**Welcome to E-XFL.COM**

## What is "Embedded - Microcontrollers"?

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

## Applications of "Embedded - Microcontrollers"

| Details | |
|---|---|
| Product Status | Active |
| Core Processor | ARM® Cortex®-M7 |
| Core Size | 32-Bit Single-Core |
| Speed | 300MHz |
| Connectivity | CANbus, Ethernet, I²C, IrDA, LINbus, MMC/SD/SDIO, QSPI, SPI, UART/USART, USB |
| Peripherals | Brown-out Detect/Reset, DMA, I²S, POR, PWM, WDT |
| Number of I/O | 75 |
| Program Memory Size | 1MB (1M x 8) |
| Program Memory Type | FLASH |
| EEPROM Size | - |
| RAM Size | 384K x 8 |
| Voltage - Supply (Vcc/Vdd) | 1.62V ~ 3.6V |
| Data Converters | A/D 10x12b; D/A 2x12b |
| Oscillator Type | Internal |
| Operating Temperature | -40°C ~ 105°C (TA) |
| Mounting Type | Surface Mount |
| Package / Case | 100-LQFP |
| Supplier Device Package | 100-LQFP (14x14) |
| Purchase URL | https://www.e-xfl.com/product-detail/microchip-technology/atsame70n20a-an |

### 16.7.4 Embedded Trace Module (ETM) Pins

The Embedded Trace Module (ETM) uses the Trace Port Interface Unit (TPIU) to export data out of the system.

The TPUI features the pins:

- TRACECLK–always exported to enable synchronization back with the data. PCK3 is used internally.
- TRACED0–3–the instruction trace stream.

### 16.7.5 Flash Patch Breakpoint (FPB)

The FPB implements hardware breakpoints.

### 16.7.6 Data Watchpoint and Trace (DWT)

The DWT contains four comparators which can be configured to generate:

- PC sampling packets at set intervals
- PC or Data watchpoint packets
- Watchpoint event to halt core

The DWT contains counters for:

- Clock cycle (CYCCNT)
- Folded instructions
- Load Store Unit (LSU) operations
- Sleep cycles
- CPI (all instruction cycles except for the first cycle)
- Interrupt overhead

### 16.7.7 Instrumentation Trace Macrocell (ITM)

The ITM is an application driven trace source that supports `printf` style debugging to trace Operating System (OS) and application events, and emits diagnostic system information. The ITM emits trace information as packets which can be generated by three different sources with several priority levels:

- Software trace: Software can write directly to ITM stimulus registers. This can be done using the `printf` function. For more information, refer to 16.7.5 Flash Patch Breakpoint (FPB).
- Hardware trace: The ITM emits packets generated by the DWT.
- Timestamping: Timestamps are emitted relative to packets. The ITM contains a 21-bit counter to generate the timestamp.

#### 16.7.7.1 How to Configure the ITM

The following example describes how to output trace data in asynchronous trace mode.

Configure the TPIU for asynchronous trace mode. Refer to 16.7.7.3 How to Configure the TPIU.

1. Enable the write accesses into the ITM registers by writing “0xC5ACCE55” into the Lock Access Register (Address: 0xE0000FB0)
2. Write 0x00010015 into the Trace Control register:
   - Enable ITM.
   - Enable Synchronization packets.
   - Enable SWO behavior.
   - Fix the ATB ID to 1.

1. Round-robin Arbitration (default)
2. Fixed Priority Arbitration

Each algorithm may be complemented by selecting a default master configuration for each slave.

When re-arbitration is required, specific conditions apply. Refer to the "Arbitration Rules" section.

#### 19.3.3.1 Arbitration Rules

Each arbiter has the ability to arbitrate between requests from two or more masters. To avoid burst breaking and to provide maximum throughput for slave interfaces, arbitration should take place during the following cycles:

- Idle cycles: When a slave is not connected to any master or is connected to a master which is not currently accessing it
- Single cycles: When a slave is performing a single access
- End of Burst cycles: When the current cycle is the last cycle of a burst transfer. For a defined length burst, predicted end of burst matches the size of the transfer but is managed differently for undefined length burst. Refer to the "Undefined Length Burst Arbitration" section.
- Slot cycle limit: When the slot cycle counter has reached the limit value indicating that the current master access is too long and must be broken. Refer to the "Slot Cycle Limit Arbitration" section.

##### 19.3.3.1.1 Undefined Length Burst Arbitration

In order to prevent slave handling during undefined length bursts, the user can trigger the re-arbitration before the end of the incremental bursts. The re-arbitration period can be selected from the following Undefined Length Burst Type (ULBT) possibilities:

1. Unlimited: no predetermined end of burst is generated. This value enables 1-Kbyte burst lengths.
2. 1-beat bursts: predetermined end of burst is generated at each single transfer during the INCR transfer.
3. 4-beat bursts: predetermined end of burst is generated at the end of each 4-beat boundary during INCR transfer.
4. 8-beat bursts: predetermined end of burst is generated at the end of each 8-beat boundary during INCR transfer.
5. 16-beat bursts: predetermined end of burst is generated at the end of each 16-beat boundary during INCR transfer.
6. 32-beat bursts: predetermined end of burst is generated at the end of each 32-beat boundary during INCR transfer.
7. 64-beat bursts: predetermined end of burst is generated at the end of each 64-beat boundary during INCR transfer.
8. 128-beat bursts: predetermined end of burst is generated at the end of each 128-beat boundary during INCR transfer.

The use of undefined length16-beat bursts, or less, is discouraged since this decreases the overall bus bandwidth due to arbitration and slave latencies at each first access of a burst.

If the master does not permanently and continuously request the same slave or has an intrinsically limited average throughput, the ULBT should be left at its default unlimited value, knowing that the AHB specification natively limits all word bursts to 256 beats and double-word bursts to 128 beats because of its 1-Kbyte address boundaries.

Unless duly needed, the ULBT should be left at its default value of 0 for power saving.

This selection is made through the ULBT field of the Master Configuration Registers (MATRIX_MCFG).

3. When Flash programming is completed, the bit EEFC_FSR.FRDY rises. If an interrupt has been enabled by setting the bit EEFC_FMR.FRDY, the interrupt line of the EEFC is activated.

Three errors can be detected in EEFC_FSR after a programming sequence:

- Command Error: A bad keyword has been written in EEFC_FCR.
- Lock Error: The page to be programmed belongs to a locked region. A command must be run previously to unlock the corresponding region.
- Flash Error: When programming is completed, the WriteVerify test of the Flash memory has failed.

Only one page can be programmed at a time. It is possible to program all the bits of a page (full page programming) or only some of the bits of the page (partial page programming).

Depending on the number of bits to be programmed within the page, the EEFC adapts the write operations required to program the Flash.

When a ‘Write Page’ (WP) command is issued, the EEFC starts the programming sequence and all the bits written at 0 in the latch buffer are cleared in the Flash memory array.

During programming, i.e., until EEFC_FSR.FDRY rises, access to the Flash is not allowed.

#### 22.4.3.2.1 Full Page Programming

To program a full page, all the bits of the page must be erased before writing the latch buffer and issuing the WP command. The latch buffer must be written in ascending order, starting from the first address of the page. See Figure 22-8.

#### 22.4.3.2.2 Partial Page Programming

To program only part of a page using the WP command, the following constraints must be respected:

- Data to be programmed must be contained in integer multiples of 128-bit address-aligned words.
- 128-bit words can be programmed only if all the corresponding bits in the Flash array are erased (at logical value ‘1’).

See 22.4.3.2.4 Programming Bytes.

#### 22.4.3.2.3 Optimized Partial Page Programming

The EEFC automatically detects the number of 128-bit words to be programmed. If only one 128-bit aligned word is to be programmed in the Flash array, the process is optimized to reduce the time needed for programming.

If several 128-bit words are to be programmed, a standard page programming operation is performed.

See Figure 22-10.

#### 22.4.3.2.4 Programming Bytes

Individual bytes can be programmed using the Partial Page Programming mode.

In this case, an area of 128 bits must be reserved for each byte.

Refer to Figure 22-11

# 23. Supply Controller (SUPC)

## 23.1 Description

The Supply Controller (SUPC) controls the supply voltages of the system and manages the Backup mode. In this mode, current consumption is reduced to a few microamps for backup power retention. Exit from this mode is possible on multiple wakeup sources. The SUPC also generates the slow clock by selecting either the slow RC oscillator or the 32.768 kHz crystal oscillator.

## 23.2 Embedded Characteristics

- Management of the Core Power Supply VDDCORE and Backup Mode via the Embedded Voltage Regulator
- Supply Monitor Detection on VDDIO or a Brownout Detection on VDDCORE Triggers a Core Reset
- Generates the Slow Clock SLCK by selecting either the 22-42 kHz Slow RC Oscillator or the 32.768 kHz Crystal Oscillator
- Backup SRAM
- Low-power Tamper Detection on Two Inputs
- Anti-tampering by Immediate Clear of the General-purpose Backup Registers
- Support of Multiple Wakeup Sources for Exit from Backup Mode
  - 14 Wakeup Inputs with Programmable Debouncing
  - Real-Time Clock Alarm
  - Real-Time Timer Alarm
  - Supply Monitor Detection on VDDIO, with Programmable Scan Period and Voltage Threshold

### 25.5.3 Reinforced Safety Watchdog Timer Status Register

**Name:** RSWDT_SR
**Offset:** 0x08
**Reset:** 0x00000000
**Property:** Read-only

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | WDUNF |
| Access | | | | | | | | |
| Reset | | | | | | | | 0 |

**Bit 0 – WDUNF** Watchdog Underflow

| Value | Description |
|---|---|
| 0 | No watchdog underflow occurred since the last read of RSWDT_SR. |
| 1 | At least one watchdog underflow occurred since the last read of RSWDT_SR. |

### 31.20.35 PLL Maximum Multiplier Value Register

**Name:** PMC_PMMR
**Offset:** 0x0130
**Reset:** 0x000007FF
**Property:** Read/Write

This register can only be written if the WPEN bit is cleared in the PMC Write Protection Mode Register.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

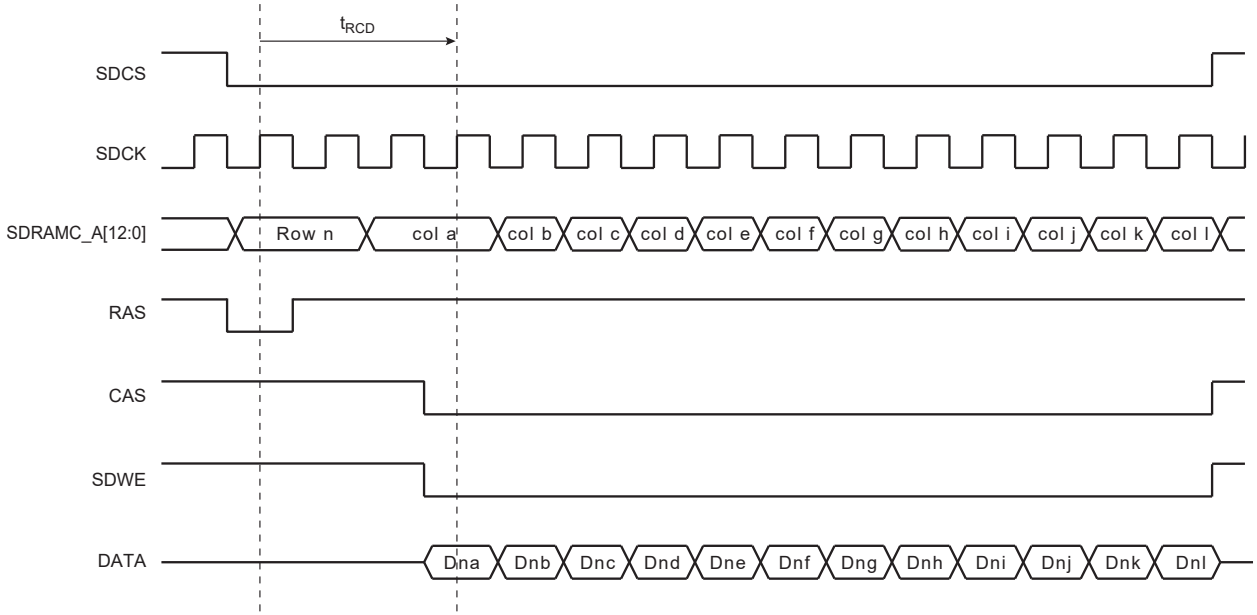
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | PLLA_MMAX[10:8] | | |
| Reset | | | | | | 1 | 1 | 1 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Access | PLLA_MMAX[7:0] | | | | | | | |
| Reset | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

**Bits 10:0 – PLLA_MMAX[10:0]** PLLA Maximum Allowed Multiplier Value
Defines the maximum value of multiplication factor that can be sent to PLLA. Any value of the MULA field (see PMC Clock Generator PLLA Register) above PLLA_MMAX is saturated to PLLA_MMAX.
PLLA_MMAX write operation is cancelled in the following cases:

- The value of MULA is currently saturated by PLLA_MMAX
- The user is trying to write a value of PLLA_MMAX that is smaller than the current value of MULA

**Figure 34-2. Write Burst SDRAM Access**




### 34.6.2 SDRAM Controller Read Cycle

The SDRAMC allows burst access, incremental burst of unspecified length or single access. In all cases, the SDRAMC keeps track of the active row in each bank, thus maximizing performance of the SDRAM. If row and bank addresses do not match the previous row/bank address, then the SDRAMC automatically generates a precharge command, activates the new row and starts the read command. To comply with the SDRAM timing parameters, additional clock cycles on SDCK are inserted between precharge and active commands ($t_{RP}$), and between active and read commands ($t_{RCD}$). These two parameters are set in the SDRAMC_CR. After a read command, additional wait states are generated to comply with the CAS latency ( 2 or 3 clock delays specified in the SDRAMC_CR).

For a single access or an incremented burst of unspecified length, the SDRAMC anticipates the next access. While the last value of the column is returned by the SDRAMC on the bus, the SDRAMC anticipates the read to the next column and thus anticipates the CAS latency. This reduces the effect of the CAS latency on the internal bus.

For burst access of specified length (4, 8, 16 words), access is not anticipated. This case leads to the best performance. If the burst is broken (border, Busy mode, etc.), the next access is handled as an incrementing burst of unspecified length.

### 37.6.3 ISI Preview Size Register

**Name:** ISI_PSIZE
**Offset:** 0x08
**Reset:** 0x00000000
**Property:** Read/Write

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | PREV_HSIZE[9:8] | |
| Access | | | | | | | R/W | R/W |
| Reset | | | | | | | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | PREV_HSIZE[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | PREV_VSIZE[9:8] | |
| Access | | | | | | | R/W | R/W |
| Reset | | | | | | | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | PREV_VSIZE[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 25:16 – PREV_HSIZE[9:0]** Horizontal Size for the Preview Path
PREV_HSIZE = Horizontal Preview size - 1 (640 max only in RGB mode).

**Bits 9:0 – PREV_VSIZE[9:0]** Vertical Size for the Preview Path
PREV_VSIZE = Vertical Preview size - 1 (480 max only in RGB mode).

When the priority queuing feature is enabled, the number of interrupt outputs from the GMAC core is increased to match the number of supported queues. The number of Interrupt Status registers is increased by the same number. Only DMA related events are reported using the individual interrupt outputs, as the GMAC can relate these events to specific queues. All other events generated within the GMAC are reported in the interrupt associated with the lowest priority queue. For the lowest priority queue (or the only queue when only 1 queue is selected), the Interrupt Status register is located at address 0x24. For all other queues, the Interrupt Status register is located at sequential addresses starting at address 0x400.

**Note:** The address matching is the first level of filtering. If there is a match, the screeners are the next level of filtering for routing the data to the appropriate queue. See MAC Filtering Block for more details.

The additional screening done by the functions Compare A, B, and C each have an enable bit and compare register field. COMPA, COMPB and COMPC in GMAC_ST2RPQ are pointers to a configured offset (OFFSVAL), value (COMPVAL), and mask (MASKVAL). If enabled, the compare is true if the data at the offset into the frame, ANDed with MASKVAL, is equal to the value of COMPVAL ANDed with MASKVAL. A 16-bit word comparison is done. The byte at the offset number of bytes from the index start is compared to bits 7:0 of the configured COMPVAL and MASKVAL. The byte at the offset number of bytes + 1 from the index start is compared to bits 15:8 of the configured COMPVAL and MASKVAL.

The offset value in bytes, OFFSVAL, ranges from 0 to 127 bytes from either the start of the frame, the byte after the EtherType field, the byte after the IP header (IPv4 or IPv6) or the byte after the TCP/UDP header. Note the logic to decode the IP header or the TCP/UDP header is reused from the TCP/UDP/IP checksum offload logic and therefore has the same restrictions on use (the main limitation is that IP fragmentation is not supported). Refer to the Checksum Offload for IP, TCP and UDP section of this documentation for further details.

Compare A, B, and C use a common set of 24 GMAC_ST2CW0/1 registers, thus all COMPA, COMPB and COMPC fields in the registers GMAC_ST2RPQ point to a single pool of 24 GMAC_ST2CW0/1 registers.

Note that Compare A, B and C together allow matching against an arbitrary 48 bits of data and so can be used to match against a MAC address.

All enabled comparisons are ANDed together to form the overall type 2 screening match.

**Related Links**

38.6.6 Checksum Offload for IP, TCP and UDP

### 38.6.4 MAC Transmit Block

The MAC transmitter can operate in either half duplex or full duplex mode and transmits frames in accordance with the Ethernet IEEE 802.3 standard. In half duplex mode, the CSMA/CD protocol of the IEEE 802.3 specification is followed.

A small input buffer receives data through the FIFO interface which will extract data in 32-bit form. All subsequent processing prior to the final output is performed in bytes.

Transmit data can be output using the MII interface.

Frame assembly starts by adding preamble and the start frame delimiter. Data is taken from the transmit FIFO interface a word at a time.

If necessary, padding is added to take the frame length to 60 bytes. CRC is calculated using an order 32-bit polynomial. This is inverted and appended to the end of the frame taking the frame length to a

- the receive and transmit bank FIFO counters,
- all registers of this endpoint (USBHS_DEVEPTCFGx, USBHS_DEVEPTISRx, the Endpoint x Control (USBHS_DEVEPTIMRx) register), except its configuration (USBHS_DEVEPTCFGx.ALLOC, USBHS_DEVEPTCFGx.EPBK, USBHS_DEVEPTCFGx.EPSIZE, USBHS_DEVEPTCFGx.EPDIR, USBHS_DEVEPTCFGx.EPTYPE) and the Data Toggle Sequence (USBHS_DEVEPTISRx.DTSEQ) field.
  Note: The interrupt sources located in USBHS_DEVEPTISRx are not cleared when a USB bus reset has been received.

The endpoint configuration remains active and the endpoint is still enabled.

The endpoint reset may be associated with a clear of the data toggle sequence as an answer to the CLEAR_FEATURE USB request. This can be achieved by writing a one to the Reset Data Toggle Set bit (RSTDTS) in the Device Endpoint x Control Set register (this sets the Reset Data Toggle bit USBHS_DEVEPTIMRx.RSTDT).

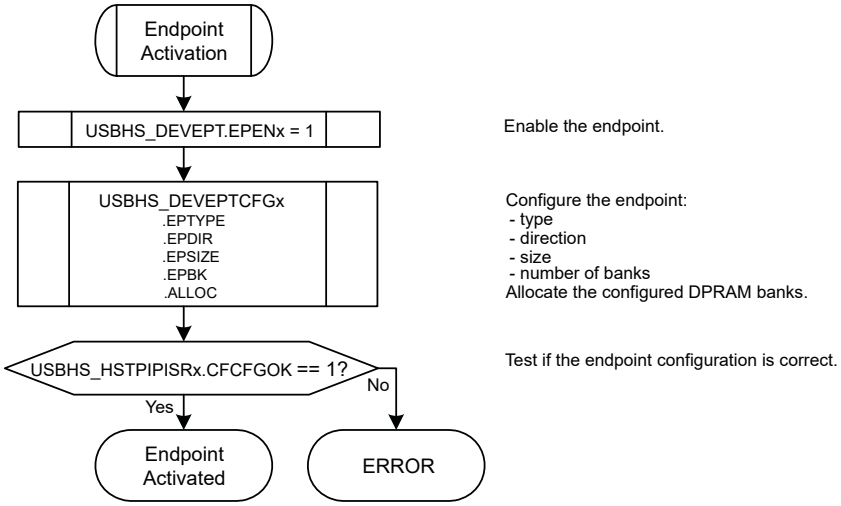
In the end, the user has to write a zero to the USBHS_DEVEPT.EPRSTx bit to complete the reset operation and to start using the FIFO.

#### 39.5.2.5 Endpoint Activation

The endpoint is maintained inactive and reset (see "Endpoint Reset" for more information) as long as it is disabled (USBHS_DEVEPT.EPENx = 0). USBHS_DEVEPTISRx.DTSEQ is also reset.

The algorithm represented in the following figure must be followed to activate an endpoint.

**Figure 39-8. Endpoint Activation Algorithm**




As long as the endpoint is not correctly configured (USBHS_HSTPIPISRx.CFGOK = 0), the controller does not acknowledge the packets sent by the host to this endpoint.

The USBHS_HSTPIPISRx.CFGOK bit is set provided that the configured size and number of banks are correct as compared to the endpoint maximal allowed values (see the Description of USB Pipes/Endpoints table) and to the maximal FIFO size (i.e., the DPRAM size).

See "DPRAM Management" for additional information.

#### 39.5.2.6 Address Setup

The USB device address is set up according to the USB protocol.

- After all kinds of resets, the USB device address is 0.

| Value | Name | Description |
|---|---|---|
| | | • for OUT endpoint, USBHS_DEVEPTIMRx.NBUSYBKE = 1 and all the banks are busy. |

**Bits 9:8 – DTSEQ[1:0]** Data Toggle Sequence
This field is set to indicate the PID of the current bank:

For IN transfers, it indicates the data toggle sequence that should be used for the next packet to be sent. This is not relative to the current bank.

For OUT transfers, this value indicates the last data toggle sequence received on the current bank.

By default, DTSEQ is 0b01, as if the last data toggle sequence was Data1, so the next sent or expected data toggle sequence should be Data0.

| Value | Name | Description |
|---|---|---|
| 0 | DATA0 | Data0 toggle sequence |
| 1 | DATA1 | Data1 toggle sequence |
| 2 | DATA2 | Reserved for high-bandwidth isochronous endpoint |
| 3 | MDATA | Reserved for high-bandwidth isochronous endpoint |

**Bit 7 – SHORTPACKET** Short Packet Interrupt

| Value | Description |
|---|---|
| 0 | Cleared when SHORTPACKETC = 1. This acknowledges the interrupt. |
| 1 | Set for non-control OUT endpoints, when a short packet has been received. This triggers a PEP_x interrupt if USBHS_DEVEPTIMRx.SHORTPACKETE = 1. |

**Bit 6 – STALLEDI** STALLed Interrupt

| Value | Description |
|---|---|
| 0 | Cleared when STALLEDIC = 1. This acknowledges the interrupt. |
| 1 | Set to signal that a STALL handshake has been sent. To do that, the software has to set the STALLRQ bit (by writing a one to the STALLRQS bit). This triggers a PEP_x interrupt if STALLEDE = 1. |

**Bit 5 – OVERFI** Overflow Interrupt
For all endpoint types, an overflow can occur during the OUT stage if the host attempts to write into a bank that is too small for the packet. The packet is acknowledged and the USBHS_DEVEPTISRx.RXOUTI bit is set as if no overflow had occurred. The bank is filled with all the first bytes of the packet that fit in.

| Value | Description |
|---|---|
| 0 | Cleared when the OVERFIC bit is written to one. This acknowledges the interrupt. |
| 1 | Set when an overflow error occurs. This triggers a PEP_x interrupt if OVERFE = 1. |

**Bit 4 – NAKINI** NAKed IN Interrupt

| Value | Description |
|---|---|
| 0 | Cleared when NAKINIC = 1. This acknowledges the interrupt. |
| 1 | Set when a NAK handshake has been sent in response to an IN request from the host. This triggers a PEP_x interrupt if NAKINE = 1. |

PCS = 1111 forbidden (no peripheral is selected)

(x = don't care)

If SPI_MR.PCSDEC = 1:

NPCS[3:0] output signals = PCS.

**Bit 7 – LLB** Local Loopback Enable
LLB controls the local loopback on the data shift register for testing in Master mode only (MISO is internally connected on MOSI).

| Value | Description |
|---|---|
| 0 | Local loopback path disabled. |
| 1 | Local loopback path enabled. |

**Bit 5 – WDRBT** Wait Data Read Before Transfer

| Value | Description |
|---|---|
| 0 | No Effect. In Master mode, a transfer can be initiated regardless of SPI_RDR state. |
| 1 | In Master mode, a transfer can start only if SPI_RDR is empty, i.e., does not contain any unread data. This mode prevents overrun error in reception. |

**Bit 4 – MODFDIS** Mode Fault Detection

| Value | Description |
|---|---|
| 0 | Mode fault detection enabled |
| 1 | Mode fault detection disabled |

**Bit 2 – PCSDEC** Chip Select Decode
When PCSDEC = 1, up to 15 chip select signals can be generated with the four NPCS lines using an external 4-bit to 16-bit decoder. The chip select registers define the characteristics of the 15 chip selects, with the following rules:

SPI_CSR0 defines peripheral chip select signals 0 to 3.

SPI_CSR1 defines peripheral chip select signals 4 to 7.

SPI_CSR2 defines peripheral chip select signals 8 to 11.

SPI_CSR3 defines peripheral chip select signals 12 to 14.

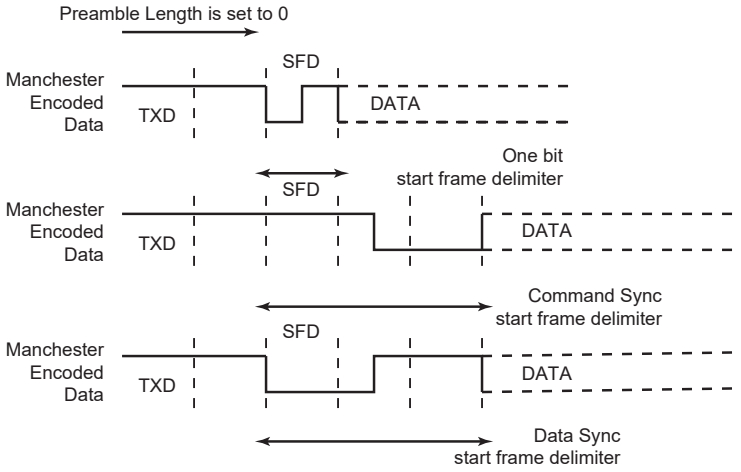
| Value | Description |
|---|---|
| 0 | The chip select lines are directly connected to a peripheral device. |
| 1 | The four NPCS chip select lines are connected to a 4-bit to 16-bit decoder. |

**Bit 1 – PS** Peripheral Select

| Value | Description |
|---|---|
| 0 | Fixed Peripheral Select |
| 1 | Variable Peripheral Select |

**Bit 0 – MSTR** Master/Slave Mode

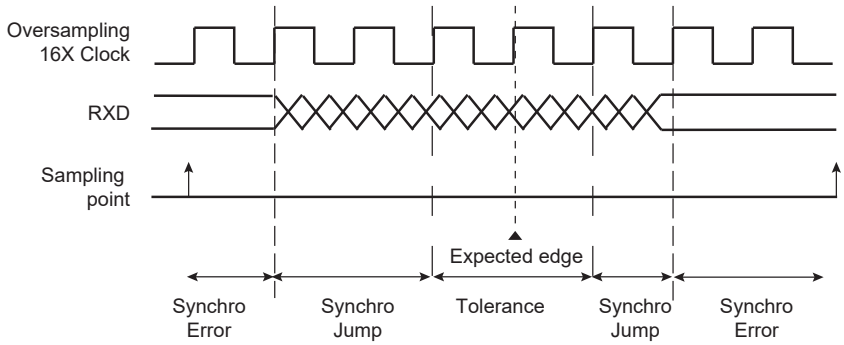
**Figure 46-9. Start Frame Delimiter**




#### 46.6.3.2.1 Drift Compensation

Drift compensation is available only in 16X Oversampling mode. A hardware recovery system allows a larger clock drift. To enable the hardware system, USART_MAN.DRIFT must be written to ‘1’. If the RXD edge is one 16X clock cycle from the expected edge, this is considered as normal jitter and no corrective action is taken. If the RXD event is between 4 and 2 clock cycles before the expected edge, then the current period is shortened by one clock cycle. If the RXD event is between 2 and 3 clock cycles after the expected edge, then the current period is lengthened by one clock cycle. These intervals are considered to be drift and so corrective actions are automatically taken.

**Figure 46-10. Bit Resynchronization**




#### 46.6.3.3 Asynchronous Receiver

If the USART is programmed in Asynchronous operating mode (SYNC = 0), the receiver oversamples the RXD input line. The oversampling is either 16 or 8 times the baud rate clock, depending on the value of US_MR.OVER.

The receiver samples the RXD line. If the line is sampled during one-half of a bit time to 0, a start bit is detected and data, parity and stop bits are successively sampled on the bit rate clock.

If the oversampling is 16 (OVER = 0), a start is detected at the eighth sample to 0. Data bits, parity bit and stop bit are assumed to have a duration corresponding to 16 oversampling clock cycles. If the oversampling is 8 (OVER = 1), a start bit is detected at the fourth sample to 0. Data bits, parity bit and stop bit are assumed to have a duration corresponding to 8 oversampling clock cycles.

The number of data bits, first bit sent and Parity mode are selected by the same fields and bits as the transmitter, i.e., respectively CHRL, MODE9, MSBF and PAR. For the synchronization mechanism only,

| | | | | |
|---|---|---|---|---|
| 0 | Baud Rate Clock Disabled | | | |
| 1 to 65535 | CD = Selected Clock / (16 × Baud Rate) | CD = Selected Clock / (8 × Baud Rate) | CD = Selected Clock / Baud Rate | CD = Selected Clock / (FI_DI_RATIO × Baud Rate) |

### 49.6.27 MCAN Receive FIFO 0 Configuration

**Name:** MCAN_RXF0C
**Offset:** 0xA0
**Reset:** 0x00000000
**Property:** Read/Write

This register can only be written if the bits CCE and INIT are set in MCAN CC Control Register.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | F0OM | F0WM[6:0] | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | | F0S[6:0] | | | | | | |
| Access | | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | F0SA[13:6] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | F0SA[5:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | | |

**Bit 31 – F0OM** FIFO 0 Operation Mode
FIFO 0 can be operated in Blocking or in Overwrite mode (see Rx FIFOs).

| Value | Description |
|---|---|
| 0 | FIFO 0 Blocking mode. |
| 1 | FIFO 0 Overwrite mode. |

**Bits 30:24 – F0WM[6:0]** Receive FIFO 0 Watermark

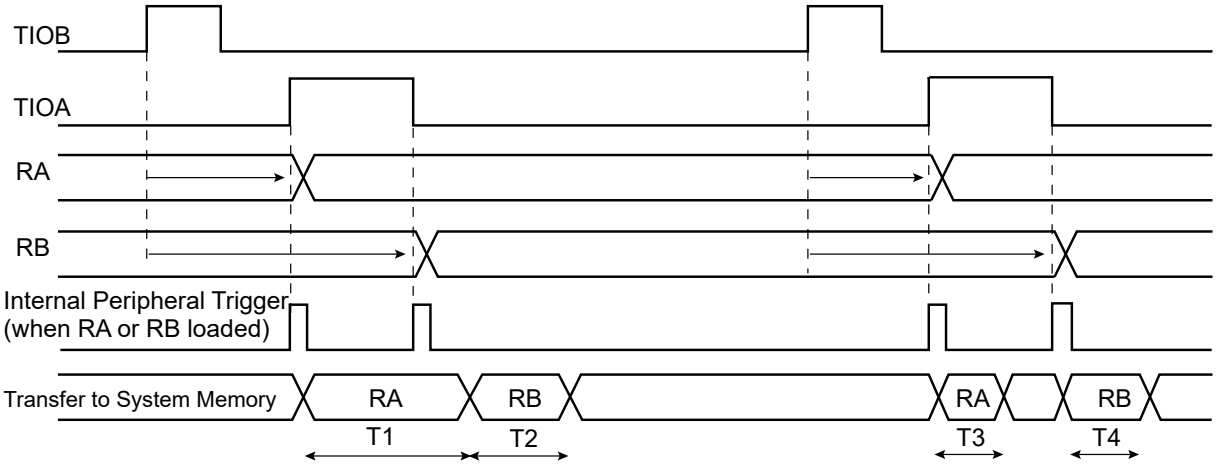
| Value | Description |
|---|---|
| 0 | Watermark interrupt disabled. |
| 1-64 | Level for Receive FIFO 0 watermark interrupt (MCAN_IR.RF0W). |
| >64 | Watermark interrupt disabled. |

**Bits 22:16 – F0S[6:0]** Receive FIFO 0 Size
The Receive FIFO 0 elements are indexed from 0 to F0S-1.

| Value | Description |
|---|---|
| 0 | No Receive FIFO 0 |
| 1-64 | Number of Receive FIFO 0 elements. |
| >64 | Values greater than 64 are interpreted as 64. |

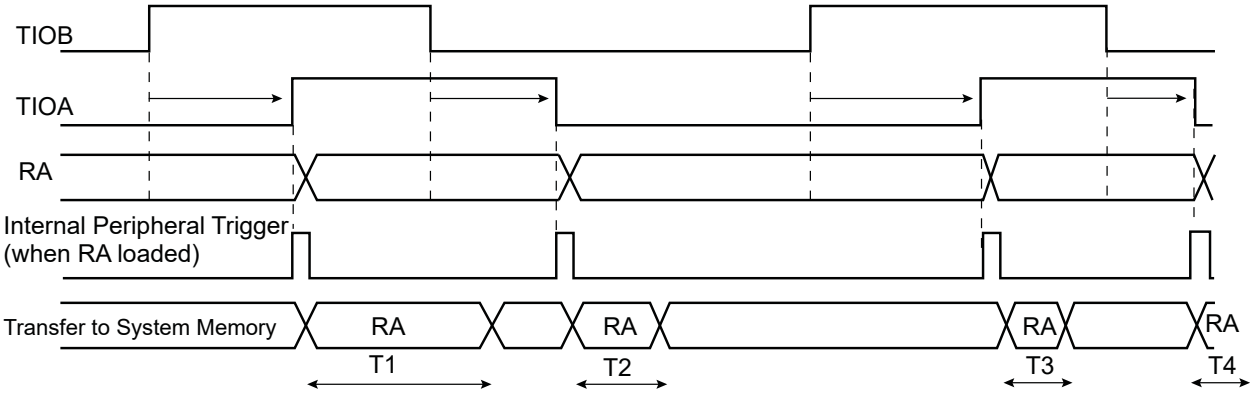
**Figure 50-5. Example of Transfer with DMAC in Capture Mode**

ETRGEDG = 1, LDRA = 1, LDRB = 2, ABETRG = 0




T1,T2,T3,T4 = System Bus load dependent ($t_{min}$ = 8 Peripheral Clocks)

ETRGEDG = 3, LDRA = 3, LDRB = 0, ABETRG = 0




T1,T2,T3,T4 = System Bus load dependent ($t_{min}$ = 8 Peripheral Clocks)

### 50.6.10 Trigger Conditions

In addition to the SYNC signal, the software trigger and the RC compare trigger, an external trigger can be defined.

The ABETRG bit in the TC_CMR selects TIOAx or TIOBx input signal as an external trigger or the trigger signal from the output comparator of the PWM module. The External Trigger Edge Selection parameter (ETRGEDG field in TC_CMR) defines the edge (rising, falling, or both) detected to generate an external trigger. If ETRGEDG = 0 (none), the external trigger is disabled.

**Figure 50-8. WAVSEL = 00 without Trigger**

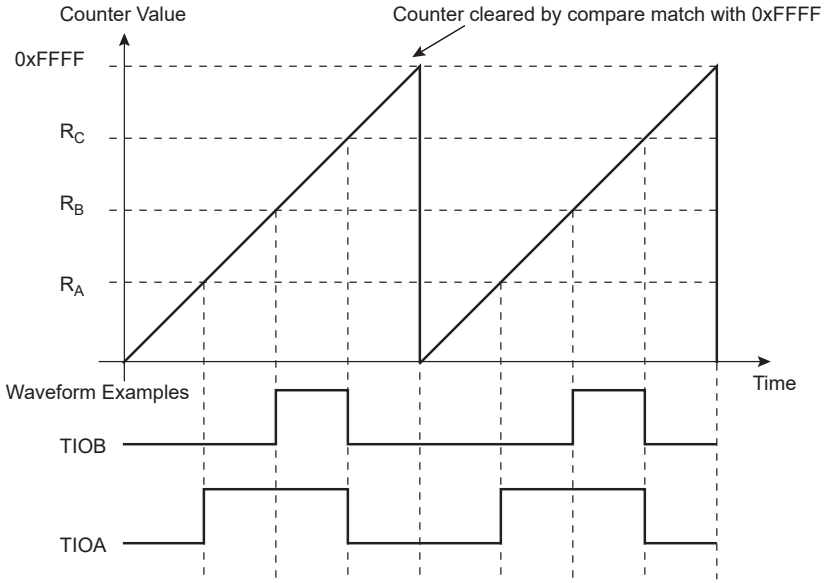
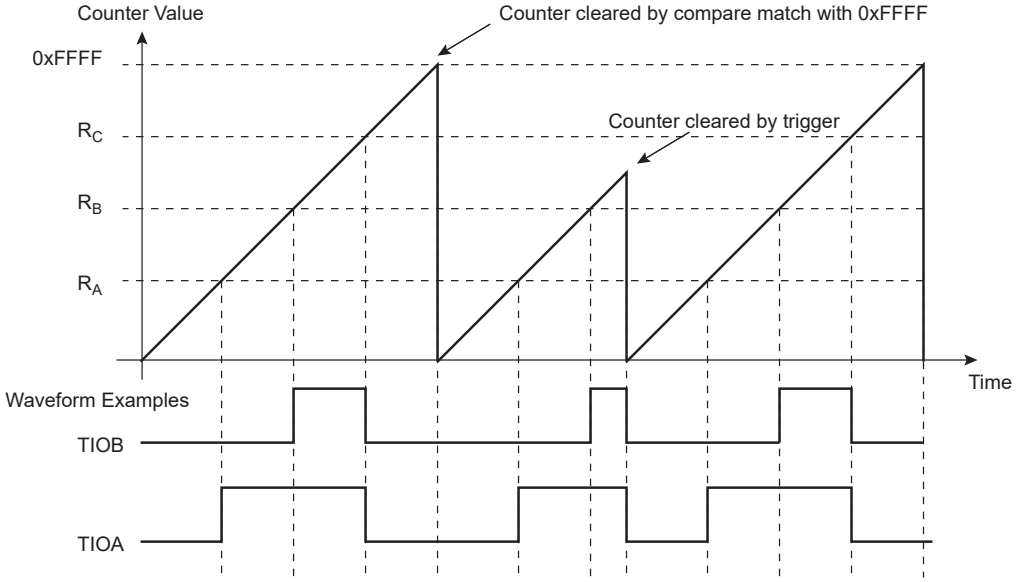



**Figure 50-9. WAVSEL = 00 with Trigger**




#### 50.6.12.2 WAVSEL = 10

When WAVSEL = 10, the value of TC_CV is incremented from 0 to the value of RC, then automatically reset on a RC Compare. Once the value of TC_CV has been reset, it is then incremented and so on.

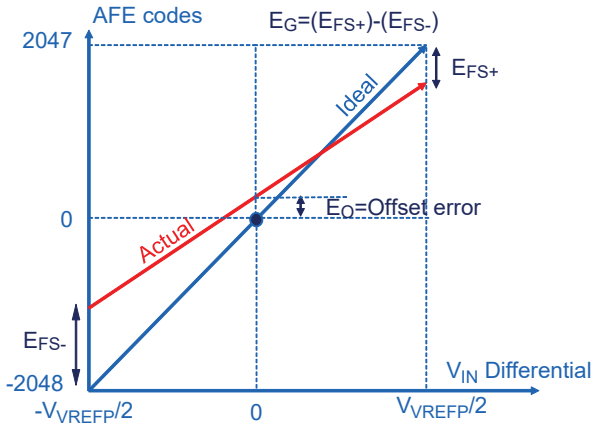
It is important to note that TC_CV can be reset at any time by an external event or a software trigger if both are programmed correctly.

Refer to the figures below.

In addition, RC Compare can stop the counter clock (CPCSTOP = 1 in TC_CMR) and/or disable the counter clock (CPCDIS = 1 in TC_CMR).

| Offset | Name | Bit Pos. | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | 23:16 | CNT[23:16] | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x0278 | PWM_DT3 | 7:0 | DTH[7:0] | | | | | | | |
| | | 15:8 | DTH[15:8] | | | | | | | |
| | | 23:16 | DTL[7:0] | | | | | | | |
| | | 31:24 | DTL[15:8] | | | | | | | |
| 0x027C | PWM_DTUPD3 | 7:0 | DTHUPD[7:0] | | | | | | | |
| | | 15:8 | DTHUPD[15:8] | | | | | | | |
| | | 23:16 | DTLUPD[7:0] | | | | | | | |
| | | 31:24 | DTLUPD[15:8] | | | | | | | |
| 0x0280 ... 0x03FF | Reserved | | | | | | | | | |
| 0x0400 | PWM_CMUPD0 | 7:0 | | | | | | | | |
| | | 15:8 | | | CPOLINVUP | | | | CPOLUP | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x0404 ... 0x041F | Reserved | | | | | | | | | |
| 0x0420 | PWM_CMUPD1 | 7:0 | | | | | | | | |
| | | 15:8 | | | CPOLINVUP | | | | CPOLUP | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x0424 ... 0x042B | Reserved | | | | | | | | | |
| 0x042C | PWM_ETRG1 | 7:0 | MAXCNT[7:0] | | | | | | | |
| | | 15:8 | MAXCNT[15:8] | | | | | | | |
| | | 23:16 | MAXCNT[23:16] | | | | | | | |
| | | 31:24 | RFEN | TRGSRC | TRGFILT | TRGEDGE | | | TRGMODE[1:0] | |
| 0x0430 | PWM_LEBR1 | 7:0 | | LEBDELAY[6:0] | | | | | | |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | PWMHREN | PWMHFEN | PWMLREN | PWMLFEN |
| | | 31:24 | | | | | | | | |
| 0x0434 ... 0x043F | Reserved | | | | | | | | | |
| 0x0440 | PWM_CMUPD2 | 7:0 | | | | | | | | |
| | | 15:8 | | | CPOLINVUP | | | | CPOLUP | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x0444 ... 0x044B | Reserved | | | | | | | | | |
| 0x044C | PWM_ETRG2 | 7:0 | MAXCNT[7:0] | | | | | | | |

**Figure 58-13. Gain and Offset Errors in Differential Mode**




where:

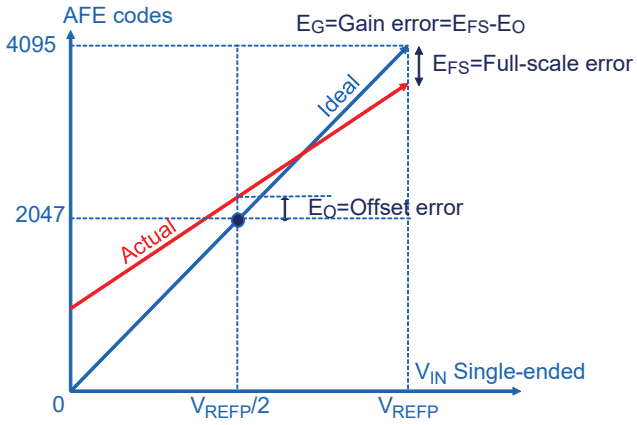
- Full-scale error $E_{FS}$ =($E_{FS+}$)-($E_{FS-}$), unit is LSB code
- Offset error $E_O$ is the offset error measured for $V_{IN}$=0V
- Gain error $E_G$=100 × $E_{FS}$ /4096, unit in %

The error values in the tables below include the sample-and-hold error as well as the PGA gain error.

#### 58.8.4.4.2 Single-ended Mode

The figure below illustrates the AFE output code relative to an input voltage $V_{IN}$ between 0V (Ground) and $V_{VREFP}$. The AFE is configured in Single-ended mode by connecting internally the negative differential input to $V_{VREFP}$/2. As the AFE continues to work internally in Differential mode, the offset is measured at $V_{VREFP}$/2. The offset at $V_{INP}$=0 can be computed using the transfer function and the corresponding $E_G$ and $E_O$.

**Figure 58-14. Gain and Offset Errors in Single-ended Mode**




where:

- Full-scale error $E_{FS}$ =($E_{FS+}$)-($E_{FS-}$), unit is LSB code
- Offset error $E_O$ is the offset error measured for $V_{REFP/2}$= 0V
- Gain error $E_G$=100 x $E_{FS}$ /4096, unit in %

The error values in the tables below include the DAC, the sample-and-hold error as well as the PGA gain error.

converted to a gain error by the AFE. The noise generated by $V_{VREFP}$ is converted by the AFE to count noise.

**Table 59-30. VREFP Electrical Characteristics**

| Symbol | Parameter | Conditions | Min | Typ | Max | Unit |
|---|---|---|---|---|---|---|
| $V_{VREFP}$ | Voltage Range | Full operational | 1.7 | – | VDDIN | V |
| | RMS Noise (see **Note 2**) | Bandwidth up to 1.74MHz VREFP=1.7V | – | – | 120 | µV |
| $R_{VREFP}$ | Input DC Impedance | AFE reference resistance bridge (see **Note 1**) | – | 4.7 | – | kOhm |
| Vin | Input Linear Range (see **Note 3**) | Operational Range | 2 | - | 98 | %VVREFP |
| $I_{VREFP}$ | Current | $V_{VREFP}$ = 3.3V | – | 0.8 | – | mA |

**Note:**

1. When the AFE is in Sleep mode, the VREFP impedance has a minimum of 10 MOhm.
2. Requested noise on VREFP.
3. Electrical parameters specified inside the operational range. Exceeding this range can introduce additional INL error up to +/- 5 LSB and temperature dependency up to +/-10 LSB.

### 59.8.3 AFE Timings

**Table 59-31. AFE Timing Characteristics**

| Symbol | Parameter | Conditions | Min | Typ | Max | Unit |
|---|---|---|---|---|---|---|
| $f_{AFE\ Clock}$ | Clock Frequency | – | 4 | 20 | 40 | MHz |
| $t_{AFE\ Clock}$ | Clock Period | – | 25 | 50 | 250 | ns |
| $f_S$ | Sampling Frequency (see **Note 1**) | – | – | – | 1.74 | MHz |
| $t_{START}$ | AFE Startup Time | Sleep mode to Normal mode | – | – | 4 | µs |
| | | Fast Wake-up mode to Normal mode | - | - | 2 | µs |

**Note:**

1. $f_s = 1 / t_{AFE\_conv}$ in Free Run mode; otherwise defined by the trigger timing.

### 59.8.4 AFE Transfer Function

The first operation of the AFE is a sampling function relative to $V_{DAC}$. $V_{DAC}$ is generated by an internal DAC0 or DAC1. All operations after the Sample-and-Hold are differential relative to an internal common mode voltage $V_{CM} = V_{VREFP}/2$.

In Differential mode, the Sample-and-Hold common mode voltage is equal to $V_{DAC} = V_{VREFP}/2$ (set by software DAC0 and DAC1 to code 512).

In Single-ended mode, $V_{DAC}$ is the common mode voltage. $V_{DAC}$ is the output of DAC0 or DAC1 voltage. All operations after the Sample-and-Hold are differential, including those in Single-ended mode.

For the formula example, the internal DAC0 or DAC1 is set for the code 512.