= 1)	Input signal (SC1n[DIFF]= 0)
00000	DAD0	ADC1_DP0 and ADC1_DM0 <sup>1</sup>	ADC1_DP0 <sup>2</sup>
00001	DAD1	ADC1_DP1 and ADC1_DM1	ADC1_DP1
00010	DAD2	Reserved	Reserved
00011	DAD3	ADC1_DP3 and ADC1_DM3 <sup>3</sup>	ADC1_DP3 <sup>4</sup>
00100 <sup>5</sup>	AD4a	Reserved	ADC1_SE4a
00101 <sup>5</sup>	AD5a	Reserved	ADC1_SE5a
00110 <sup>5</sup>	AD6a	Reserved	ADC1_SE6a
00111 <sup>5</sup>	AD7a	Reserved	ADC1_SE7a
00100 <sup>5</sup>	AD4b	Reserved	ADC1_SE4b
00101 <sup>5</sup>	AD5b	Reserved	ADC1_SE5b
00110 <sup>5</sup>	AD6b	Reserved	ADC1_SE6b
00111 <sup>5</sup>	AD7b	Reserved	ADC1_SE7b
01000	AD8	Reserved	ADC1_SE8 <sup>6</sup>
01001	AD9	Reserved	ADC1_SE9 <sup>7</sup>
01010	AD10	Reserved	Reserved
01011	AD11	Reserved	Reserved
01100	AD12	Reserved	ADC1_SE12
01101	AD13	Reserved	ADC1_SE13
01110	AD14	Reserved	ADC1_SE14

Table continues on the next page...

### Analog

ADC Channel (SC1n[ADCH])	Channel	Input signal (SC1n[DIFF]= 1)	Input signal (SC1n[DIFF]= 0)
01111	AD15	Reserved	ADC1_SE15
10000	AD16	Reserved	ADC1_SE16
10001	AD17	Reserved	ADC1_SE17
10010	AD18	Reserved	VREF Output/ADC1_SE18
10011	AD19	Reserved	ADC1_DM0 <sup>8</sup>
10100	AD20	Reserved	ADC1_DM1
10101	AD21	Reserved	Reserved
10110	AD22	Reserved	Reserved
10111	AD23	Reserved	12-bit DAC1 Output/ ADC1_SE23
11000	AD24	Reserved	Reserved
11001	AD25	Reserved	Reserved
11010	AD26	Temperature Sensor (Diff)	Temperature Sensor (S.E)
11011	AD27	Bandgap (Diff) <sup>9</sup>	Bandgap (S.E) <sup>9</sup>
11100	AD28	Reserved	Reserved
11101	AD29	-VREFH (Diff)	VREFH (S.E)
11110	AD30	Reserved	VREFL
11111	AD31	Module Disabled	Module Disabled

1. Interleaved with ADC0\_DP3 and ADC0\_DM3
2. Interleaved with ADC0\_DP3
3. Interleaved with ADC0\_DP0 and ADC0\_DM0
4. Interleaved with ADC0\_DP0
5. ADCx\_CFG2[MUXSEL] bit selects between ADCx\_SEn channels a and b. Refer to MUXSEL description in ADC chapter for details.
6. Interleaved with ADC0\_SE8
7. Interleaved with ADC0\_SE9
8. Interleaved with ADC0\_DM3
9. This is the PMC bandgap 1V reference voltage not the VREF module 1.2 V reference voltage. Prior to reading from this ADC channel, ensure that you enable the bandgap buffer by setting the PMC\_REGSC[BGBE] bit. Refer to the device data sheet for the bandgap voltage (V<sub>BG</sub>) specification.

#### 3.7.1.3.2.3 ADC1 Channel Assignment for 100-Pin Package

ADC Channel (SC1n[ADCH])	Channel	Input signal (SC1n[DIFF]= 1)	Input signal (SC1n[DIFF]= 0)
00000	DAD0	ADC1_DP0 and ADC1_DM0 <sup>1</sup>	ADC1_DP0 <sup>2</sup>
00001	DAD1	ADC1_DP1 and ADC1_DM1	ADC1_DP1
00010	DAD2	Reserved	Reserved
00011	DAD3	ADC1_DP3 and ADC1_DM3 <sup>3</sup>	ADC1_DP3 <sup>4</sup>
00100 <sup>5</sup>	AD4a	Reserved	ADC1_SE4a
00101 <sup>5</sup>	AD5a	Reserved	ADC1_SE5a
00110 <sup>5</sup>	AD6a	Reserved	ADC1_SE6a
00111 <sup>5</sup>	AD7a	Reserved	ADC1_SE7a

Table continues on the next page...

ADC Channel (SC1n[ADCH])	Channel	Input signal (SC1n[DIFF]= 1)	Input signal (SC1n[DIFF]= 0)
00100 <sup>5</sup>	AD4b	Reserved	ADC1_SE4b
00101 <sup>5</sup>	AD5b	Reserved	ADC1_SE5b
00110 <sup>5</sup>	AD6b	Reserved	ADC1_SE6b
00111 <sup>5</sup>	AD7b	Reserved	ADC1_SE7b
01000	AD8	Reserved	ADC1_SE8 <sup>6</sup>
01001	AD9	Reserved	ADC1_SE9 <sup>7</sup>
01010	AD10	Reserved	Reserved
01011	AD11	Reserved	Reserved
01100	AD12	Reserved	Reserved
01101	AD13	Reserved	Reserved
01110	AD14	Reserved	ADC1_SE14
01111	AD15	Reserved	ADC1_SE15
10000	AD16	Reserved	Reserved
10001	AD17	Reserved	ADC1_SE17
10010	AD18	Reserved	VREF Output/ADC1_SE18
10011	AD19	Reserved	ADC1_DM0 <sup>8</sup>
10100	AD20	Reserved	ADC1_DM1
10101	AD21	Reserved	Reserved
10110	AD22	Reserved	Reserved
10111	AD23	Reserved	Reserved
11000	AD24	Reserved	Reserved
11001	AD25	Reserved	Reserved
11010	AD26	Temperature Sensor (Diff)	Temperature Sensor (S.E)
11011	AD27	Bandgap (Diff) <sup>9</sup>	Bandgap (S.E) <sup>9</sup>
11100	AD28	Reserved	Reserved
11101	AD29	-VREFH (Diff)	VREFH (S.E)
11110	AD30	Reserved	VREFL
11111	AD31	Module Disabled	Module Disabled

1. Interleaved with ADC0\_DP3 and ADC0\_DM3
2. Interleaved with ADC0\_DP3
3. Interleaved with ADC0\_DP0 and ADC0\_DM0
4. Interleaved with ADC0\_DP0
5. ADCx\_CFG2[MUXSEL] bit selects between ADCx\_SEn channels a and b. Refer to MUXSEL description in ADC chapter for details.
6. Interleaved with ADC0\_SE8
7. Interleaved with ADC0\_SE9
8. Interleaved with ADC0\_DM3
9. This is the PMC bandgap 1V reference voltage not the VREF module 1.2 V reference voltage. Prior to reading from this ADC channel, ensure that you enable the bandgap buffer by setting the PMC\_REGSC[BGBE] bit. Refer to the device data sheet for the bandgap voltage ( $V_{BG}$ ) specification.

### 3.7.1.4 ADC Channels MUX Selection

The following figure shows the assignment of ADCx\_SEn channels a and b through a MUX selection to ADC. To select between alternate set of channels, refer to ADCx\_CFG2[MUXSEL] bit settings for more details.

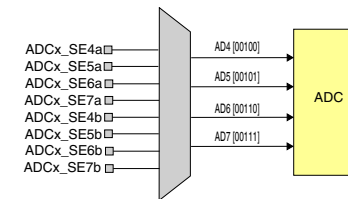


Figure 3-36. ADCx\_SEn channels a and b selection

### 3.7.1.5 ADC Hardware Interleaved Channels

The AD8 and AD9 channels on ADCx are interleaved in hardware using the following configuration.

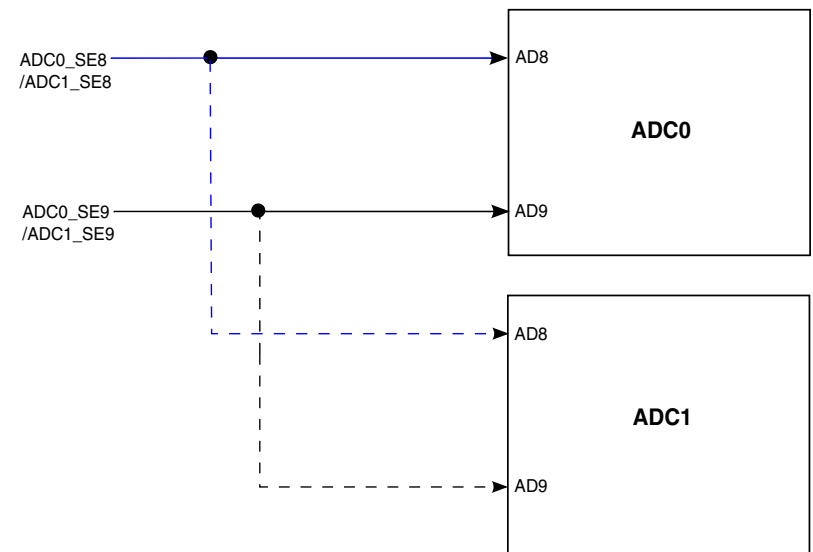


Figure 3-37. ADC hardware interleaved channels integration



### 3.7.1.6 ADC Reference Options

The ADC supports the following references:

- VREFH/VREFL - connected as the primary reference option
- 1.2 V VREF\_OUT - connected as the  $V_{ALT}$  reference option

ADCx\_SC2[REFSEL] bit selects the voltage reference sources for ADC. Refer to REFSEL description in ADC chapter for more details.

### 3.7.1.7 ADC triggers

The ADC supports both software and hardware triggers. The primary hardware mechanism for triggering the ADC is the PDB. The PDB itself can be triggered by other peripherals. For example: RTC (Alarm, Seconds) signal is connected to the PDB. The PDB input trigger can receive the RTC (alarm/seconds) trigger forcing ADC conversions in run mode (where PDB is enabled). On the other hand, the ADC can conduct conversions in low power modes, not triggered by PDB. This allows the ADC to do conversions in low power mode and store the output in the result register. The ADC generates interrupt when the data is ready in the result register that wakes the system from low power mode. The PDB can also be bypassed by using the ADCxTRGSEL bits in the SIM\_SOPT7 register.

For operation of triggers in different modes, refer to Power Management chapter.

### 3.7.1.8 Alternate clock

For this device, the alternate clock is connected to OSCERCLK.

#### NOTE

This clock option is only usable when OSCERCLK is in the MHz range. A system with OSCERCLK in the kHz range has the optional clock source below minimum ADC clock operating frequency.

### 3.7.1.9 ADC low-power modes

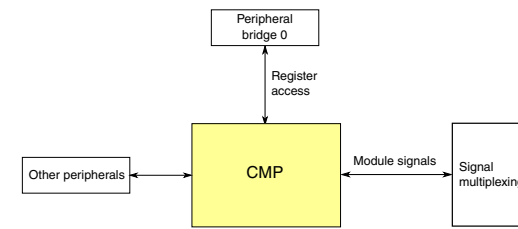
This table shows the ADC low-power modes and the corresponding chip low-power modes.

**Table 3-49. ADC low-power modes**

Module mode	Chip mode
Wait	Wait, VLPW
Normal Stop	Stop, VLPS
Low Power Stop	LLS, VLLS3, VLLS2, VLLS1, VLLS0

### 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 3-38. CMP configuration**

**Table 3-50. Reference links to related information**

Topic	Related module	Reference
Full description	Comparator (CMP)	<a href="#">Comparator</a>
System memory map		<a href="#">System memory map</a>
Clocking		<a href="#">Clock distribution</a>
Power management		<a href="#">Power management</a>
Signal multiplexing	Port control	<a href="#">Signal multiplexing</a>

### 3.7.2.1 CMP input connections

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

**Table 3-51. CMP input connections**

CMP Inputs	CMP0	CMP1	CMP2
IN0	CMP0_IN0	CMP1_IN0	CMP2_IN0
IN1	CMP0_IN1	CMP1_IN1	CMP2_IN1

*Table continues on the next page...*

Table 3-51. CMP input connections (continued)

CMP Inputs	CMP0	CMP1	CMP2
IN2	CMP0_IN2	ADC0_SE16/CMP1_IN2 <sup>1</sup>	ADC1_SE16/CMP2_IN2 <sup>1</sup>
IN3	CMP0_IN3	12-bit DAC0_OUT/CMP1_IN3	12-bit DAC1_OUT/ CMP2_IN3 <sup>1</sup>
IN4	12-bit DAC1_OUT/CMP0_IN4	—	—
IN5	VREF Output/CMP0_IN5	VREF Output/CMP1_IN5	—
IN6	Bandgap	Bandgap	Bandgap
IN7	6b DAC0 Reference	6b DAC1 Reference	—

1. Reserved on the 100 LQFP package.

### 3.7.2.2 CMP external references

The 6-bit DAC sub-block supports selection of two references. For this device, the references are connected as follows:

- VREF\_OUT -  $V_{in1}$  input
- VDD -  $V_{in2}$  input

### 3.7.2.3 External window/sample input

Individual PDB pulse-out signals control each CMP Sample/Window timing.

### 3.7.3 12-bit DAC 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.

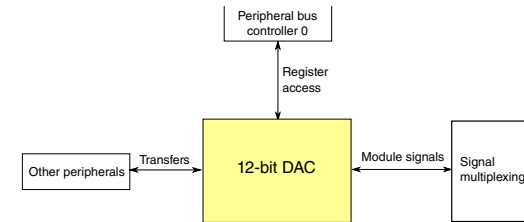


Figure 3-39. 12-bit DAC configuration

Table 3-52. Reference links to related information

Topic	Related module	Reference
Full description	12-bit DAC	<a href="#">12-bit DAC</a>
System memory map		<a href="#">System memory map</a>
Clocking		<a href="#">Clock distribution</a>
Power management		<a href="#">Power management</a>
Signal multiplexing	Port control	<a href="#">Signal multiplexing</a>

### 3.7.3.1 12-bit DAC Overview

This device contains two 12-bit digital-to-analog converters (DAC) with programmable reference generator output. The DAC includes a FIFO for DMA support.

### 3.7.3.2 12-bit DAC instantiation

In this chip, the 100-pin package has 1 DAC module. The 121-pin and 144-pin packages have 2 DAC modules.

### 3.7.3.3 12-bit DAC Output

The output of the DAC can be placed on an external pin or set as one of the inputs to the analog comparator or ADC.

### 3.7.3.4 12-bit DAC Reference

For this device VREF\_OUT and VDDA are selectable as the DAC reference. VREF\_OUT is connected to the DACREF\_1 input and VDDA is connected to the DACREF\_2 input. Use DACx\_C0[DACRFS] control bit to select between these two options.

Be aware that if the DAC and ADC use the VREF\_OUT reference simultaneously, some degradation of ADC accuracy is to be expected due to DAC switching.

### 3.7.3.5 DAC0 Base Addresses

DAC0 can be accessed through both AIPS0 and AIPS1. When accessed through AIPS0, the base address is 4003\_F000h and when accessed through AIPS1, the base address is 400C\_C000h.

## 3.7.4 VREF 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.

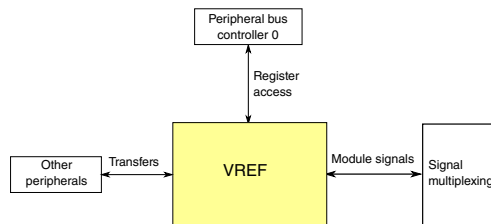


Figure 3-40. VREF configuration

Table 3-53. Reference links to related information

Topic	Related module	Reference
Full description	VREF	<a href="#">VREF</a>
System memory map		<a href="#">System memory map</a>
Clocking		<a href="#">Clock distribution</a>
Power management		<a href="#">Power management</a>
Signal multiplexing	Port control	<a href="#">Signal multiplexing</a>

### 3.7.4.1 VREF Overview

This device includes a voltage reference (VREF) to supply an accurate 1.2 V voltage output.

The voltage reference can provide a reference voltage to external peripherals or a reference to analog peripherals, such as the ADC, DAC, or CMP.

#### NOTE

PMC\_REGSC[BGEN] bit must be set if the VREF regulator is required to remain operating in VLPx modes.

#### NOTE

For either an internal or external reference if the VREF\_OUT functionality is being used, VREF\_OUT signal must be connected to an output load capacitor. Refer the device data sheet for more details.

## 3.8 Timers

### 3.8.1 PDB 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.

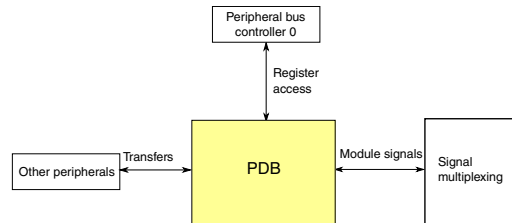


Figure 3-41. PDB configuration

Table 3-54. Reference links to related information

Topic	Related module	Reference
Full description	PDB	<a href="#">PDB</a>
System memory map		<a href="#">System memory map</a>
Clocking		<a href="#">Clock distribution</a>
Power management		<a href="#">Power management</a>
Signal multiplexing	Port control	<a href="#">Signal multiplexing</a>

### 3.8.1.1 PDB Instantiation

#### 3.8.1.1.1 PDB Output Triggers

Table 3-55. PDB output triggers

Number of PDB channels for ADC trigger	2
Number of pre-triggers per PDB channel	2
Number of DAC triggers	2
Number of Pulse Out	3

#### 3.8.1.1.2 PDB Input Trigger Connections

Table 3-56. PDB Input Trigger Options

PDB Trigger	PDB Input
0000	External Trigger
0001	CMP 0
0010	CMP 1
0011	CMP 2
0100	PIT Ch 0 Output
0101	PIT Ch 1 Output

Table continues on the next page...

Table 3-56. PDB Input Trigger Options (continued)

PDB Trigger	PDB Input
0110	PIT Ch 2 Output
0111	PIT Ch 3 Output
1000	FTM0 initialization trigger and channel triggers, as programmed in the FTM external trigger register (EXTTRIG)
1001	FTM1 initialization trigger and channel triggers, as programmed in the FTM external trigger register (EXTTRIG)
1010	FTM2 initialization trigger and channel triggers, as programmed in the FTM external trigger register (EXTTRIG)
1011	FTM3 initialization trigger and channel triggers, as programmed in the FTM external trigger register (EXTTRIG)
1100	RTC Alarm
1101	RTC Seconds
1110	LPTMR Output
1111	Software Trigger

### 3.8.1.2 PDB Module Interconnections

PDB trigger outputs	Connection
Channel 0 triggers	ADC0 trigger
Channel 1 triggers	ADC1 trigger and synchronous input 1 of FTM0
DAC triggers	DAC0 and DAC1 trigger
Pulse-out	Pulse-out connected to each CMP module's sample/window input to control sample operation

### 3.8.1.3 Back-to-back acknowledgement connections

In this MCU, PDB back-to-back operation acknowledgment connections are implemented as follows:

- PDB channel 0 pre-trigger 0 acknowledgement input: ADC1SC1B\_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

So, the back-to-back chain is connected as a ring:

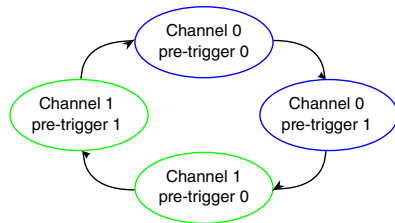


Figure 3-42. PDB back-to-back chain

The application code can set the PDBx\_CHnCI[BB] bits to configure the PDB pre-triggers as a single chain or several chains.

### 3.8.1.4 PDB Interval Trigger Connections to DAC

In this MCU, PDB interval trigger connections to DAC are implemented as follows.

- PDB interval trigger 0 connects to DAC0 hardware trigger input.
- PDB interval trigger 1 connects to DAC1 hardware trigger input.

### 3.8.1.5 DAC External Trigger Input Connections

In this MCU, the following DAC external trigger inputs are implemented.

- DAC external trigger input 1: ADC1SC1A\_COCO

#### NOTE

Application code can set the PDBx\_DACINTCn[EXT] bit to allow DAC external trigger input when the corresponding ADC Conversion complete flag, ADCx\_SC1n[COCO], is set.

### 3.8.1.6 Pulse-Out Connection

Individual PDB Pulse-Out signals are connected to each CMP block and used for sample window.

### 3.8.1.7 Pulse-Out Enable Register Implementation

The following table shows the comparison of pulse-out enable register at the module and chip level.

Table 3-57. PDB pulse-out enable register

Register	Module implementation	Chip implementation
POnEN	7:0 - POEN 31:8 - Reserved	0 - POEN[0] for CMP0 1 - POEN[1] for CMP1 2 - POEN[2] for CMP2 31:3 - Reserved

### 3.8.2 FlexTimer 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.

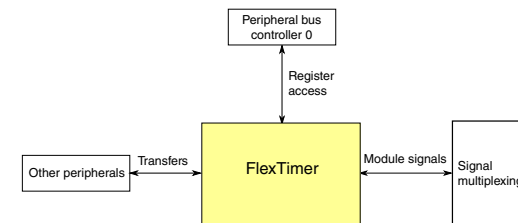


Figure 3-43. FlexTimer configuration

Table 3-58. Reference links to related information

Topic	Related module	Reference
Full description	FlexTimer	<a href="#">FlexTimer</a>
System memory map		<a href="#">System memory map</a>
Clocking		<a href="#">Clock distribution</a>
Power management		<a href="#">Power management</a>
Signal multiplexing	Port control	<a href="#">Signal multiplexing</a>

### 3.8.2.1 Instantiation Information

This device contains four FlexTimer modules.

The following table shows how these modules are configured.

**Table 3-59. FTM Instantiations**

FTM instance	Number of channels	Features/usage
FTM0	8	3-phase motor + 2 general purpose or stepper motor
FTM1	2 <sup>1</sup>	Quadrature decoder or general purpose
FTM2	2 <sup>1</sup>	Quadrature decoder or general purpose
FTM3	8	3-phase motor + 2 general purpose or stepper motor

1. Only channels 0 and 1 are available.

Compared with the FTM0 and FTM3 configuration, the FTM1 and FTM2 configuration adds the Quadrature decoder feature and reduces the number of channels.

### 3.8.2.2 External Clock Options

By default each FTM is clocked by the internal bus clock (the FTM refers to it as system clock). Each module contains a register setting that allows the module to be clocked from an external clock instead. There are two external FTM\_CLKINx pins that can be selected by any FTM module via the SIM\_SOPT4 register.

### 3.8.2.3 Fixed frequency clock

The fixed frequency clock for each FTM is MCGFFCLK.

### 3.8.2.4 FTM Interrupts

The FlexTimer has multiple sources of interrupt. However, these sources are OR'd together to generate a single interrupt request per FTM module to the interrupt controller. When an FTM interrupt occurs, read the FTM status registers (FMS, SC, and STATUS) to determine the exact interrupt source.

### 3.8.2.5 FTM Fault Detection Inputs

The following fault detection input options for the FTM modules are selected via the SIM\_SOPT4 register. The external pin option is selected by default.

- FTM0 FAULT0 = FTM0\_FLT0 pin or CMP0 output

### Timers

- FTM0 FAULT1 = FTM0\_FLT1 pin or CMP1 output
- FTM0 FAULT2 = FTM0\_FLT2 pin or CMP2 output
- FTM0 FAULT3 = FTM0\_FLT3 pin

- FTM1 FAULT0 = FTM1\_FLT0 pin or CMP0 output
- FTM1 FAULT1 = CMP1 output
- FTM1 FAULT2 = CMP2 output

- FTM2 FAULT0 = FTM2\_FLT0 pin or CMP0 output
- FTM2 FAULT1 = CMP1 output
- FTM2 FAULT2 = CMP2 output

- FTM3 FAULT0 = FTM3\_FLT0 pin or CMP0 output
- FTM3 FAULT1 = CMP2 output

### 3.8.2.6 FTM Hardware Triggers

The FTM synchronization hardware triggers are connected in the chip as follows:

- FTM0 hardware trigger 0 = CMP0 Output or FTM1 Match (when enabled in the FTM1 External Trigger (EXTTRIG) register)
- FTM0 hardware trigger 1 = PDB channel 1 Trigger Output or FTM2 Match (when enabled in the FTM2 External Trigger (EXTTRIG) register)
- FTM0 hardware trigger 2 = FTM0\_FLT0 pin
- FTM1 hardware trigger 0 = CMP0 Output
- FTM1 hardware trigger 1 = CMP1 Output
- FTM1 hardware trigger 2 = FTM1\_FLT0 pin

For the triggers with more than one option, SIM\_SOPT4 controls the selection.

### 3.8.2.7 Input capture options for FTM module instances

The following channel 0 input capture source options are selected via SIM\_SOPT4. The external pin option is selected by default.

- FTM1 channel 0 input capture = FTM1\_CH0 pin or CMP0 output or CMP1 output or USB start of frame pulse
- FTM2 channel 0 input capture = FTM2\_CH0 pin or CMP0 output or CMP1 output

#### NOTE

When the USB start of frame pulse option is selected as an FTM channel input capture, disable the USB SOF token

interrupt in the USB Interrupt Enable register (INTEN[SOFTOKEN]) to avoid USB enumeration conflicts.

### 3.8.2.8 FTM output triggers for other modules

FTM output triggers can be selected as input triggers for the PDB and ADC modules. See [PDB Instantiation](#) and [ADC triggers](#).

### 3.8.2.9 FTM Global Time Base

This chip provides the optional FTM global time base feature (see [Global time base \(GTB\)](#)).

FTM0 provides the only source for the FTM global time base. The other FTM modules can share the time base as shown in the following figure:

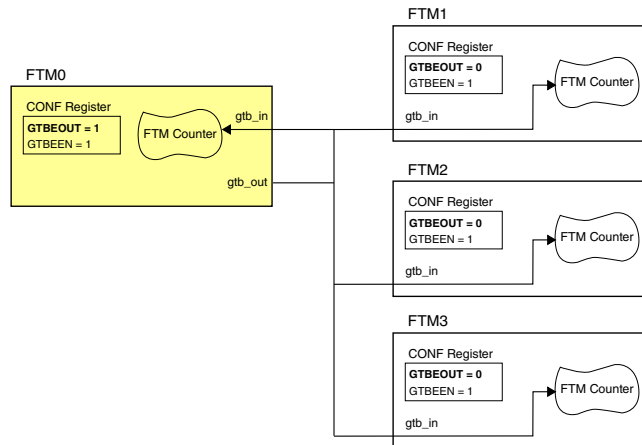


Figure 3-44. FTM Global Time Base Configuration

### 3.8.2.10 FTM BDM and debug halt mode

In the FTM chapter, references to the chip being in "BDM" are the same as the chip being in "debug halt mode".

### 3.8.2.11 FTM2 Base Addresses

FTM2 can be accessed through both AIPS0 and AIPS1. When accessed through AIPS0, the base address is 4003\_A000h and when accessed through AIPS1, the base address is 400B\_8000h.

### 3.8.2.12 FTM registers

FTM1 and FTM2 do not have the C2SC, C2V, C3SC, C3V, C4SC, C4V, C5SC, C5V, C6SC, C6V, C7SC, and C7V registers. FTM0 has these registers.

## 3.8.3 PIT 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.

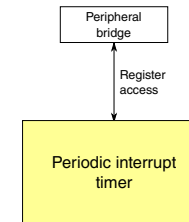


Figure 3-45. PIT configuration

Table 3-60. Reference links to related information

Topic	Related module	Reference
Full description	PIT	<a href="#">PIT</a>
System memory map		<a href="#">System memory map</a>
Clocking		<a href="#">Clock Distribution</a>
Power management		<a href="#">Power management</a>

### 3.8.3.1 PIT/DMA Periodic Trigger Assignments

The PIT generates periodic trigger events to the DMA Mux as shown in the table below.

Table 3-61. PIT channel assignments for periodic DMA triggering

DMA Channel Number	PIT Channel
DMA Channel 0	PIT Channel 0
DMA Channel 1	PIT Channel 1
DMA Channel 2	PIT Channel 2
DMA Channel 3	PIT Channel 3

### 3.8.3.2 PIT/ADC Triggers

PIT triggers are selected as ADCx trigger sources using the SIM\_SOPT7[ADCxTRGSEL] fields. For more details, refer to SIM chapter.

## 3.8.4 Low-power timer 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.

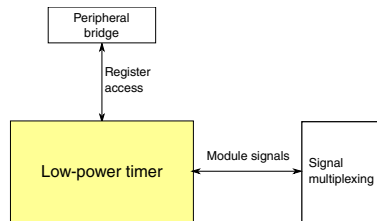


Figure 3-46. LPTMR configuration

Table 3-62. Reference links to related information

Topic	Related module	Reference
Full description	Low-power timer	<a href="#">Low-power timer</a>
System memory map		<a href="#">System memory map</a>
Clocking		<a href="#">Clock Distribution</a>
Power management		<a href="#">Power management</a>
Signal Multiplexing	Port control	<a href="#">Signal Multiplexing</a>

### 3.8.4.1 LPTMR prescaler/glitch filter clocking options

The prescaler and glitch filter of the LPTMR module can be clocked from one of four sources determined by the LPTMR0\_PSR[PCS] bitfield. The following table shows the chip-specific clock assignments for this bitfield.

#### NOTE

The chosen clock must remain enabled if the LPTMR is to continue operating in all required low-power modes.

LPTMR0_PSR[PCS]	Prescaler/glitch filter clock number	Chip clock
00	0	MCGIRCLK — internal reference clock (not available in VLPS/LLS/VLLS modes)
01	1	LPO — 1 kHz clock (not available in VLLS0 mode)
10	2	ERCLK32K — secondary external reference clock
11	3	OSCERCLK — external reference clock (not available in VLLS0 mode)

See [Clock Distribution](#) for more details on these clocks.

### 3.8.4.2 LPTMR pulse counter input options

The LPTMR\_CSR[TPS] bitfield configures the input source used in pulse counter mode. The following table shows the chip-specific input assignments for this bitfield.

LPTMR_CSR[TPS]	Pulse counter input number	Chip input
00	0	CMP0 output
01	1	LPTMR_ALT1 pin
10	2	LPTMR_ALT2 pin
11	3	Reserved

## 3.8.5 CMT 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.



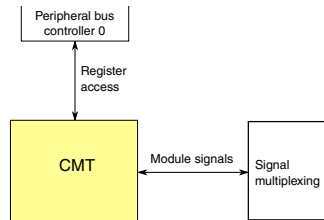


Figure 3-47. CMT configuration

Table 3-63. Reference links to related information

Topic	Related module	Reference
Full description	Carrier modulator transmitter (CMT)	<a href="#">CMT</a>
System memory map		<a href="#">System memory map</a>
Clocking		<a href="#">Clock distribution</a>
Power management		<a href="#">Power management</a>
Signal multiplexing	Port control	<a href="#">Signal multiplexing</a>

### 3.8.5.1 Instantiation Information

This device contains one CMT module.

### 3.8.5.2 IRO Drive Strength

The IRO pad requires higher current drive than can be obtained from a single pad. For this device, the pin associated with the CMT\_IRO signal is doubled bonded to two pads.

SIM\_SOPT2[PTD7PAD] can be used to configure the pin associated with the CMT\_IRO signal as a higher current output port pin.

## 3.8.6 RTC 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.

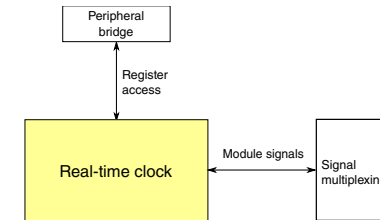


Figure 3-48. RTC configuration

Table 3-64. Reference links to related information

Topic	Related module	Reference
Full description	RTC	<a href="#">RTC</a>
System memory map		<a href="#">System memory map</a>
Clocking		<a href="#">Clock Distribution</a>
Power management		<a href="#">Power management</a>

### 3.8.6.1 RTC\_CLKOUT signal

When the RTC is enabled and the port control module selects the RTC\_CLKOUT function, the RTC\_CLKOUT signal outputs a 1 Hz or 32 kHz output derived from RTC oscillator as shown below.

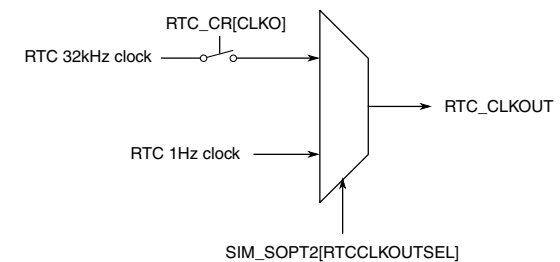


Figure 3-49. RTC\_CLKOUT generation

## 3.9 Communication interfaces

## Chapter 21 Direct Memory Access Multiplexer (DMAMUX)

### 21.1 Introduction

#### NOTE

For the chip-specific implementation details of this module's instances, see the chip configuration information.

#### 21.1.1 Overview

The Direct Memory Access Multiplexer (DMAMUX) routes DMA sources, called slots, to any of the 16 DMA channels. This process is illustrated in the following figure.

#### Introduction

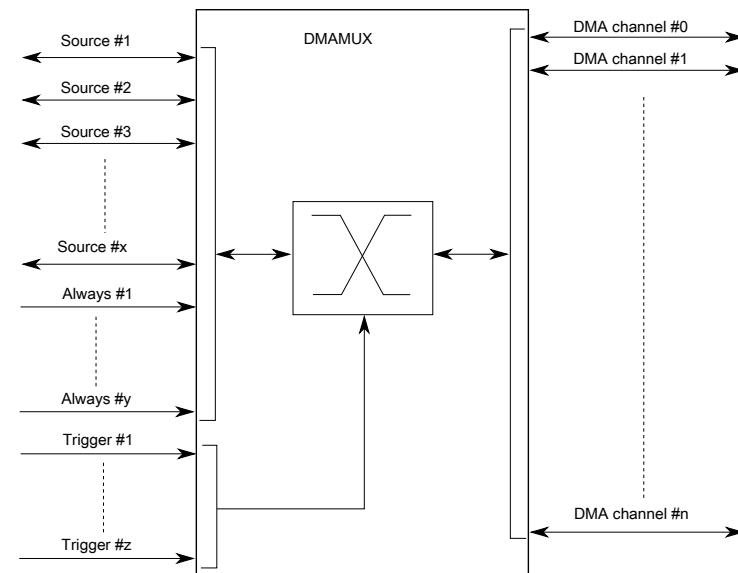


Figure 21-1. DMAMUX block diagram

#### 21.1.2 Features

The DMAMUX module provides these features:

- Up to 52 peripheral slots and up to 10 always-on slots can be routed to 16 channels.
- 16 independently selectable DMA channel routers.
  - The first four channels additionally provide a trigger functionality.
- Each channel router can be assigned to one of the possible peripheral DMA slots or to one of the always-on slots.

#### 21.1.3 Modes of operation

The following operating modes are available:

- Disabled mode

In this mode, the DMA channel is disabled. Because disabling and enabling of DMA channels is done primarily via the DMA configuration registers, this mode is used mainly as the reset state for a DMA channel in the DMA channel MUX. It may also be used to temporarily suspend a DMA channel while reconfiguration of the system takes place, for example, changing the period of a DMA trigger.

- Normal mode

In this mode, a DMA source is routed directly to the specified DMA channel. The operation of the DMAMUX in this mode is completely transparent to the system.

- Periodic Trigger mode

In this mode, a DMA source may only request a DMA transfer, such as when a transmit buffer becomes empty or a receive buffer becomes full, periodically.

Configuration of the period is done in the registers of the periodic interrupt timer (PIT). This mode is available only for channels 0–3.

## 21.2 External signal description

The DMAMUX has no external pins.

## 21.3 Memory map/register definition

This section provides a detailed description of all memory-mapped registers in the DMAMUX.

**DMAMUX memory map**

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4002_1000	Channel Configuration register (DMAMUX_CHCFG0)	8	R/W	00h	<a href="#">21.3.1/466</a>
4002_1001	Channel Configuration register (DMAMUX_CHCFG1)	8	R/W	00h	<a href="#">21.3.1/466</a>
4002_1002	Channel Configuration register (DMAMUX_CHCFG2)	8	R/W	00h	<a href="#">21.3.1/466</a>
4002_1003	Channel Configuration register (DMAMUX_CHCFG3)	8	R/W	00h	<a href="#">21.3.1/466</a>
4002_1004	Channel Configuration register (DMAMUX_CHCFG4)	8	R/W	00h	<a href="#">21.3.1/466</a>
4002_1005	Channel Configuration register (DMAMUX_CHCFG5)	8	R/W	00h	<a href="#">21.3.1/466</a>
4002_1006	Channel Configuration register (DMAMUX_CHCFG6)	8	R/W	00h	<a href="#">21.3.1/466</a>
4002_1007	Channel Configuration register (DMAMUX_CHCFG7)	8	R/W	00h	<a href="#">21.3.1/466</a>
4002_1008	Channel Configuration register (DMAMUX_CHCFG8)	8	R/W	00h	<a href="#">21.3.1/466</a>

*Table continues on the next page...*

**DMAMUX memory map (continued)**

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4002_1009	Channel Configuration register (DMAMUX_CHCFG9)	8	R/W	00h	<a href="#">21.3.1/466</a>
4002_100A	Channel Configuration register (DMAMUX_CHCFG10)	8	R/W	00h	<a href="#">21.3.1/466</a>
4002_100B	Channel Configuration register (DMAMUX_CHCFG11)	8	R/W	00h	<a href="#">21.3.1/466</a>
4002_100C	Channel Configuration register (DMAMUX_CHCFG12)	8	R/W	00h	<a href="#">21.3.1/466</a>
4002_100D	Channel Configuration register (DMAMUX_CHCFG13)	8	R/W	00h	<a href="#">21.3.1/466</a>
4002_100E	Channel Configuration register (DMAMUX_CHCFG14)	8	R/W	00h	<a href="#">21.3.1/466</a>
4002_100F	Channel Configuration register (DMAMUX_CHCFG15)	8	R/W	00h	<a href="#">21.3.1/466</a>

### 21.3.1 Channel Configuration register (DMAMUX\_CHCFGn)

Each of the DMA channels can be independently enabled/disabled and associated with one of the DMA slots (peripheral slots or always-on slots) in the system.

#### NOTE

Setting multiple CHCFG registers with the same source value will result in unpredictable behavior. This is true, even if a channel is disabled (ENBL==0).

Before changing the trigger or source settings, a DMA channel must be disabled via CHCFGn[ENBL].

Address: 4002\_1000h base + 0h offset + (1d × i), where i=0d to 15d

Bit	7	6	5	4	3	2	1	0
Read	ENBL	TRIG	SOURCE					
Write								
Reset	0	0	0	0	0	0	0	0

**DMAMUX\_CHCFGn field descriptions**

Field	Description
7 ENBL	DMA Channel Enable  Enables the DMA channel.  0 DMA channel is disabled. This mode is primarily used during configuration of the DMAMux. The DMA has separate channel enables/disables, which should be used to disable or reconfigure a DMA channel. 1 DMA channel is enabled
6 TRIG	DMA Channel Trigger Enable  Enables the periodic trigger capability for the triggered DMA channel.

*Table continues on the next page...*

## DMAMUX\_CHCFGn field descriptions (continued)

Field	Description
	0 Triggering is disabled. If triggering is disabled and ENBL is set, the DMA Channel will simply route the specified source to the DMA channel. (Normal mode)
	1 Triggering is enabled. If triggering is enabled and ENBL is set, the DMAMUX is in Periodic Trigger mode.
SOURCE	DMA Channel Source (Slot)  Specifies which DMA source, if any, is routed to a particular DMA channel. See the chip-specific DMAMUX information for details about the peripherals and their slot numbers.

## 21.4 Functional description

The primary purpose of the DMAMUX is to provide flexibility in the system's use of the available DMA channels.

As such, configuration of the DMAMUX is intended to be a static procedure done during execution of the system boot code. However, if the procedure outlined in [Enabling and configuring sources](#) is followed, the configuration of the DMAMUX may be changed during the normal operation of the system.

Functionally, the DMAMUX channels may be divided into two classes:

- Channels that implement the normal routing functionality plus periodic triggering capability
- Channels that implement only the normal routing functionality

### 21.4.1 DMA channels with periodic triggering capability

Besides the normal routing functionality, the first 4 channels of the DMAMUX provide a special periodic triggering capability that can be used to provide an automatic mechanism to transmit bytes, frames, or packets at fixed intervals without the need for processor intervention.

The trigger is generated by the periodic interrupt timer (PIT); as such, the configuration of the periodic triggering interval is done via configuration registers in the PIT. See the section on periodic interrupt timer for more information on this topic.

## Note

Because of the dynamic nature of the system (due to DMA channel priorities, bus arbitration, interrupt service routine lengths, etc.), the number of clock cycles between a trigger and the actual DMA transfer cannot be guaranteed.

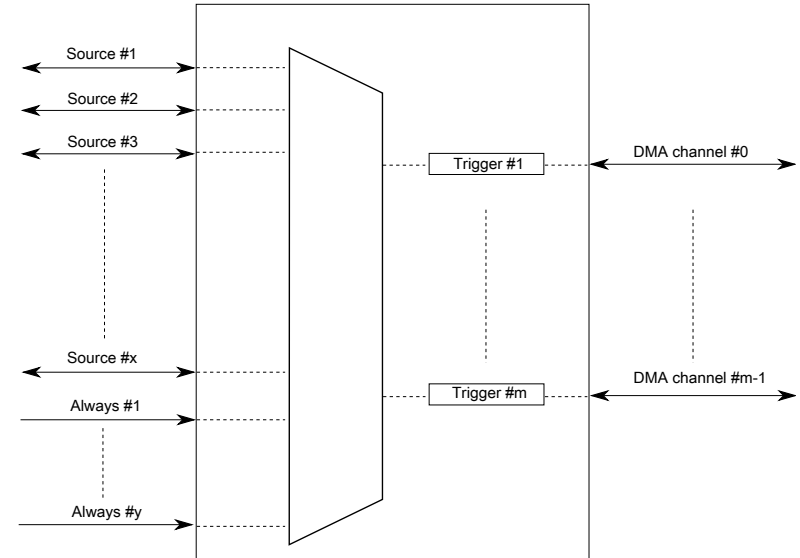


Figure 21-2. DMAMUX triggered channels

The DMA channel triggering capability allows the system to schedule regular DMA transfers, usually on the transmit side of certain peripherals, without the intervention of the processor. This trigger works by gating the request from the peripheral to the DMA until a trigger event has been seen. This is illustrated in the following figure.

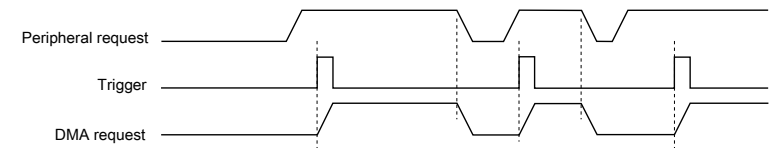
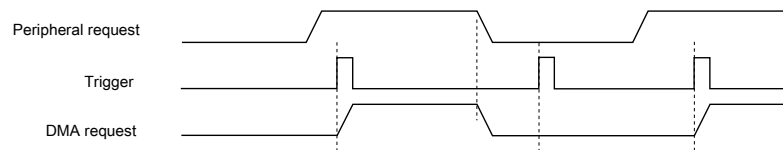


Figure 21-3. DMAMUX channel triggering: normal operation

After the DMA request has been serviced, the peripheral will negate its request, effectively resetting the gating mechanism until the peripheral reasserts its request and the next trigger event is seen. This means that if a trigger is seen, but the peripheral is not requesting a transfer, then that trigger will be ignored. This situation is illustrated in the following figure.



**Figure 21-4. DMAMUX channel triggering: ignored trigger**

This triggering capability may be used with any peripheral that supports DMA transfers, and is most useful for two types of situations:

- Periodically polling external devices on a particular bus

As an example, the transmit side of an SPI is assigned to a DMA channel with a trigger, as described above. After it has been set up, the SPI will request DMA transfers, presumably from memory, as long as its transmit buffer is empty. By using a trigger on this channel, the SPI transfers can be automatically performed every 5  $\mu$ s (as an example). On the receive side of the SPI, the SPI and DMA can be configured to transfer receive data into memory, effectively implementing a method to periodically read data from external devices and transfer the results into memory without processor intervention.

- Using the GPIO ports to drive or sample waveforms

By configuring the DMA to transfer data to one or more GPIO ports, it is possible to create complex waveforms using tabular data stored in on-chip memory. Conversely, using the DMA to periodically transfer data from one or more GPIO ports, it is possible to sample complex waveforms and store the results in tabular form in on-chip memory.

A more detailed description of the capability of each trigger, including resolution, range of values, and so on, may be found in the periodic interrupt timer section.

## 21.4.2 DMA channels with no triggering capability

The other channels of the DMAMUX provide the normal routing functionality as described in [Modes of operation](#).

## 21.4.3 Always-enabled DMA sources

In addition to the peripherals that can be used as DMA sources, there are 10 additional DMA sources that are always enabled. Unlike the peripheral DMA sources, where the peripheral controls the flow of data during DMA transfers, the sources that are always enabled provide no such "throttling" of the data transfers. These sources are most useful in the following cases:

- Performing DMA transfers to/from GPIO—Moving data from/to one or more GPIO pins, either unthrottled (that is, as fast as possible), or periodically (using the DMA triggering capability).
- Performing DMA transfers from memory to memory—Moving data from memory to memory, typically as fast as possible, sometimes with software activation.
- Performing DMA transfers from memory to the external bus, or vice-versa—Similar to memory to memory transfers, this is typically done as quickly as possible.
- Any DMA transfer that requires software activation—Any DMA transfer that should be explicitly started by software.

In cases where software should initiate the start of a DMA transfer, an always-enabled DMA source can be used to provide maximum flexibility. When activating a DMA channel via software, subsequent executions of the minor loop require that a new start event be sent. This can either be a new software activation, or a transfer request from the DMA channel MUX. The options for doing this are:

- Transfer all data in a single minor loop.

By configuring the DMA to transfer all of the data in a single minor loop (that is, major loop counter = 1), no reactivation of the channel is necessary. The disadvantage to this option is the reduced granularity in determining the load that the DMA transfer will impose on the system. For this option, the DMA channel must be disabled in the DMA channel MUX.

- Use explicit software reactivation.

In this option, the DMA is configured to transfer the data using both minor and major loops, but the processor is required to reactivate the channel by writing to the DMA registers *after every minor loop*. For this option, the DMA channel must be disabled in the DMA channel MUX.

- Use an always-enabled DMA source.

In this option, the DMA is configured to transfer the data using both minor and major loops, and the DMA channel MUX does the channel reactivation. For this option, the DMA channel should be enabled and pointing to an "always enabled" source. Note that the reactivation of the channel can be continuous (DMA triggering is disabled) or can use the DMA triggering capability. In this manner, it is possible to execute periodic transfers of packets of data from one source to another, without processor intervention.

## 21.5 Initialization/application information

This section provides instructions for initializing the DMA channel MUX.

### 21.5.1 Reset

The reset state of each individual bit is shown in [Memory map/register definition](#). In summary, after reset, all channels are disabled and must be explicitly enabled before use.

### 21.5.2 Enabling and configuring sources

To enable a source with periodic triggering:

1. Determine with which DMA channel the source will be associated. Note that only the first 4 DMA channels have periodic triggering capability.
2. Clear the CHCFG[ENBL] and CHCFG[TRIG] fields of the DMA channel.
3. Ensure that the DMA channel is properly configured in the DMA. The DMA channel may be enabled at this point.
4. Configure the corresponding timer.
5. Select the source to be routed to the DMA channel. Write to the corresponding CHCFG register, ensuring that the CHCFG[ENBL] and CHCFG[TRIG] fields are set.

#### NOTE

The following is an example. See the chip configuration details for the number of this device's DMA channels that have triggering capability.

To configure source #5 transmit for use with DMA channel 1, with periodic triggering capability:

1. Write 0x00 to CHCFG1.

#### Initialization/application information

2. Configure channel 1 in the DMA, including enabling the channel.
3. Configure a timer for the desired trigger interval.
4. Write 0xC5 to CHCFG1.

The following code example illustrates steps 1 and 4 above:

```
void DMAMUX_Init(uint8_t DMA_CH, uint8_t DMAMUX_SOURCE)
{
    DMAMUX_0.CHCFG[DMA_CH].B.SOURCE = DMAMUX_SOURCE;
    DMAMUX_0.CHCFG[DMA_CH].B.ENBL   = 1;
    DMAMUX_0.CHCFG[DMA_CH].B.TRIG   = 1;
}
```

To enable a source, without periodic triggering:

1. Determine with which DMA channel the source will be associated. Note that only the first 4 DMA channels have periodic triggering capability.
2. Clear the CHCFG[ENBL] and CHCFG[TRIG] fields of the DMA channel.
3. Ensure that the DMA channel is properly configured in the DMA. The DMA channel may be enabled at this point.
4. Select the source to be routed to the DMA channel. Write to the corresponding CHCFG register, ensuring that CHCFG[ENBL] is set while CHCFG[TRIG] is cleared.

#### NOTE

The following is an example. See the chip configuration details for the number of this device's DMA channels that have triggering capability.

To configure source #5 transmit for use with DMA channel 1, with no periodic triggering capability:

1. Write 0x00 to CHCFG1.
2. Configure channel 1 in the DMA, including enabling the channel.
3. Write 0x85 to CHCFG1.

The following code example illustrates steps 1 and 3 above:

```
In File registers.h:
#define DMAMUX_BASE_ADDR 0x40021000/* Example only ! */
/* Following example assumes char is 8-bits */
volatile unsigned char *CHCFG0 = (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x0000);
volatile unsigned char *CHCFG1 = (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x0001);
volatile unsigned char *CHCFG2 = (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x0002);
volatile unsigned char *CHCFG3 = (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x0003);
volatile unsigned char *CHCFG4 = (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x0004);
volatile unsigned char *CHCFG5 = (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x0005);
volatile unsigned char *CHCFG6 = (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x0006);
volatile unsigned char *CHCFG7 = (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x0007);
volatile unsigned char *CHCFG8 = (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x0008);
volatile unsigned char *CHCFG9 = (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x0009);
volatile unsigned char *CHCFG10 = (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x000A);
volatile unsigned char *CHCFG11 = (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x000B);
volatile unsigned char *CHCFG12 = (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x000C);
volatile unsigned char *CHCFG13 = (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x000D);
```

## Chapter 21 Direct Memory Access Multiplexer (DMAMUX)

```
volatile unsigned char *CHCFG14= (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x000E);
volatile unsigned char *CHCFG15= (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x000F);
```

```
In File main.c:
#include "registers.h"
:
:
*CHCFG1 = 0x00;
*CHCFG1 = 0x85;
```

To disable a source:

A particular DMA source may be disabled by not writing the corresponding source value into any of the CHCFG registers. Additionally, some module-specific configuration may be necessary. See the appropriate section for more details.

To switch the source of a DMA channel:

1. Disable the DMA channel in the DMA and reconfigure the channel for the new source.
2. Clear the CHCFG[ENBL] and CHCFG[TRIG] bits of the DMA channel.
3. Select the source to be routed to the DMA channel. Write to the corresponding CHCFG register, ensuring that the CHCFG[ENBL] and CHCFG[TRIG] fields are set.

To switch DMA channel 8 from source #5 transmit to source #7 transmit:

1. In the DMA configuration registers, disable DMA channel 8 and reconfigure it to handle the transfers to peripheral slot 7. This example assumes channel 8 doesn't have triggering capability.
2. Write 0x00 to CHCFG8.
3. Write 0x87 to CHCFG8. (In this example, setting CHCFG[TRIG] would have no effect due to the assumption that channel 8 does not support the periodic triggering functionality.)

The following code example illustrates steps 2 and 3 above:

```
In File registers.h:
#define DMAMUX_BASE_ADDR    0x40021000/* Example only ! */
/* Following example assumes char is 8-bits */
volatile unsigned char *CHCFG0 = (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x0000);
volatile unsigned char *CHCFG1 = (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x0001);
volatile unsigned char *CHCFG2 = (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x0002);
volatile unsigned char *CHCFG3 = (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x0003);
volatile unsigned char *CHCFG4 = (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x0004);
volatile unsigned char *CHCFG5 = (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x0005);
volatile unsigned char *CHCFG6 = (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x0006);
volatile unsigned char *CHCFG7 = (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x0007);
volatile unsigned char *CHCFG8 = (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x0008);
volatile unsigned char *CHCFG9 = (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x0009);
volatile unsigned char *CHCFG10= (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x000A);
volatile unsigned char *CHCFG11= (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x000B);
volatile unsigned char *CHCFG12= (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x000C);
volatile unsigned char *CHCFG13= (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x000D);
volatile unsigned char *CHCFG14= (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x000E);
volatile unsigned char *CHCFG15= (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x000F);
```

## Initialization/application information

```
In File main.c:
#include "registers.h"
:
:
*CHCFG8 = 0x00;
*CHCFG8 = 0x87;
```

### 34.4.2 Core engine / control logic

This block contains RNGA's control logic as well as its core engine used to generate random data.

#### 34.4.2.1 Control logic

The control logic contains the address decoder, all addressable registers, and control state machines for RNGA. This block is responsible for communication with both the peripheral interface and the Output (OR) register interface. The block also controls the core engine to generate random data. The general functionality of the block is as follows:

After reset, RNGA operates in Normal mode as follows:

1. The core engine generates entropy and stores it in the shift registers.
2. After you enable random-data generation by loading CR[GO], every 256 clock cycles the core engine generates a new random-data word. If SR[OREG\_LVL] = 0, then the control block loads the new random data into OR and set SR[OREG\_LVL] = 1; else the new data is discarded.

#### 34.4.2.2 Core engine

The core engine block contains the logic used to generate random data. The logic within the core engine contains the internal shift registers as well as the logic used to generate the two oscillator-based clocks. The control logic determines how the shift registers are configured as well as when the oscillator clocks are turned on.

## 34.5 Initialization/application information

The intended general operation of RNGA is as follows:

1. Reset/initialize.
2. Write 1 to CR[INTM], CR[HA], and CR[GO].
3. Poll SR[OREG\_LVL] until it is not 0.
4. When SR[OREG\_LVL] is not 0, read the available random data from OR[RANDOUT].
5. Repeat steps 3 and 4 as needed.

For application information, see [Overview](#).



## Chapter 35

# Analog-to-Digital Converter (ADC)

### 35.1 Introduction

#### NOTE

For the chip-specific implementation details of this module's instances, see the chip configuration information.

The 16-bit analog-to-digital converter (ADC) is a successive approximation ADC designed for operation within an integrated microcontroller system-on-chip.

#### NOTE

For the chip specific modes of operation, see the power management information of the device.

#### 35.1.1 Features

Following are the features of the ADC module.

- Linear successive approximation algorithm with up to 16-bit resolution
- Up to four pairs of differential and 24 single-ended external analog inputs
- Output modes:
  - differential 16-bit, 13-bit, 11-bit, and 9-bit modes
  - single-ended 16-bit, 12-bit, 10-bit, and 8-bit modes
- Output format in 2's complement 16-bit sign extended for differential modes
- Output in right-justified unsigned format for single-ended
- Single or continuous conversion, that is, automatic return to idle after single conversion

#### Introduction

- Configurable sample time and conversion speed/power
- Conversion complete/hardware average complete flag and interrupt
- Input clock selectable from up to four sources
- Operation in low-power modes for lower noise
- Asynchronous clock source for lower noise operation with option to output the clock
- Selectable hardware conversion trigger with hardware channel select
- Automatic compare with interrupt for less-than, greater-than or equal-to, within range, or out-of-range, programmable value
- Temperature sensor
- Hardware average function
- Selectable voltage reference: external or alternate
- Self-Calibration mode

#### 35.1.2 Block diagram

The following figure is the ADC module block diagram.

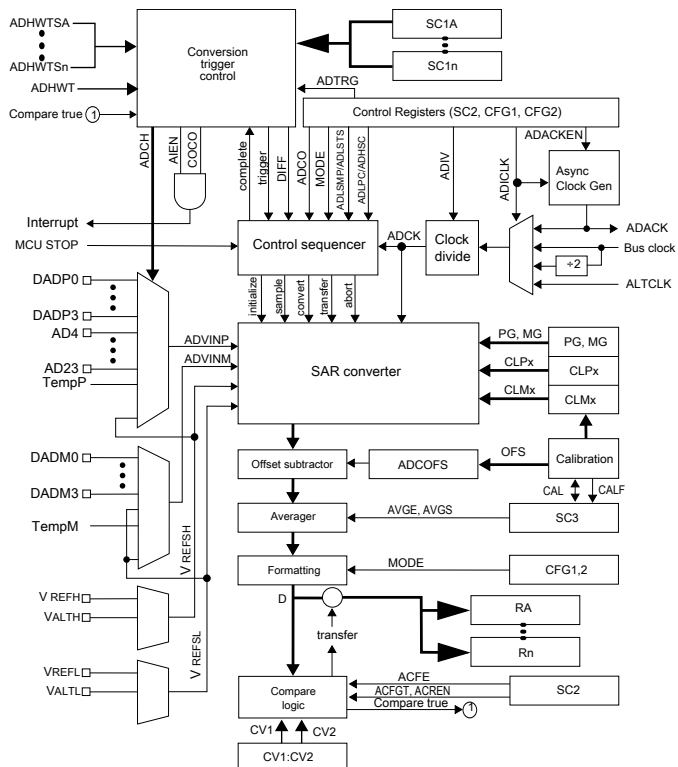


Figure 35-1. ADC block diagram

## 35.2 ADC signal descriptions

The ADC module supports up to 4 pairs of differential inputs and up to 24 single-ended inputs.

Each differential pair requires two inputs, DADPx and DADMx. The ADC also requires four supply/reference/ground connections.

### ADC signal descriptions

#### NOTE

For the number of channels supported on this device as well as information regarding other chip-specific inputs into the ADC block, see the chip-specific ADC configuration information.

Table 35-1. ADC signal descriptions

Signal	Description	I/O
DADP3–DADP0	Differential Analog Channel Inputs	I
DADM3–DADM0	Differential Analog Channel Inputs	I
AD <sub>n</sub>	Single-Ended Analog Channel Inputs	I
V <sub>REFSH</sub>	Voltage Reference Select High	I
V <sub>REFSL</sub>	Voltage Reference Select Low	I
V <sub>DDA</sub>	Analog Power Supply	I
V <sub>SSA</sub>	Analog Ground	I

### 35.2.1 Analog Power (V<sub>DDA</sub>)

The ADC analog portion uses V<sub>DDA</sub> as its power connection. In some packages, V<sub>DDA</sub> is connected internally to V<sub>DD</sub>. If externally available, connect the V<sub>DDA</sub> pin to the same voltage potential as V<sub>DD</sub>. External filtering may be necessary to ensure clean V<sub>DDA</sub> for good results.

### 35.2.2 Analog Ground (V<sub>SSA</sub>)

The ADC analog portion uses V<sub>SSA</sub> as its ground connection. In some packages, V<sub>SSA</sub> is connected internally to V<sub>SS</sub>. If externally available, connect the V<sub>SSA</sub> pin to the same voltage potential as V<sub>SS</sub>.

### 35.2.3 Voltage Reference Select

V<sub>REFSH</sub> and V<sub>REFSL</sub> are the high and low reference voltages for the ADC module.

The ADC can be configured to accept one of two voltage reference pairs for V<sub>REFSH</sub> and V<sub>REFSL</sub>. Each pair contains a positive reference that must be between the minimum Ref Voltage High and V<sub>DDA</sub>, and a ground reference that must be at the same potential as V<sub>SSA</sub>. The two pairs are external (V<sub>REFH</sub> and V<sub>REFL</sub>) and alternate (V<sub>ALTH</sub> and V<sub>ALT</sub>). These voltage references are selected using SC2[REFSEL]. The alternate V<sub>ALTH</sub> and

$V_{ALTL}$  voltage reference pair may select additional external pins or internal sources depending on MCU configuration. See the chip configuration information on the Voltage References specific to this MCU.

In some packages,  $V_{REFH}$  is connected in the package to  $V_{DDA}$  and  $V_{REFL}$  to  $V_{SSA}$ . If externally available, the positive reference(s) may be connected to the same potential as  $V_{DDA}$  or may be driven by an external source to a level between the minimum Ref Voltage High and the  $V_{DDA}$  potential.  $V_{REFH}$  must never exceed  $V_{DDA}$ . Connect the ground references to the same voltage potential as  $V_{SSA}$ .

### 35.2.4 Analog Channel Inputs (ADx)

The ADC module supports up to 24 single-ended analog inputs. A single-ended input is selected for conversion through the  $SC1[ADCH]$  channel select bits when  $SC1n[DIFF]$  is low.

### 35.2.5 Differential Analog Channel Inputs (DADx)

The ADC module supports up to four differential analog channel inputs. Each differential analog input is a pair of external pins,  $DADPx$  and  $DADMx$ , referenced to each other to provide the most accurate analog to digital readings. A differential input is selected for conversion through  $SC1[ADCH]$  when  $SC1n[DIFF]$  is high. All  $DADPx$  inputs may be used as single-ended inputs if  $SC1n[DIFF]$  is low. In certain MCU configurations, some  $DADMx$  inputs may also be used as single-ended inputs if  $SC1n[DIFF]$  is low. For ADC connections specific to this device, see the chip-specific ADC information.

## 35.3 Memory map and register definitions

This section describes the ADC registers.

ADC memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/page
4003_B000	ADC Status and Control Registers 1 (ADC0_SC1A)	32	R/W	0000_001Fh	<a href="#">35.3.1/853</a>
4003_B004	ADC Status and Control Registers 1 (ADC0_SC1B)	32	R/W	0000_001Fh	<a href="#">35.3.1/853</a>
4003_B008	ADC Configuration Register 1 (ADC0_CFG1)	32	R/W	0000_0000h	<a href="#">35.3.2/857</a>
4003_B00C	ADC Configuration Register 2 (ADC0_CFG2)	32	R/W	0000_0000h	<a href="#">35.3.3/858</a>
4003_B010	ADC Data Result Register (ADC0_RA)	32	R	0000_0000h	<a href="#">35.3.4/859</a>

Table continues on the next page...

ADC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/page
4003_B014	ADC Data Result Register (ADC0_RB)	32	R	0000_0000h	<a href="#">35.3.4/859</a>
4003_B018	Compare Value Registers (ADC0_CV1)	32	R/W	0000_0000h	<a href="#">35.3.5/861</a>
4003_B01C	Compare Value Registers (ADC0_CV2)	32	R/W	0000_0000h	<a href="#">35.3.5/861</a>
4003_B020	Status and Control Register 2 (ADC0_SC2)	32	R/W	0000_0000h	<a href="#">35.3.6/862</a>
4003_B024	Status and Control Register 3 (ADC0_SC3)	32	R/W	0000_0000h	<a href="#">35.3.7/864</a>
4003_B028	ADC Offset Correction Register (ADC0_OFS)	32	R/W	0000_0004h	<a href="#">35.3.8/865</a>
4003_B02C	ADC Plus-Side Gain Register (ADC0_PG)	32	R/W	0000_8200h	<a href="#">35.3.9/866</a>
4003_B030	ADC Minus-Side Gain Register (ADC0_MG)	32	R/W	0000_8200h	<a href="#">35.3.10/866</a>
4003_B034	ADC Plus-Side General Calibration Value Register (ADC0_CLPD)	32	R/W	0000_000Ah	<a href="#">35.3.11/867</a>
4003_B038	ADC Plus-Side General Calibration Value Register (ADC0_CLPS)	32	R/W	0000_0020h	<a href="#">35.3.12/868</a>
4003_B03C	ADC Plus-Side General Calibration Value Register (ADC0_CLP4)	32	R/W	0000_0200h	<a href="#">35.3.13/868</a>
4003_B040	ADC Plus-Side General Calibration Value Register (ADC0_CLP3)	32	R/W	0000_0100h	<a href="#">35.3.14/869</a>
4003_B044	ADC Plus-Side General Calibration Value Register (ADC0_CLP2)	32	R/W	0000_0080h	<a href="#">35.3.15/869</a>
4003_B048	ADC Plus-Side General Calibration Value Register (ADC0_CLP1)	32	R/W	0000_0040h	<a href="#">35.3.16/870</a>
4003_B04C	ADC Plus-Side General Calibration Value Register (ADC0_CLP0)	32	R/W	0000_0020h	<a href="#">35.3.17/870</a>
4003_B054	ADC Minus-Side General Calibration Value Register (ADC0_CLMD)	32	R/W	0000_000Ah	<a href="#">35.3.18/871</a>
4003_B058	ADC Minus-Side General Calibration Value Register (ADC0_CLMS)	32	R/W	0000_0020h	<a href="#">35.3.19/871</a>
4003_B05C	ADC Minus-Side General Calibration Value Register (ADC0_CLM4)	32	R/W	0000_0200h	<a href="#">35.3.20/872</a>
4003_B060	ADC Minus-Side General Calibration Value Register (ADC0_CLM3)	32	R/W	0000_0100h	<a href="#">35.3.21/872</a>
4003_B064	ADC Minus-Side General Calibration Value Register (ADC0_CLM2)	32	R/W	0000_0080h	<a href="#">35.3.22/873</a>
4003_B068	ADC Minus-Side General Calibration Value Register (ADC0_CLM1)	32	R/W	0000_0040h	<a href="#">35.3.23/873</a>
4003_B06C	ADC Minus-Side General Calibration Value Register (ADC0_CLM0)	32	R/W	0000_0020h	<a href="#">35.3.24/874</a>
400B_B000	ADC Status and Control Registers 1 (ADC1_SC1A)	32	R/W	0000_001Fh	<a href="#">35.3.1/853</a>
400B_B004	ADC Status and Control Registers 1 (ADC1_SC1B)	32	R/W	0000_001Fh	<a href="#">35.3.1/853</a>
400B_B008	ADC Configuration Register 1 (ADC1_CFG1)	32	R/W	0000_0000h	<a href="#">35.3.2/857</a>
400B_B00C	ADC Configuration Register 2 (ADC1_CFG2)	32	R/W	0000_0000h	<a href="#">35.3.3/858</a>
400B_B010	ADC Data Result Register (ADC1_RA)	32	R	0000_0000h	<a href="#">35.3.4/859</a>

Table continues on the next page...

## ADC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
400B_B014	ADC Data Result Register (ADC1_RB)	32	R	0000_0000h	<a href="#">35.3.4/859</a>
400B_B018	Compare Value Registers (ADC1_CV1)	32	R/W	0000_0000h	<a href="#">35.3.5/861</a>
400B_B01C	Compare Value Registers (ADC1_CV2)	32	R/W	0000_0000h	<a href="#">35.3.5/861</a>
400B_B020	Status and Control Register 2 (ADC1_SC2)	32	R/W	0000_0000h	<a href="#">35.3.6/862</a>
400B_B024	Status and Control Register 3 (ADC1_SC3)	32	R/W	0000_0000h	<a href="#">35.3.7/864</a>
400B_B028	ADC Offset Correction Register (ADC1_OFS)	32	R/W	0000_0004h	<a href="#">35.3.8/865</a>
400B_B02C	ADC Plus-Side Gain Register (ADC1_PG)	32	R/W	0000_8200h	<a href="#">35.3.9/866</a>
400B_B030	ADC Minus-Side Gain Register (ADC1_MG)	32	R/W	0000_8200h	<a href="#">35.3.10/866</a>
400B_B034	ADC Plus-Side General Calibration Value Register (ADC1_CLPD)	32	R/W	0000_000Ah	<a href="#">35.3.11/867</a>
400B_B038	ADC Plus-Side General Calibration Value Register (ADC1_CLPS)	32	R/W	0000_0020h	<a href="#">35.3.12/868</a>
400B_B03C	ADC Plus-Side General Calibration Value Register (ADC1_CLP4)	32	R/W	0000_0200h	<a href="#">35.3.13/868</a>
400B_B040	ADC Plus-Side General Calibration Value Register (ADC1_CLP3)	32	R/W	0000_0100h	<a href="#">35.3.14/869</a>
400B_B044	ADC Plus-Side General Calibration Value Register (ADC1_CLP2)	32	R/W	0000_0080h	<a href="#">35.3.15/869</a>
400B_B048	ADC Plus-Side General Calibration Value Register (ADC1_CLP1)	32	R/W	0000_0040h	<a href="#">35.3.16/870</a>
400B_B04C	ADC Plus-Side General Calibration Value Register (ADC1_CLP0)	32	R/W	0000_0020h	<a href="#">35.3.17/870</a>
400B_B054	ADC Minus-Side General Calibration Value Register (ADC1_CLMD)	32	R/W	0000_000Ah	<a href="#">35.3.18/871</a>
400B_B058	ADC Minus-Side General Calibration Value Register (ADC1_CLMS)	32	R/W	0000_0020h	<a href="#">35.3.19/871</a>
400B_B05C	ADC Minus-Side General Calibration Value Register (ADC1_CLM4)	32	R/W	0000_0200h	<a href="#">35.3.20/872</a>
400B_B060	ADC Minus-Side General Calibration Value Register (ADC1_CLM3)	32	R/W	0000_0100h	<a href="#">35.3.21/872</a>
400B_B064	ADC Minus-Side General Calibration Value Register (ADC1_CLM2)	32	R/W	0000_0080h	<a href="#">35.3.22/873</a>
400B_B068	ADC Minus-Side General Calibration Value Register (ADC1_CLM1)	32	R/W	0000_0040h	<a href="#">35.3.23/873</a>
400B_B06C	ADC Minus-Side General Calibration Value Register (ADC1_CLM0)	32	R/W	0000_0020h	<a href="#">35.3.24/874</a>

## 35.3.1 ADC Status and Control Registers 1 (ADCx\_SC1n)

SC1A is used for both software and hardware trigger modes of operation.

## Memory map and register definitions

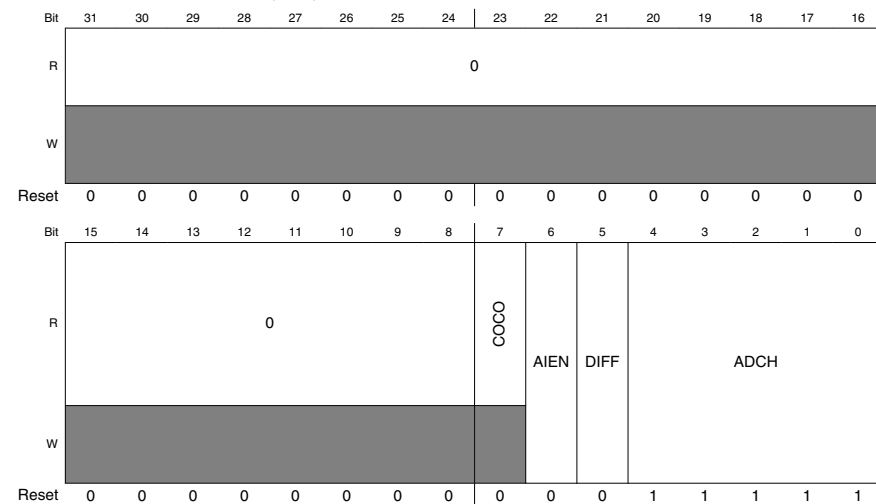
To allow sequential conversions of the ADC to be triggered by internal peripherals, the ADC can have more than one status and control register: one for each conversion. The SC1B–SC1n registers indicate potentially multiple SC1 registers for use only in hardware trigger mode. See the chip configuration information about the number of SC1n registers specific to this device. The SC1n registers have identical fields, and are used in a "ping-pong" approach to control ADC operation.

At any one point in time, only one of the SC1n registers is actively controlling ADC conversions. Updating SC1A while SC1n is actively controlling a conversion is allowed, and vice-versa for any of the SC1n registers specific to this MCU.

Writing SC1A while SC1A is actively controlling a conversion aborts the current conversion. In Software Trigger mode, when SC2[ADTRG]=0, writes to SC1A subsequently initiate a new conversion, if SC1[ADCH] contains a value other than all 1s (module disabled).

Writing any of the SC1n registers while that specific SC1n register is actively controlling a conversion aborts the current conversion. None of the SC1B–SC1n registers are used for software trigger operation and therefore writes to the SC1B–SC1n registers do not initiate a new conversion.

Address: Base address + 0h offset + (4d × i), where i=0d to 1d



## ADCx\_SC1n field descriptions

Field	Description
31–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7 COCO	Conversion Complete Flag  This is a read-only field that is set each time a conversion is completed when the compare function is disabled, or SC2[ACFE]=0 and the hardware average function is disabled, or SC3[AVGE]=0. When the compare function is enabled, or SC2[ACFE]=1, COCO is set upon completion of a conversion only if the compare result is true. When the hardware average function is enabled, or SC3[AVGE]=1, COCO is set upon completion of the selected number of conversions (determined by AVGS). COCO in SC1A is also set at the completion of a calibration sequence. COCO is cleared when the respective SC1n register is written or when the respective Rn register is read.  0 Conversion is not completed. 1 Conversion is completed.
6 AIEN	Interrupt Enable  Enables conversion complete interrupts. When COCO becomes set while the respective AIEN is high, an interrupt is asserted.  0 Conversion complete interrupt is disabled. 1 Conversion complete interrupt is enabled.
5 DIFF	Differential Mode Enable  Configures the ADC to operate in differential mode. When enabled, this mode automatically selects from the differential channels, and changes the conversion algorithm and the number of cycles to complete a conversion.  0 Single-ended conversions and input channels are selected. 1 Differential conversions and input channels are selected.
ADCH	Input channel select  Selects one of the input channels. The input channel decode depends on the value of DIFF. DAD0-DAD3 are associated with the input pin pairs DADPx and DADmX.  <b>NOTE:</b> Some of the input channel options in the bitfield-setting descriptions might not be available for your device. For the actual ADC channel assignments for your device, see the Chip Configuration details.  The successive approximation converter subsystem is turned off when the channel select bits are all set, that is, ADCH = 11111. This feature allows explicit disabling of the ADC and isolation of the input channel from all sources. Terminating continuous conversions this way prevents an additional single conversion from being performed. It is not necessary to set ADCH to all 1s to place the ADC in a low-power state when continuous conversions are not enabled because the module automatically enters a low-power state when a conversion completes.  00000 When DIFF=0, DADP0 is selected as input; when DIFF=1, DAD0 is selected as input. 00001 When DIFF=0, DADP1 is selected as input; when DIFF=1, DAD1 is selected as input. 00010 When DIFF=0, DADP2 is selected as input; when DIFF=1, DAD2 is selected as input. 00011 When DIFF=0, DADP3 is selected as input; when DIFF=1, DAD3 is selected as input. 00100 When DIFF=0, AD4 is selected as input; when DIFF=1, it is reserved. 00101 When DIFF=0, AD5 is selected as input; when DIFF=1, it is reserved. 00110 When DIFF=0, AD6 is selected as input; when DIFF=1, it is reserved. 00111 When DIFF=0, AD7 is selected as input; when DIFF=1, it is reserved.

Table continues on the next page...

## ADCx\_SC1n field descriptions (continued)

Field	Description
01000	When DIFF=0, AD8 is selected as input; when DIFF=1, it is reserved.
01001	When DIFF=0, AD9 is selected as input; when DIFF=1, it is reserved.
01010	When DIFF=0, AD10 is selected as input; when DIFF=1, it is reserved.
01011	When DIFF=0, AD11 is selected as input; when DIFF=1, it is reserved.
01100	When DIFF=0, AD12 is selected as input; when DIFF=1, it is reserved.
01101	When DIFF=0, AD13 is selected as input; when DIFF=1, it is reserved.
01110	When DIFF=0, AD14 is selected as input; when DIFF=1, it is reserved.
01111	When DIFF=0, AD15 is selected as input; when DIFF=1, it is reserved.
10000	When DIFF=0, AD16 is selected as input; when DIFF=1, it is reserved.
10001	When DIFF=0, AD17 is selected as input; when DIFF=1, it is reserved.
10010	When DIFF=0, AD18 is selected as input; when DIFF=1, it is reserved.
10011	When DIFF=0, AD19 is selected as input; when DIFF=1, it is reserved.
10100	When DIFF=0, AD20 is selected as input; when DIFF=1, it is reserved.
10101	When DIFF=0, AD21 is selected as input; when DIFF=1, it is reserved.
10110	When DIFF=0, AD22 is selected as input; when DIFF=1, it is reserved.
10111	When DIFF=0, AD23 is selected as input; when DIFF=1, it is reserved.
11000	Reserved.
11001	Reserved.
11010	When DIFF=0, Temp Sensor (single-ended) is selected as input; when DIFF=1, Temp Sensor (differential) is selected as input.
11011	When DIFF=0, Bandgap (single-ended) is selected as input; when DIFF=1, Bandgap (differential) is selected as input.
11100	Reserved.
11101	When DIFF=0, V <sub>REFSH</sub> is selected as input; when DIFF=1, -V <sub>REFSH</sub> (differential) is selected as input. Voltage reference selected is determined by SC2[REFSEL].
11110	When DIFF=0, V <sub>REFSL</sub> is selected as input; when DIFF=1, it is reserved. Voltage reference selected is determined by SC2[REFSEL].
11111	Module is disabled.

### 35.3.2 ADC Configuration Register 1 (ADCx\_CFG1)

The configuration Register 1 (CFG1) selects the mode of operation, clock source, clock divide, and configuration for low power or long sample time.

Address: Base address + 8h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								ADLPC	ADIV		ADLSMP	MODE		ADICK	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ADCx\_CFG1 field descriptions

Field	Description
31–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7 ADLPC	Low-Power Configuration Controls the power configuration of the successive approximation converter. This optimizes power consumption when higher sample rates are not required.  0 Normal power configuration. 1 Low-power configuration. The power is reduced at the expense of maximum clock speed.
6–5 ADIV	Clock Divide Select Selects the divide ratio used by the ADC to generate the internal clock ADCK.  00 The divide ratio is 1 and the clock rate is input clock. 01 The divide ratio is 2 and the clock rate is (input clock)/2. 10 The divide ratio is 4 and the clock rate is (input clock)/4. 11 The divide ratio is 8 and the clock rate is (input clock)/8.
4 ADLSMP	Sample Time Configuration Selects between different sample times based on the conversion mode selected. This field adjusts the sample period to allow higher impedance inputs to be accurately sampled or to maximize conversion speed for lower impedance inputs. Longer sample times can also be used to lower overall power consumption if continuous conversions are enabled and high conversion rates are not required. When ADLSMP=1, the long sample time select bits, (ADLSTS[1:0]), can select the extent of the long sample time.

Table continues on the next page...

ADCx\_CFG1 field descriptions (continued)

Field	Description
	0 Short sample time. 1 Long sample time.
3–2 MODE	Conversion mode selection Selects the ADC resolution mode.  00 When DIFF=0:It is single-ended 8-bit conversion; when DIFF=1, it is differential 9-bit conversion with 2's complement output. 01 When DIFF=0:It is single-ended 12-bit conversion ; when DIFF=1, it is differential 13-bit conversion with 2's complement output. 10 When DIFF=0:It is single-ended 10-bit conversion. ; when DIFF=1, it is differential 11-bit conversion with 2's complement output 11 When DIFF=0:It is single-ended 16-bit conversion...; when DIFF=1, it is differential 16-bit conversion with 2's complement output
ADICK	Input Clock Select Selects the input clock source to generate the internal clock, ADCK. Note that when the ADACK clock source is selected, it is not required to be active prior to conversion start. When it is selected and it is not active prior to a conversion start, when CFG2[ADACKEN]=0, the asynchronous clock is activated at the start of a conversion and deactivated when conversions are terminated. In this case, there is an associated clock startup delay each time the clock source is re-activated.  00 Bus clock 01 Bus clock divided by 2(BUSCLK/2) 10 Alternate clock (ALTCLK) 11 Asynchronous clock (ADACK)

### 35.3.3 ADC Configuration Register 2 (ADCx\_CFG2)

Configuration Register 2 (CFG2) selects the special high-speed configuration for very high speed conversions and selects the long sample time duration during long sample mode.

Address: Base address + Ch offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								0		MUXSEL		ADACKEN	ADHSC	ADLSTS	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

## ADCx\_CFG2 field descriptions

Field	Description
31–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7–5 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
4 MUXSEL	ADC Mux Select  Changes the ADC mux setting to select between alternate sets of ADC channels.  0 ADxxa channels are selected. 1 ADxxb channels are selected.
3 ADACKEN	Asynchronous Clock Output Enable  Enables the asynchronous clock source and the clock source output regardless of the conversion and status of CFG1[ADICLK]. Based on MCU configuration, the asynchronous clock may be used by other modules. See chip configuration information. Setting this field allows the clock to be used even while the ADC is idle or operating from a different clock source. Also, latency of initiating a single or first-continuous conversion with the asynchronous clock selected is reduced because the ADACK clock is already operational.  0 Asynchronous clock output disabled; Asynchronous clock is enabled only if selected by ADICLK and a conversion is active. 1 Asynchronous clock and clock output is enabled regardless of the state of the ADC.
2 ADHSC	High-Speed Configuration  Configures the ADC for very high-speed operation. The conversion sequence is altered with 2 ADCK cycles added to the conversion time to allow higher speed conversion clocks.  0 Normal conversion sequence selected. 1 High-speed conversion sequence selected with 2 additional ADCK cycles to total conversion time.
ADLSTS	Long Sample Time Select  Selects between the extended sample times when long sample time is selected, that is, when CFG1[ADLSMP]=1. This allows higher impedance inputs to be accurately sampled or to maximize conversion speed for lower impedance inputs. Longer sample times can also be used to lower overall power consumption when continuous conversions are enabled if high conversion rates are not required.  00 Default longest sample time; 20 extra ADCK cycles; 24 ADCK cycles total. 01 12 extra ADCK cycles; 16 ADCK cycles total sample time. 10 6 extra ADCK cycles; 10 ADCK cycles total sample time. 11 2 extra ADCK cycles; 6 ADCK cycles total sample time.

## 35.3.4 ADC Data Result Register (ADCx\_Rn)

The data result registers (Rn) contain the result of an ADC conversion of the channel selected by the corresponding status and channel control register (SC1A:SC1n). For every status and channel control register, there is a corresponding data result register.

## Memory map and register definitions

Unused bits in R n are cleared in unsigned right-aligned modes and carry the sign bit (MSB) in sign-extended 2's complement modes. For example, when configured for 10-bit single-ended mode, D[15:10] are cleared. When configured for 11-bit differential mode, D[15:10] carry the sign bit, that is, bit 10 extended through bit 15.

The following table describes the behavior of the data result registers in the different modes of operation.

Table 35-2. Data result register description

Conversion mode	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	Format
16-bit differential	S	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	Signed 2's complement
16-bit single-ended	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	Unsigned right justified
13-bit differential	S	S	S	S	D	D	D	D	D	D	D	D	D	D	D	D	Sign-extended 2's complement
12-bit single-ended	0	0	0	0	D	D	D	D	D	D	D	D	D	D	D	D	Unsigned right-justified
11-bit differential	S	S	S	S	S	S	D	D	D	D	D	D	D	D	D	D	Sign-extended 2's complement
10-bit single-ended	0	0	0	0	0	0	D	D	D	D	D	D	D	D	D	D	Unsigned right-justified
9-bit differential	S	S	S	S	S	S	S	S	D	D	D	D	D	D	D	D	Sign-extended 2's complement
8-bit single-ended	0	0	0	0	0	0	0	0	D	D	D	D	D	D	D	D	Unsigned right-justified

## NOTE

S: Sign bit or sign bit extension;

D: Data, which is 2's complement data if indicated

Address: Base address + 10h offset + (4d × i), where i=0d to 1d

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	0																D															
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	

## ADCx\_Rn field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
D	Data result

### 35.3.5 Compare Value Registers (ADCx\_CVn)

The Compare Value Registers (CV1 and CV2) contain a compare value used to compare the conversion result when the compare function is enabled, that is, SC2[ACFE]=1. This register is formatted in the same way as the Rn registers in different modes of operation for both bit position definition and value format using unsigned or sign-extended 2's complement. Therefore, the compare function uses only the CVn fields that are related to the ADC mode of operation.

The compare value 2 register (CV2) is used only when the compare range function is enabled, that is, SC2[ACREN]=1.

Address: Base address + 18h offset + (4d × i), where i=0d to 1d

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	0																CV															
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

ADCx\_CVn field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
CV	Compare Value.

### 35.3.6 Status and Control Register 2 (ADCx\_SC2)

The status and control register 2 (SC2) contains the conversion active, hardware/software trigger select, compare function, and voltage reference select of the ADC module.

Address: Base address + 20h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								ADACT	ADTRG	ACFE	ACFGT	ACREN	DMAEN	REFSEL	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ADCx\_SC2 field descriptions

Field	Description
31–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7 ADACT	Conversion Active  Indicates that a conversion or hardware averaging is in progress. ADACT is set when a conversion is initiated and cleared when a conversion is completed or aborted.  0 Conversion not in progress. 1 Conversion in progress.
6 ADTRG	Conversion Trigger Select  Selects the type of trigger used for initiating a conversion. Two types of trigger are selectable:

Table continues on the next page...



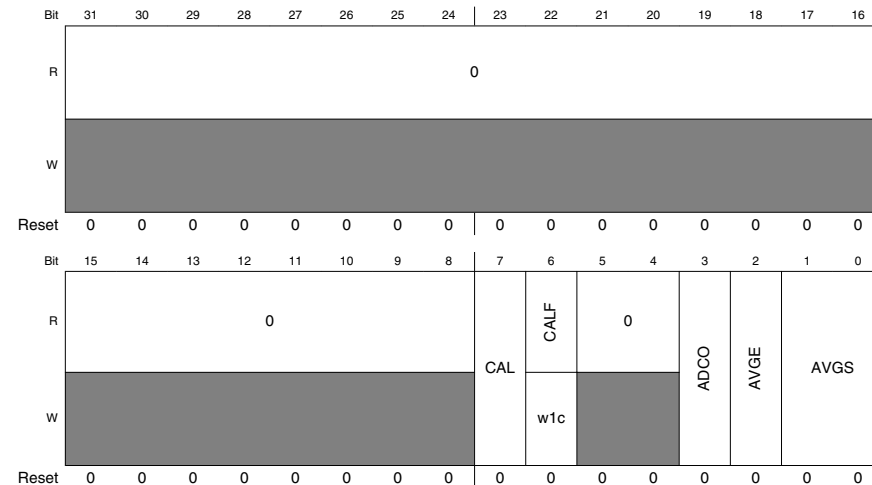
## ADCx\_SC2 field descriptions (continued)

Field	Description
	<ul style="list-style-type: none"> <li>Software trigger: When software trigger is selected, a conversion is initiated following a write to SC1A.</li> <li>Hardware trigger: When hardware trigger is selected, a conversion is initiated following the assertion of the ADHWT input after a pulse of the ADHWTSn input.</li> </ul> <p>0 Software trigger selected. 1 Hardware trigger selected.</p>
5 ACFE	<p>Compare Function Enable</p> <p>Enables the compare function.</p> <p>0 Compare function disabled. 1 Compare function enabled.</p>
4 ACFGT	<p>Compare Function Greater Than Enable</p> <p>Configures the compare function to check the conversion result relative to the CV1 and CV2 based upon the value of ACREN. ACFE must be set for ACFGT to have any effect.</p> <p>0 Configures less than threshold, outside range not inclusive and inside range not inclusive; functionality based on the values placed in CV1 and CV2. 1 Configures greater than or equal to threshold, outside and inside ranges inclusive; functionality based on the values placed in CV1 and CV2.</p>
3 ACREN	<p>Compare Function Range Enable</p> <p>Configures the compare function to check if the conversion result of the input being monitored is either between or outside the range formed by CV1 and CV2 determined by the value of ACFGT. ACFE must be set for ACFGT to have any effect.</p> <p>0 Range function disabled. Only CV1 is compared. 1 Range function enabled. Both CV1 and CV2 are compared.</p>
2 DMAEN	<p>DMA Enable</p> <p>0 DMA is disabled. 1 DMA is enabled and will assert the ADC DMA request during an ADC conversion complete event noted when any of the SC1n[COCO] flags is asserted.</p>
REFSEL	<p>Voltage Reference Selection</p> <p>Selects the voltage reference source used for conversions.</p> <p>00 Default voltage reference pin pair, that is, external pins <math>V_{REFH}</math> and <math>V_{REFL}</math> 01 Alternate reference pair, that is, <math>V_{ALTH}</math> and <math>V_{ALTl}</math>. This pair may be additional external pins or internal sources depending on the MCU configuration. See the chip configuration information for details specific to this MCU 10 Reserved 11 Reserved</p>

## 35.3.7 Status and Control Register 3 (ADCx\_SC3)

The Status and Control Register 3 (SC3) controls the calibration, continuous convert, and hardware averaging functions of the ADC module.

Address: Base address + 24h offset



## ADCx\_SC3 field descriptions

Field	Description
31–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7 CAL	<p>Calibration</p> <p>Begins the calibration sequence when set. This field stays set while the calibration is in progress and is cleared when the calibration sequence is completed. CALF must be checked to determine the result of the calibration sequence. Once started, the calibration routine cannot be interrupted by writes to the ADC registers or the results will be invalid and CALF will set. Setting CAL will abort any current conversion.</p>
6 CALF	<p>Calibration Failed Flag</p> <p>Displays the result of the calibration sequence. The calibration sequence will fail if SC2[ADTRG] = 1, any ADC register is written, or any stop mode is entered before the calibration sequence completes. Writing 1 to CALF clears it.</p> <p>0 Calibration completed normally. 1 Calibration failed. ADC accuracy specifications are not guaranteed.</p>
5–4 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

Table continues on the next page...

**ADCx\_SC3 field descriptions (continued)**

Field	Description
3 ADCO	Continuous Conversion Enable Enables continuous conversions.  0 One conversion or one set of conversions if the hardware average function is enabled, that is, AVGE=1, after initiating a conversion. 1 Continuous conversions or sets of conversions if the hardware average function is enabled, that is, AVGE=1, after initiating a conversion.
2 AVGE	Hardware Average Enable Enables the hardware average function of the ADC.  0 Hardware average function disabled. 1 Hardware average function enabled.
AVGS	Hardware Average Select Determines how many ADC conversions will be averaged to create the ADC average result.  00 4 samples averaged. 01 8 samples averaged. 10 16 samples averaged. 11 32 samples averaged.

**35.3.8 ADC Offset Correction Register (ADCx\_OFS)**

The ADC Offset Correction Register (OFS) contains the user-selected or calibration-generated offset error correction value. This register is a 2's complement, left-justified, 16-bit value. The value in OFS is subtracted from the conversion and the result is transferred into the result registers, Rn. If the result is greater than the maximum or less than the minimum result value, it is forced to the appropriate limit for the current mode of operation.

For more information regarding the calibration procedure, please refer to the [Calibration function](#) section.

Address: Base address + 28h offset

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																																
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	1	0	0

**ADCx\_OFS field descriptions**

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

Table continues on the next page...

**ADCx\_OFS field descriptions (continued)**

Field	Description
OFS	Offset Error Correction Value

**35.3.9 ADC Plus-Side Gain Register (ADCx\_PG)**

The Plus-Side Gain Register (PG) contains the gain error correction for the plus-side input in differential mode or the overall conversion in single-ended mode. PG, a 16-bit real number in binary format, is the gain adjustment factor, with the radix point fixed between PG[15] and PG[14]. This register must be written by the user with the value described in the calibration procedure. Otherwise, the gain error specifications may not be met.

For more information regarding the calibration procedure, please refer to the [Calibration function](#) section.

Address: Base address + 2Ch offset

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																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0

**ADCx\_PG field descriptions**

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
PG	Plus-Side Gain

**35.3.10 ADC Minus-Side Gain Register (ADCx\_MG)**

The Minus-Side Gain Register (MG) contains the gain error correction for the minus-side input in differential mode. This register is ignored in single-ended mode. MG, a 16-bit real number in binary format, is the gain adjustment factor, with the radix point fixed between MG[15] and MG[14]. This register must be written by the user with the value described in the calibration procedure. Otherwise, the gain error specifications may not be met.

For more information regarding the calibration procedure, please refer to the [Calibration function](#) section.

Address: Base address + 30h offset

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																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0

**ADCx\_MG field descriptions**

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
MG	Minus-Side Gain

**35.3.11 ADC Plus-Side General Calibration Value Register (ADCx\_CLPD)**

The Plus-Side General Calibration Value Registers (CLPx) contain calibration information that is generated by the calibration function. These registers contain seven calibration values of varying widths: CLP0[5:0], CLP1[6:0], CLP2[7:0], CLP3[8:0], CLP4[9:0], CLPS[5:0], and CLPD[5:0]. CLPx are automatically set when the self-calibration sequence is done, that is, CAL is cleared. If these registers are written by the user after calibration, the linearity error specifications may not be met.

For more information regarding the calibration procedure, please refer to the [Calibration function](#) section.

Address: Base address + 34h offset

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																																
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	1	0	1	0

**ADCx\_CLPD field descriptions**

Field	Description
31–6 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
CLPD	Calibration Value Calibration Value

**35.3.12 ADC Plus-Side General Calibration Value Register (ADCx\_CLPS)**

For more information, see CLPD register description.

Address: Base address + 38h offset

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																																
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	1	0	0	0	0	0	0

**ADCx\_CLPS field descriptions**

Field	Description
31–6 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
CLPS	Calibration Value Calibration Value

**35.3.13 ADC Plus-Side General Calibration Value Register (ADCx\_CLP4)**

For more information, see CLPD register description.

Address: Base address + 3Ch offset

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																																
Reset	0	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

**ADCx\_CLP4 field descriptions**

Field	Description
31–10 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
CLP4	Calibration Value Calibration Value

### 35.3.14 ADC Plus-Side General Calibration Value Register (ADCx\_CLP3)

For more information, see CLPD register description.

Address: Base address + 40h offset

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	0																CLP3															
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	1	0	0	0	0	0	0	0	0

ADCx\_CLP3 field descriptions

Field	Description
31–9 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
CLP3	Calibration Value
	Calibration Value

### 35.3.15 ADC Plus-Side General Calibration Value Register (ADCx\_CLP2)

For more information, see CLPD register description.

Address: Base address + 44h offset

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	0																CLP2															
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	1	0	0	0	0	0	0	0

ADCx\_CLP2 field descriptions

Field	Description
31–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
CLP2	Calibration Value
	Calibration Value

### 35.3.16 ADC Plus-Side General Calibration Value Register (ADCx\_CLP1)

For more information, see CLPD register description.

Address: Base address + 48h offset

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	0																CLP1															
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	1	0	0	0	0	0	0	0

ADCx\_CLP1 field descriptions

Field	Description
31–7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
CLP1	Calibration Value
	Calibration Value

### 35.3.17 ADC Plus-Side General Calibration Value Register (ADCx\_CLP0)

For more information, see CLPD register description.

Address: Base address + 4Ch offset

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	0																CLP0															
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	1	1	0	0	0	0	0

ADCx\_CLP0 field descriptions

Field	Description
31–6 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
CLP0	Calibration Value
	Calibration Value

### 35.3.18 ADC Minus-Side General Calibration Value Register (ADCx\_CLMD)

The Minus-Side General Calibration Value (CLMx) registers contain calibration information that is generated by the calibration function. These registers contain seven calibration values of varying widths: CLM0[5:0], CLM1[6:0], CLM2[7:0], CLM3[8:0], CLM4[9:0], CLMS[5:0], and CLMD[5:0]. CLMx are automatically set when the self-calibration sequence is done, that is, CAL is cleared. If these registers are written by the user after calibration, the linearity error specifications may not be met.

For more information regarding the calibration procedure, please refer to the [Calibration function](#) section.

Address: Base address + 54h offset

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																																
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	1	0	0

ADCx\_CLMD field descriptions

Field	Description
31–6 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
CLMD	Calibration Value Calibration Value

### 35.3.19 ADC Minus-Side General Calibration Value Register (ADCx\_CLMS)

For more information, see CLMD register description.

Address: Base address + 58h offset

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																																
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	1	0	0

ADCx\_CLMS field descriptions

Field	Description
31–6 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
CLMS	Calibration Value

Table continues on the next page...

ADCx\_CLMS field descriptions (continued)

Field	Description
CLMS	Calibration Value Calibration Value

### 35.3.20 ADC Minus-Side General Calibration Value Register (ADCx\_CLM4)

For more information, see CLMD register description.

Address: Base address + 5Ch offset

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																																
Reset	0	0	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

ADCx\_CLM4 field descriptions

Field	Description
31–10 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
CLM4	Calibration Value Calibration Value

### 35.3.21 ADC Minus-Side General Calibration Value Register (ADCx\_CLM3)

For more information, see CLMD register description.

Address: Base address + 60h offset

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																																
Reset	0	0	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

ADCx\_CLM3 field descriptions

Field	Description
31–9 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
CLM3	Calibration Value Calibration Value

### 35.3.22 ADC Minus-Side General Calibration Value Register (ADCx\_CLM2)

For more information, see CLMD register description.

Address: Base address + 64h offset

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	0																CLM2															
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	1	0	0	0	0	0	0	0

ADCx\_CLM2 field descriptions

Field	Description
31–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
CLM2	Calibration Value Calibration Value

### 35.3.23 ADC Minus-Side General Calibration Value Register (ADCx\_CLM1)

For more information, see CLMD register description.

Address: Base address + 68h offset

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	0																CLM1															
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	1	0	0	0	0	0	0	0

ADCx\_CLM1 field descriptions

Field	Description
31–7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
CLM1	Calibration Value Calibration Value

### 35.3.24 ADC Minus-Side General Calibration Value Register (ADCx\_CLM0)

For more information, see CLMD register description.

Address: Base address + 6Ch offset

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	0																CLM0															
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	1	0	0	0	0	0	0	0

ADCx\_CLM0 field descriptions

Field	Description
31–6 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
CLM0	Calibration Value Calibration Value

## 35.4 Functional description

The ADC module is disabled during reset, in Low-Power Stop mode, or when SC1n[ADCH] are all high; see the power management information for details. The module is idle when a conversion has completed and another conversion has not been initiated. When it is idle and the asynchronous clock output enable is disabled, or CFG2[ADACKEN]=0, the module is in its lowest power state. The ADC can perform an analog-to-digital conversion on any of the software selectable channels. All modes perform conversion by a successive approximation algorithm.

To meet accuracy specifications, the ADC module must be calibrated using the on-chip calibration function.

See [Calibration function](#) for details on how to perform calibration.

When the conversion is completed, the result is placed in the Rn data registers. The respective SC1n[COCO] is then set and an interrupt is generated if the respective conversion complete interrupt has been enabled, or, when SC1n[AIEN]=1.

The ADC module has the capability of automatically comparing the result of a conversion with the contents of the CV1 and CV2 registers. The compare function is enabled by setting SC2[ACFE] and operates in any of the conversion modes and configurations.

The ADC module has the capability of automatically averaging the result of multiple conversions. The hardware average function is enabled by setting SC3[AVGE] and operates in any of the conversion modes and configurations.

#### NOTE

For the chip specific modes of operation, see the power management information of this MCU.

### 35.4.1 Clock select and divide control

One of four clock sources can be selected as the clock source for the ADC module.

This clock source is then divided by a configurable value to generate the input clock ADCK, to the module. The clock is selected from one of the following sources by means of CFG1[ADICK].

- Bus clock. This is the default selection following reset.
- Bus clock divided by two. For higher bus clock rates, this allows a maximum divide-by-16 of the bus clock using CFG1[ADIV].
- ALTCLK: As defined for this MCU. See the chip configuration information. Conversions are possible using ALTCLK as the input clock source while the MCU is in Normal Stop mode.
- Asynchronous clock (ADACK): This clock is generated from a clock source within the ADC module. When the ADACK clock source is selected, it is not required to be active prior to conversion start. When it is selected and it is not active prior to a conversion start CFG2[ADACKEN]=0, ADACK is activated at the start of a conversion and deactivated when conversions are terminated. In this case, there is an associated clock startup delay each time the clock source is re-activated. To avoid the conversion time variability and latency associated with the ADACK clock startup, set CFG2[ADACKEN]=1 and wait the worst-case startup time of 5  $\mu$ s prior to initiating any conversions using the ADACK clock source. Conversions are possible using ADACK as the input clock source while the MCU is in Normal Stop mode. See [Power Control](#) for more information.

Whichever clock is selected, its frequency must fall within the specified frequency range for ADCK. If the available clocks are too slow, the ADC may not perform according to specifications. If the available clocks are too fast, the clock must be divided to the appropriate frequency. This divider is specified by CFG1[ADIV] and can be divide-by 1, 2, 4, or 8.

### 35.4.2 Voltage reference selection

The ADC can be configured to accept one of the two voltage reference pairs as the reference voltage ( $V_{REFSH}$  and  $V_{REFSL}$ ) used for conversions.

Each pair contains a positive reference that must be between the minimum Ref Voltage High and  $V_{DDA}$ , and a ground reference that must be at the same potential as  $V_{SSA}$ . The two pairs are external ( $V_{REFH}$  and  $V_{REFL}$ ) and alternate ( $V_{ALTH}$  and  $V_{ALTTL}$ ). These voltage references are selected using SC2[REFSEL]. The alternate ( $V_{ALTH}$  and  $V_{ALTTL}$ ) voltage reference pair may select additional external pins or internal sources depending on MCU configuration. See the chip configuration information on the voltage references specific to this MCU.

### 35.4.3 Hardware trigger and channel selects

The ADC module has a selectable asynchronous hardware conversion trigger, ADHWT, that is enabled when SC2[ADTRG] is set and a hardware trigger select event, ADHWTSn, has occurred.

This source is not available on all MCUs. See the chip-specific ADC information for information on the ADHWT source and the ADHWTSn configurations specific to this MCU.

When an ADHWT source is available and hardware trigger is enabled, that is SC2[ADTRG]=1, a conversion is initiated on the rising-edge of ADHWT after a hardware trigger select event, that is, ADHWTSn, has occurred. If a conversion is in progress when a rising-edge of a trigger occurs, the rising-edge is ignored. In continuous convert configuration, only the initial rising-edge to launch continuous conversions is observed, and until conversion is aborted, the ADC continues to do conversions on the same SCn register that initiated the conversion. The hardware trigger function operates in conjunction with any of the conversion modes and configurations.

The hardware trigger select event, ADHWTSn, must be set prior to the receipt of the ADHWT signal. If these conditions are not met, the converter may ignore the trigger or use the incorrect configuration. If a hardware trigger select event is asserted during a conversion, it must stay asserted until the end of current conversion and remain set until the receipt of the ADHWT signal to trigger a new conversion. The channel and status fields selected for the conversion depend on the active trigger select signal:

- ADHWTSn active selects SC1A.
- ADHWTSn active selects SC1n.

**Note**

Asserting more than one hardware trigger select signal (ADHWTSn) at the same time results in unknown results. To avoid this, select only one hardware trigger select signal (ADHWTSn) prior to the next intended conversion.

When the conversion is completed, the result is placed in the Rn registers associated with the ADHWTSn received. For example:

- ADHWTSn active selects RA register
- ADHWTSn active selects Rn register

The conversion complete flag associated with the ADHWTSn received, that is, SC1n[COCO], is then set and an interrupt is generated if the respective conversion complete interrupt has been enabled, that is, SC1n[AIEN]=1.

**35.4.4 Conversion control**

Conversions can be performed as determined by CFG1[MODE] and SC1n[DIFF] as shown in the description of CFG1[MODE].

Conversions can be initiated by a software or hardware trigger.

In addition, the ADC module can be configured for:

- Low-power operation
- Long sample time
- Continuous conversion
- Hardware average
- Automatic compare of the conversion result to a software determined compare value

**35.4.4.1 Initiating conversions**

A conversion is initiated:

- Following a write to SC1A, with SC1n[ADCH] not all 1's, if software triggered operation is selected, that is, when SC2[ADTRG]=0.
- Following a hardware trigger, or ADHWT event, if hardware triggered operation is selected, that is, SC2[ADTRG]=1, and a hardware trigger select event, ADHWTSn, has occurred. The channel and status fields selected depend on the active trigger select signal:
  - ADHWTSn active selects SC1A.

**Functional description**

- ADHWTSn active selects SC1n.
- if neither is active, the off condition is selected

**Note**

Selecting more than one ADHWTSn prior to a conversion completion will result in unknown results. To avoid this, select only one ADHWTSn prior to a conversion completion.

- Following the transfer of the result to the data registers when continuous conversion is enabled, that is, when SC3[ADCO] = 1.

If continuous conversions are enabled, a new conversion is automatically initiated after the completion of the current conversion. In software triggered operation, that is, when SC2[ADTRG] = 0, continuous conversions begin after SC1A is written and continue until aborted. In hardware triggered operation, that is, when SC2[ADTRG] = 1 and one ADHWTSn event has occurred, continuous conversions begin after a hardware trigger event and continue until aborted.

If hardware averaging is enabled, a new conversion is automatically initiated after the completion of the current conversion until the correct number of conversions are completed. In software triggered operation, conversions begin after SC1A is written. In hardware triggered operation, conversions begin after a hardware trigger. If continuous conversions are also enabled, a new set of conversions to be averaged are initiated following the last of the selected number of conversions.

**35.4.4.2 Completing conversions**

A conversion is completed when the result of the conversion is transferred into the data result registers, Rn. If the compare functions are disabled, this is indicated by setting of SC1n[COCO]. If hardware averaging is enabled, the respective SC1n[COCO] sets only if the last of the selected number of conversions is completed. If the compare function is enabled, the respective SC1n[COCO] sets and conversion result data is transferred only if the compare condition is true. If both hardware averaging and compare functions are enabled, then the respective SC1n[COCO] sets only if the last of the selected number of conversions is completed and the compare condition is true. An interrupt is generated if the respective SC1n[AIEN] is high at the time that the respective SC1n[COCO] is set.



### 35.4.4.3 Aborting conversions

Any conversion in progress is aborted when:

- Writing to SC1A while it is actively controlling a conversion, aborts the current conversion. In Software Trigger mode, when SC2[ADTRG]=0, a write to SC1A initiates a new conversion if SC1A[ADCH] is equal to a value other than all 1s. Writing to any of the SC1B–SC1n registers while that specific SC1B–SC1n register is actively controlling a conversion aborts the current conversion. The SC1(B-n) registers are not used for software trigger operation and therefore writes to the SC1(B-n) registers do not initiate a new conversion.
- A write to any ADC register besides the SC1A-SC1n registers occurs. This indicates that a change in mode of operation has occurred and the current conversion is therefore invalid.
- The MCU is reset or enters Low-Power Stop modes.
- The MCU enters Normal Stop mode with ADACK or Alternate Clock Sources not enabled.

When a conversion is aborted, the contents of the data registers, Rn, are not altered. The data registers continue to be the values transferred after the completion of the last successful conversion. If the conversion was aborted by a reset or Low-Power Stop modes, RA and Rn return to their reset states.

### 35.4.4.4 Power control

The ADC module remains in its idle state until a conversion is initiated. If ADACK is selected as the conversion clock source, but the asynchronous clock output is disabled, that is CFG2[ADACKEN]=0, the ADACK clock generator also remains in its idle state (disabled) until a conversion is initiated. If the asynchronous clock output is enabled, that is, CFG2[ADACKEN]=1, it remains active regardless of the state of the ADC or the MCU power mode.

Power consumption when the ADC is active can be reduced by setting CFG1[ADLPC]. This results in a lower maximum value for  $f_{ADCK}$ .

### 35.4.4.5 Sample time and total conversion time

For short sample, that is, when CFG1[ADLSMP]=0, there is a 2-cycle adder for first conversion over the base sample time of four ADCK cycles. For high-speed conversions, that is, when CFG2[ADHSC]=1, there is an additional 2-cycle adder on any conversion. The table below summarizes sample times for the possible ADC configurations.

ADC configuration			Sample time (ADCK cycles)	
CFG1[ADLSMP]	CFG2[ADLSTS]	CFG2[ADHSC]	First or Single	Subsequent
0	X	0	6	4
1	00	0	24	
1	01	0	16	
1	10	0	10	
1	11	0	6	
0	X	1	8	6
1	00	1	26	
1	01	1	18	
1	10	1	12	
1	11	1	8	

The total conversion time depends upon:

- The sample time as determined by CFG1[ADLSMP] and CFG2[ADLSTS]
- The MCU bus frequency
- The conversion mode, as determined by CFG1[MODE] and SC1n[DIFF]
- The high-speed configuration, that is, CFG2[ADHSC]
- The frequency of the conversion clock, that is,  $f_{ADCK}$ .

CFG2[ADHSC] is used to configure a higher clock input frequency. This will allow faster overall conversion times. To meet internal ADC timing requirements, CFG2[ADHSC] adds additional ADCK cycles. Conversions with CFG2[ADHSC]=1 take two more ADCK cycles. CFG2[ADHSC] must be used when the ADCLK exceeds the limit for CFG2[ADHSC]=0.

After the module becomes active, sampling of the input begins.

1. CFG1[ADLSMP] and CFG2[ADLSTS] select between sample times based on the conversion mode that is selected.
2. When sampling is completed, the converter is isolated from the input channel and a successive approximation algorithm is applied to determine the digital value of the analog signal.
3. The result of the conversion is transferred to Rn upon completion of the conversion algorithm.

If the bus frequency is less than  $f_{ADCK}$ , precise sample time for continuous conversions cannot be guaranteed when short sample is enabled, that is, when CFG1[ADLSMP]=0.

The maximum total conversion time is determined by the clock source chosen and the divide ratio selected. The clock source is selectable by CFG1[ADICLK], and the divide ratio is specified by CFG1[ADIV].

The maximum total conversion time for all configurations is summarized in the equation below. See the following tables for the variables referenced in the equation.

$$\text{ConversionTime} = \text{SFCAdder} + \text{AverageNum} \times (\text{BCT} + \text{LSTAdder} + \text{HSCAdder})$$

Equation 1. Conversion time equation

Table 35-3. Single or first continuous time adder (SFCAdder)

CFG1[ADLSMP]	CFG2[ADACKEN]	CFG1[ADICLK]	Single or first continuous time adder (SFCAdder)
1	x	0x, 10	3 ADCK cycles + 5 bus clock cycles
1	1	11	3 ADCK cycles + 5 bus clock cycles <sup>1</sup>
1	0	11	5 $\mu$ s + 3 ADCK cycles + 5 bus clock cycles
0	x	0x, 10	5 ADCK cycles + 5 bus clock cycles
0	1	11	5 ADCK cycles + 5 bus clock cycles <sup>1</sup>
0	0	11	5 $\mu$ s + 5 ADCK cycles + 5 bus clock cycles

1. To achieve this time, CFG2[ADACKEN] must be 1 for at least 5  $\mu$ s prior to the conversion is initiated.

Table 35-4. Average number factor (AverageNum)

SC3[AVGE]	SC3[AVGS]	Average number factor (AverageNum)
0	xx	1
1	00	4
1	01	8
1	10	16
1	11	32

Table 35-5. Base conversion time (BCT)

Mode	Base conversion time (BCT)
8b single-ended	17 ADCK cycles
9b differential	27 ADCK cycles
10b single-ended	20 ADCK cycles
11b differential	30 ADCK cycles
12b single-ended	20 ADCK cycles
13b differential	30 ADCK cycles

Table continues on the next page...

Table 35-5. Base conversion time (BCT) (continued)

Mode	Base conversion time (BCT)
16b single-ended	25 ADCK cycles
16b differential	34 ADCK cycles

Table 35-6. Long sample time adder (LSTAdder)

CFG1[ADLSMP]	CFG2[ADLSTS]	Long sample time adder (LSTAdder)
0	xx	0 ADCK cycles
1	00	20 ADCK cycles
1	01	12 ADCK cycles
1	10	6 ADCK cycles
1	11	2 ADCK cycles

Table 35-7. High-speed conversion time adder (HSCAdder)

CFG2[ADHSC]	High-speed conversion time adder (HSCAdder)
0	0 ADCK cycles
1	2 ADCK cycles

### Note

The ADCK frequency must be between  $f_{ADCK}$  minimum and  $f_{ADCK}$  maximum to meet ADC specifications.

### 35.4.4.6 Conversion time examples

The following examples use the [Equation 1 on page 881](#), and the information provided in [Table 35-3](#) through [Table 35-7](#).

#### 35.4.4.6.1 Typical conversion time configuration

A typical configuration for ADC conversion is:

- 10-bit mode, with the bus clock selected as the input clock source
- The input clock divide-by-1 ratio selected
- Bus frequency of 8 MHz
- Long sample time disabled
- High-speed conversion disabled

The conversion time for a single conversion is calculated by using the [Equation 1 on page 881](#), and the information provided in [Table 35-3](#) through [Table 35-7](#). The table below lists the variables of [Equation 1 on page 881](#).

**Table 35-8. Typical conversion time**

Variable	Time
SFCAdder	5 ADCK cycles + 5 bus clock cycles
AverageNum	1
BCT	20 ADCK cycles
LSTAdder	0
HSCAdder	0

The resulting conversion time is generated using the parameters listed in the preceding table. Therefore, for a bus clock and an ADCK frequency equal to 8 MHz, the resulting conversion time is 3.75  $\mu$ s.

#### 35.4.4.6.2 Long conversion time configuration

A configuration for long ADC conversion is:

- 16-bit differential mode with the bus clock selected as the input clock source
- The input clock divide-by-8 ratio selected
- Bus frequency of 8 MHz
- Long sample time enabled
- Configured for longest adder
- High-speed conversion disabled
- Average enabled for 32 conversions

The conversion time for this conversion is calculated by using the [Equation 1 on page 881](#), and the information provided in [Table 35-3](#) through [Table 35-7](#). The following table lists the variables of the [Equation 1 on page 881](#).

**Table 35-9. Typical conversion time**

Variable	Time
SFCAdder	3 ADCK cycles + 5 bus clock cycles
AverageNum	32
BCT	34 ADCK cycles
LSTAdder	20 ADCK cycles
HSCAdder	0

#### Functional description

The resulting conversion time is generated using the parameters listed in the preceding table. Therefore, for bus clock equal to 8 MHz and ADCK equal to 1 MHz, the resulting conversion time is 57.625  $\mu$ s, that is, AverageNum. This results in a total conversion time of 1.844 ms.

#### 35.4.4.6.3 Short conversion time configuration

A configuration for short ADC conversion is:

- 8-bit Single-Ended mode with the bus clock selected as the input clock source
- The input clock divide-by-1 ratio selected
- Bus frequency of 20 MHz
- Long sample time disabled
- High-speed conversion enabled

The conversion time for this conversion is calculated by using the [Equation 1 on page 881](#), and the information provided in [Table 35-3](#) through [Table 35-7](#). The table below lists the variables of [Equation 1 on page 881](#).

**Table 35-10. Typical conversion time**

Variable	Time
SFCAdder	5 ADCK cycles + 5 bus clock cycles
AverageNum	1
BCT	17 ADCK cycles
LSTAdder	0 ADCK cycles
HSCAdder	2

The resulting conversion time is generated using the parameters listed in the preceding table. Therefore, for bus clock and ADCK frequency equal to 20 MHz, the resulting conversion time is 1.45  $\mu$ s.

#### 35.4.4.7 Hardware average function

The hardware average function can be enabled by setting SC3[AVGE]=1 to perform a hardware average of multiple conversions. The number of conversions is determined by the AVGS[1:0] bits, which can select 4, 8, 16, or 32 conversions to be averaged. While the hardware average function is in progress, SC2[ADACT] will be set.

After the selected input is sampled and converted, the result is placed in an accumulator from which an average is calculated once the selected number of conversions have been completed. When hardware averaging is selected, the completion of a single conversion will not set SC1n[COCO].