Welcome to E-XFL.COM

Understanding Embedded - FPGAs (Field Programmable Gate Array)

Embedded - FPGAs, or Field Programmable Gate Arrays, are advanced integrated circuits that offer unparalleled flexibility and performance for digital systems. Unlike traditional fixed-function logic devices, FPGAs can be programmed and reprogrammed to execute a wide array of logical operations, enabling customized functionality tailored to specific applications. This reprogrammability allows developers to iterate designs quickly and implement complex functions without the need for custom hardware.

Applications of Embedded - FPGAs

The versatility of Embedded - FPGAs makes them indispensable in numerous fields. In telecommunications.

Details	
Product Status	Active
Number of LABs/CLBs	-
Number of Logic Elements/Cells	-
Total RAM Bits	36864
Number of I/O	114
Number of Gates	250000
Voltage - Supply	1.425V ~ 1.575V
Mounting Type	Surface Mount
Operating Temperature	0°C ~ 85°C (TJ)
Package / Case	256-LBGA
Supplier Device Package	256-FPBGA (17x17)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/afs250-fgg256

Embedded Memories

Flash Memory Blocks

The flash memory available in each Fusion device is composed of one to four flash blocks, each 2 Mbits in density. Each block operates independently with a dedicated flash controller and interface. Fusion flash memory blocks combine fast access times (60 ns random access and 10 ns access in Read-Ahead mode) with a configurable 8-, 16-, or 32-bit datapath, enabling high-speed flash operation without wait states. The memory block is organized in pages and sectors. Each page has 128 bytes, with 33 pages comprising one sector and 64 sectors per block. The flash block can support multiple partitions. The only constraint on size is that partition boundaries must coincide with page boundaries. The flexibility and granularity enable many use models and allow added granularity in programming updates.

Fusion devices support two methods of external access to the flash memory blocks. The first method is a serial interface that features a built-in JTAG-compliant port, which allows in-system programmability during user or monitor/test modes. This serial interface supports programming of an AES-encrypted stream. Data protected with security measures can be passed through the JTAG interface, decrypted, and then programmed in the flash block. The second method is a soft parallel interface.

FPGA logic or an on-chip soft microprocessor can access flash memory through the parallel interface. Since the flash parallel interface is implemented in the FPGA fabric, it can potentially be customized to meet special user requirements. For more information, refer to the CoreCFI Handbook. The flash memory parallel interface provides configurable byte-wide (×8), word-wide (×16), or dual-word-wide (×32) data-port options. Through the programmable fl ash parallel interface, the on-chip and off-chip memories can be cascaded for wider or deeper configurations.

The flash memory has built-in security. The user can configure either the entire flash block or the small blocks to protect against unintentional or intrusive attempts to change or destroy the storage contents. Each on-chip flash memory block has a dedicated controller, enabling each block to operate independently.

The flash block logic consists of the following sub-blocks:

Flash block Contains all stored data. The flash block contains 64 sectors and each sector contains 33 pages of data.

Page Buffer Contains the contents of the current page being modified. A page contains 8 blocks of data.

Block Buffer Contains the contents of the last block accessed. A block contains 128 data bits.

ECC Logic The flash memory stores error correction information with each block to perform single-bit error correction and double-bit error detection on all data blocks.

User Nonvolatile FlashROM

In addition to the flash blocks, Fusion devices have 1 Kbit of user-accessible, nonvolatile FlashROM on-chip. The FlashROM is organized as 8×128-bit pages. The FlashROM can be used in diverse system applications:

Internet protocol addressing (wireless or fixed)

System calibration settings

Device serialization and/or inventory control

Subscription-based business models (for example, set-top boxes)

Secure key storage for communications algorithms protected by security

Asset management/tracking

Date stamping

Version management

The FlashROM is written using the standard IEEE 1532 JTAG programming interface. Pages can be individually programmed (erased and written). On-chip AES decryption can be used selectively over public networks to load data such as security keys stored in the FlashROM for a user design.

The FlashROM can be programmed (erased and written) via the JTAG programming interface, and its contents can be read back either through the JTAG programming interface or via direct FPGA core addressing.

Global Clocking

Fusion devices have extensive support for multiple clocking domains. In addition to the CCC and PLL support described above, there are on-chip oscillators as well as a comprehensive global clock distribution network.

The integrated RC oscillator generates a 100 MHz clock. It is used internally to provide a known clock source to the flash memory read and write control. It can also be used as a source for the PLLs.

The crystal oscillator supports the following operating modes:

Crystal (32.768 KHz to 20 MHz)

Ceramic (500 KHz to 8 MHz)

RC (32.768 KHz to 4 MHz)

Each VersaTile input and output port has access to nine VersaNets: six main and three quadrant global networks. The VersaNets can be driven by the CCC or directly accessed from the core via MUXes. The VersaNets can be used to distribute low-skew clock signals or for rapid distribution of high-fanout nets.

Digital I/Os with Ad vanced I/O Standards

The Fusion family of FPGAs features a flexible digital I/O structure, supporting a range of voltages (1.5 V, 1.8 V, 2.5 V, and 3.3 V). Fusion FPGAs support many different digital I/O standards, both single-ended and differential.

The I/Os are organized into banks, with four or five banks per device. The configuration of these banks determines the I/O standards supported. The banks along the east and west sides of the device support the full range of I/O standards (single-ended and differential). The south bank supports the Analog Quads (analog I/O). In the family's two smaller devices, the north bank supports multiple single-ended digital I/O standards. In the family s larger devices, the north bank is divided into two banks of digital Pro I/Os, supporting a wide variety of single-ended, differential, and voltage-referenced I/O standards.

Each I/O module contains several input, output, and enable registers. These registers allow the implementation of the following applications:

Single-Data-Rate (SDR) applications

Double-Data-Rate (DDR) applications DDR LVDS I/O for chip-to-chip communications

Fusion banks support LVPECL, LVDS, BLVDS, and M-LVDS with 20 multi-drop points.

VersaTiles

The Fusion core consists of VersaTiles, which are also used in the successful ProASIC3 family. The Fusion VersaTile supports the following:

All 3-input logic functions LUT-3 equivalent

Latch with clear or set

D-flip-flop with clear or set and optional enable

Refer to Figure 1-2 for the VersaTile configuration arrangement.

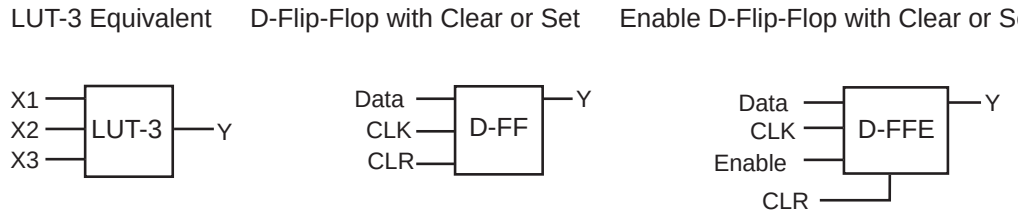


Figure 1-2 VersaTile Configurations

Array Coordinates

During many place-and-route operations in the Microsemi Designer software tool, it is possible to set constraints that require array coordinates. Table 2-3 is provided as a reference. The array coordinates are measured from the lower left (0, 0). They can be used in region constraints for specific logic groups/blocks, designated by a wildcard, and can contain core cells, memories, and I/Os.

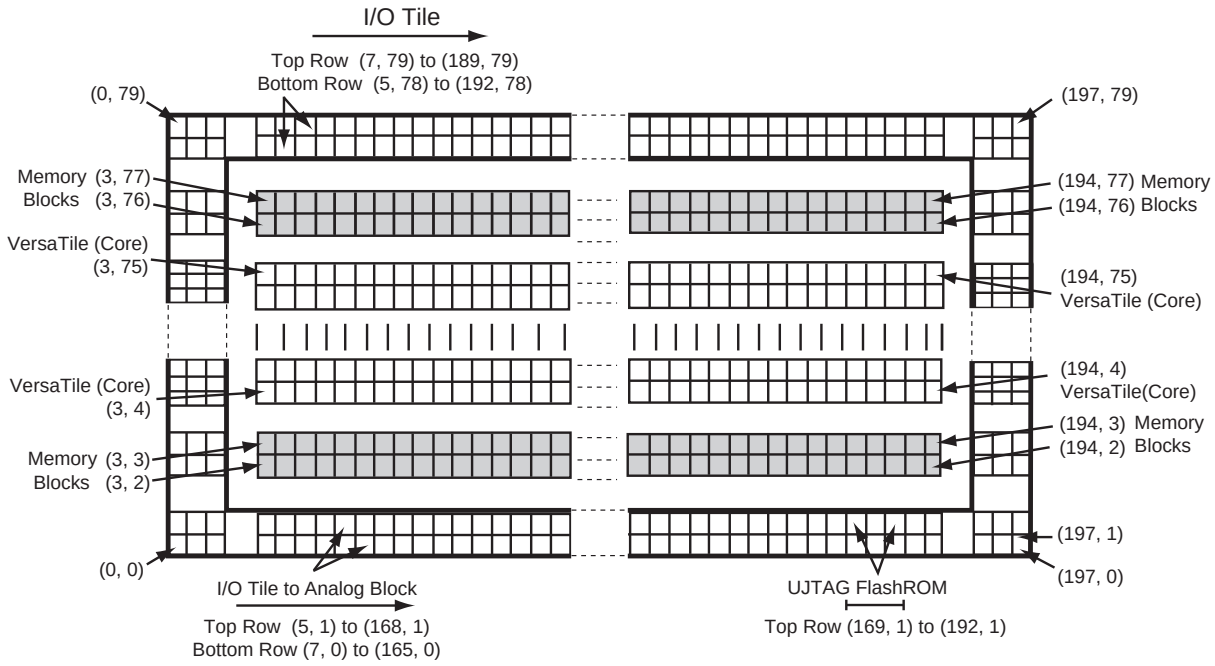
Table 2-3 provides array coordinates of core cells and memory blocks.

I/O and cell coordinates are used for placement constraints. Two coordinate systems are needed because there is not a one-to-one correspondence between I/O cells and edge core cells. In addition, the I/O coordinate system changes depending on the die/package combination. It is not listed in Table 2-3. The Designer ChipPlanner tool provides array coordinates of all I/O locations. I/O and cell coordinates are used for placement constraints. However, I/O placement is easier by package pin assignment.

Figure 2-7 illustrates the array coordinates of an AFS600 device. For more information on how to use array coordinates for region/placement constraints, see the Designer User's Guide or online help (available in the software) for Fusion software tools.

Table 2-3 Array Coordinates

Device	VersaTiles				Memory Rows		All	
	Min.		Max.		Bottom	Top	Min.	Max.
	x	y	x	y	(x, y)	(x, y)	(x, y)	(x, y)
AFS090	3	2	98	25	None	(3, 26)	(0, 0)	(101, 29)
AFS250	3	2	130	49	None	(3, 50)	(0, 0)	(133, 53)
AFS600	3	4	194	75	(3, 2)	(3, 76)	(0, 0)	(197, 79)
AFS1500	3	4	322	123	(3, 2)	(3, 124)	(0, 0)	(325, 129)



Note: The vertical I/O tile coordinates are not shown. West side coordinates are {(0, 2) to (2, 2)} to {(0, 77) to (2, 77)}; east side coordinates are {(195, 2) to (197, 2)} to {(195, 77) to (197, 77)}.

Figure 2-7 Array Coordinates for AFS600

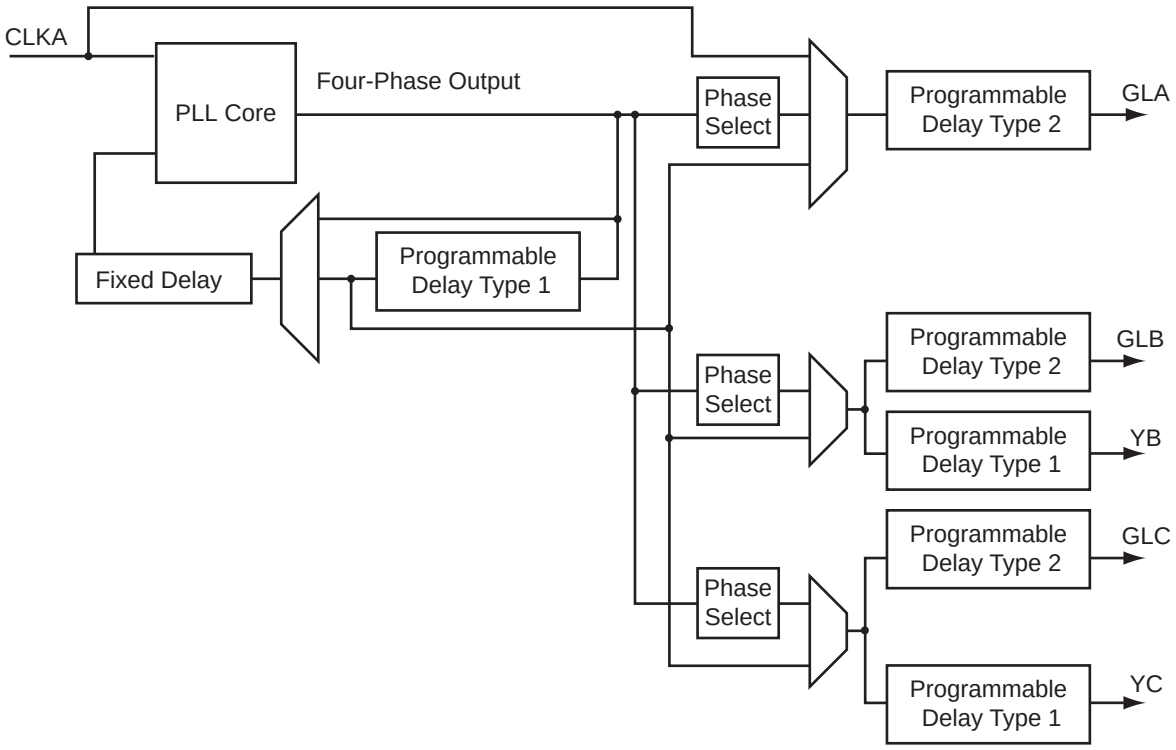
CCC Physical Implementation

The CCC circuit is composed of the following (Figure 2-23):

- PLL core
- 3 phase selectors
- 6 programmable delays and 1 fixed delay
- 5 programmable frequency dividers that provide frequency multiplication/division (not shown in Figure 2-23 because they are automatically configured based on the user's required frequencies)
- 1 dynamic shift register that provides CCC dynamic reconfiguration capability (not shown)

CCC Programming

The CCC block is fully configurable. It is configured via static flash configuration bits in the array, set by the user in the programming bitstream, or configured through an asynchronous dedicated shift register, dynamically accessible from inside the Fusion device. The dedicated shift register permits changes of parameters such as PLL divide ratios and delays during device operation. This latter mode allows the user to dynamically reconfigure the PLL without the need for core programming. The register file is accessed through a simple serial interface.



Note: Clock divider and multiplier blocks are not shown in this figure or in SmartGen. They are automatically configured based on the user's required frequencies.

Figure 2-23 PLL Block

Program Operation

A Program operation is initiated by asserting the PROGRAM signal on the interface. Program operations save the contents of the Page Buffer to the FB Array. Due to the technologies inherent in the FB, the total programming (including erase) time per page of the eNVM is 6.8 ms. While the FB is writing the data to the array, the BUSY signal will be asserted.

During a Program operation, the sector and page addresses on ADDR are compared with the stored address for the page (and sector) in the Page Buffer. If there is a mismatch between the two addresses, the Program operation will be aborted and an error will be reported on the STATUS output.

It is possible to write the Page Buffer to a different page in memory. When asserting the PROGRAM pin, if OVERWRITEPAGE is asserted as well, the FB will write the contents of the Page Buffer to the sector and page designated on the ADDR inputs if the destination page is not Overwrite Protected.

A Program operation can be utilized to either modify the contents of the page in the flash memory block or change the protections for the page. Setting the OVERWRITEPROTECT bit on the interface while asserting the PROGRAM pin will put the page addressed into Overwrite Protect Mode. Overwrite Protect Mode safeguards a page from being inadvertently overwritten during subsequent Program or Erase operations.

Program operations that result in a STATUS value of ’01’ do not modify the addressed page. For all other values of STATUS, the addressed page is modified. Program errors include the following:

1. Attempting to program a page that is Overwrite Protected (STATUS = ’01’)
2. Attempting to program a page that is not in the Page Buffer when the Page Buffer has entered Page Loss Protection Mode (STATUS = ’01’)
3. Attempting to perform a program with OVERWRITEPAGE set when the page addressed has been Overwrite Protected (STATUS = ’01’)
4. The Write Count of the page programmed exceeding the Write Threshold defined in the part specification (STATUS = ’11’)
5. The ECC Logic determining that there is an uncorrectable error within the programmed page (STATUS = ’10’)
6. Attempting to program a page that is not in the Page Buffer when OVERWRITEPAGE is not set and the page in the Page Buffer is modified (STATUS = ’01’)
7. Attempting to program the page in the Page Buffer when the Page Buffer is not modified

The waveform for a Program operation is shown in Figure 2-36.

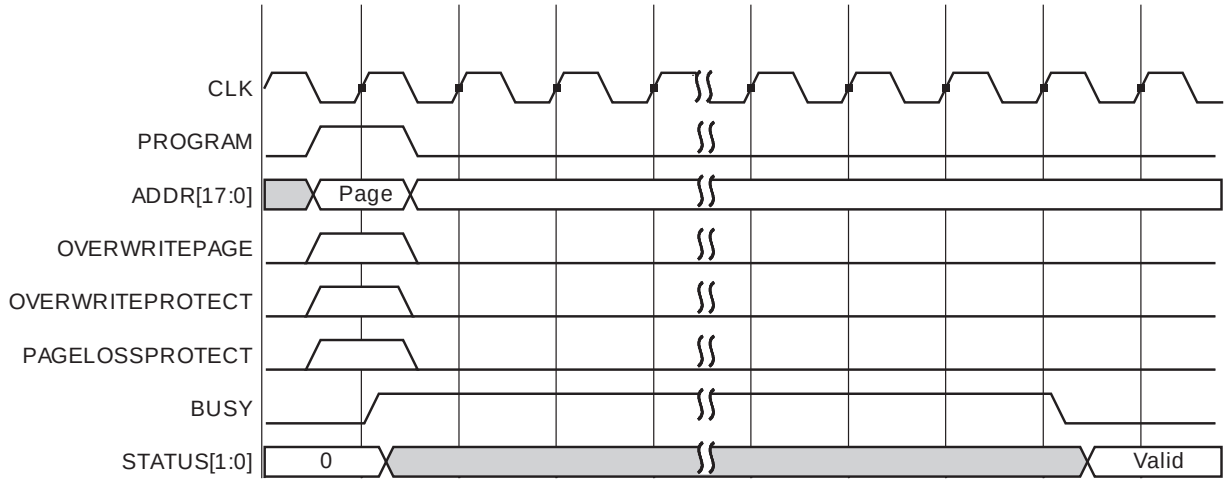


Figure 2-36 FB Program Waveform

Note: OVERWRITEPAGE is only sampled when the PROGRAM or ERASEPAGE pins are asserted. OVERWRITEPAGE is ignored in all other operations.

Read Operation

Read operations are designed to read data from the FB Array, Page Buffer, Block Buffer, or status registers. Read operations support a normal read and a read-ahead mode (done by asserting READNEXT). Also, the timing for Read operations is dependent on the setting of PIPE.

The following diagrams illustrate representative timing for Non-Pipe Mode (Figure 2-38) and Pipe Mode (Figure 2-39) reads of the flash memory block interface.

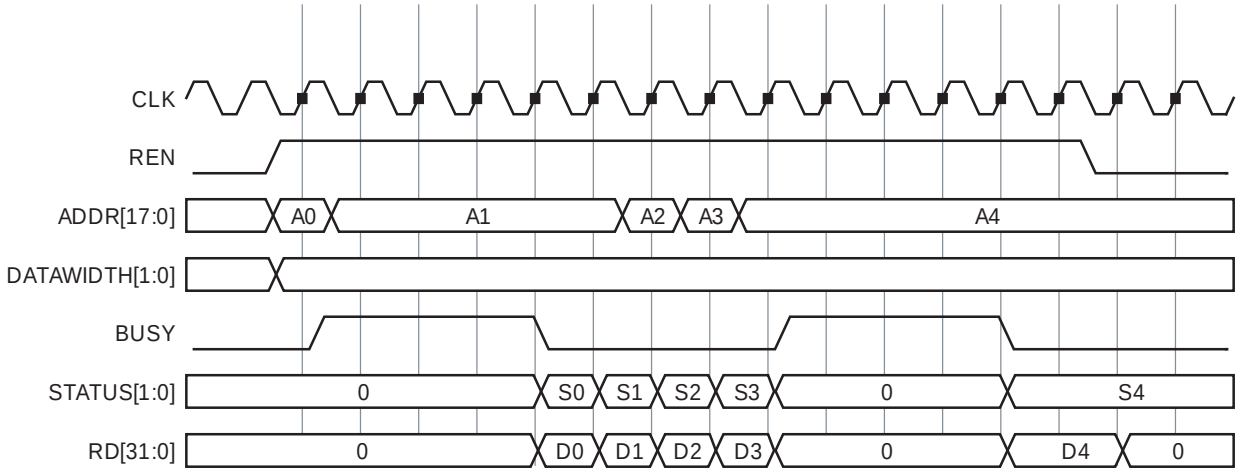


Figure 2-38 Read Waveform (Non-Pipe Mode, 32-bit access)

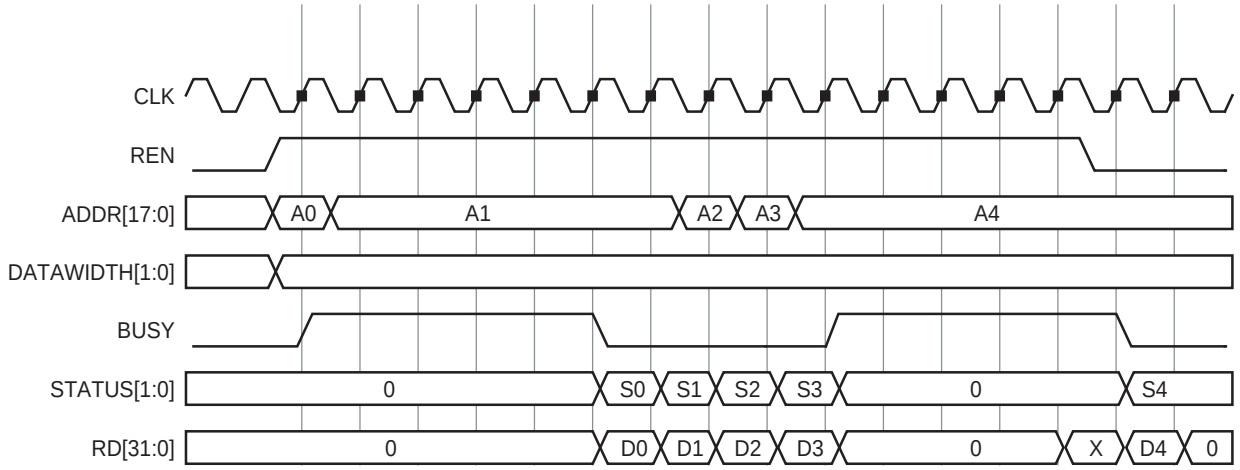


Figure 2-39 Read Waveform (Pipe Mode, 32-bit access)

Analog Block

With the Fusion family, Microsemi has introduced the world's first mixed-mode FPGA solution. Supporting a robust analog peripheral mix, Fusion devices will support a wide variety of applications. It is this Analog Block that separates Fusion from all other FPGA solutions on the market today.

By combining both flash and high-speed CMOS processes in a single chip, these devices offer the best of both worlds. The high-performance CMOS is used for building RAM resources. These high-performance structures support device operation up to 350 MHz. Additionally, the advanced Microsemi 0.13 µm flash process incorporates high-voltage transi stors and a high-isolation, triple-well process. Both of these are suited for the flash-based programmable logic and nonvolatile memory structures.

High-voltage transistors support the integration of analog technology in several ways. They aid in noise immunity so that the analog portions of the chip can be better isolated from the digital portions, increasing analog accuracy. Because they support high voltages, Microsemi flash FPGAs can be connected directly to high-voltage input signals, eliminating the need for external resistor divider networks, reducing component count, and increasing accuracy. By supporting higher internal voltages, the Microsemi advanced flash process enables high dynamic range on analog circuitry, increasing precision and signal noise ratio. Microsemi flash FP GAs also drive high-voltage outputs, eliminating the need for external level shifters and drivers.

The unique triple-well process enables the integration of high-performance analog features with increased noise immunity and better isolation. By increasing the efficiency of analog design, the triple-well process also enables a smaller overall design size, reducing die size and cost.

The Analog Block consists of the Analog Quad I/O structure, RTC (for details refer to the "Real-Time Counter System" section on page 2-31), ADC, and ACM. All of these elements are combined in the single Analog Block macro, with which the user implements this functionality (Figure 2-64).

The Analog Block needs to be reset/reinitialized after the core powers up or the device is programmed. An external reset/initialize signal, which can come from the internal voltage regulator when it powers up, must be applied.

Typical scaling factors are given in Table 2-57 on page 2-130, and the gain error (which contributes to the minimum and maximum) is in Table 2-49 on page 2-117.

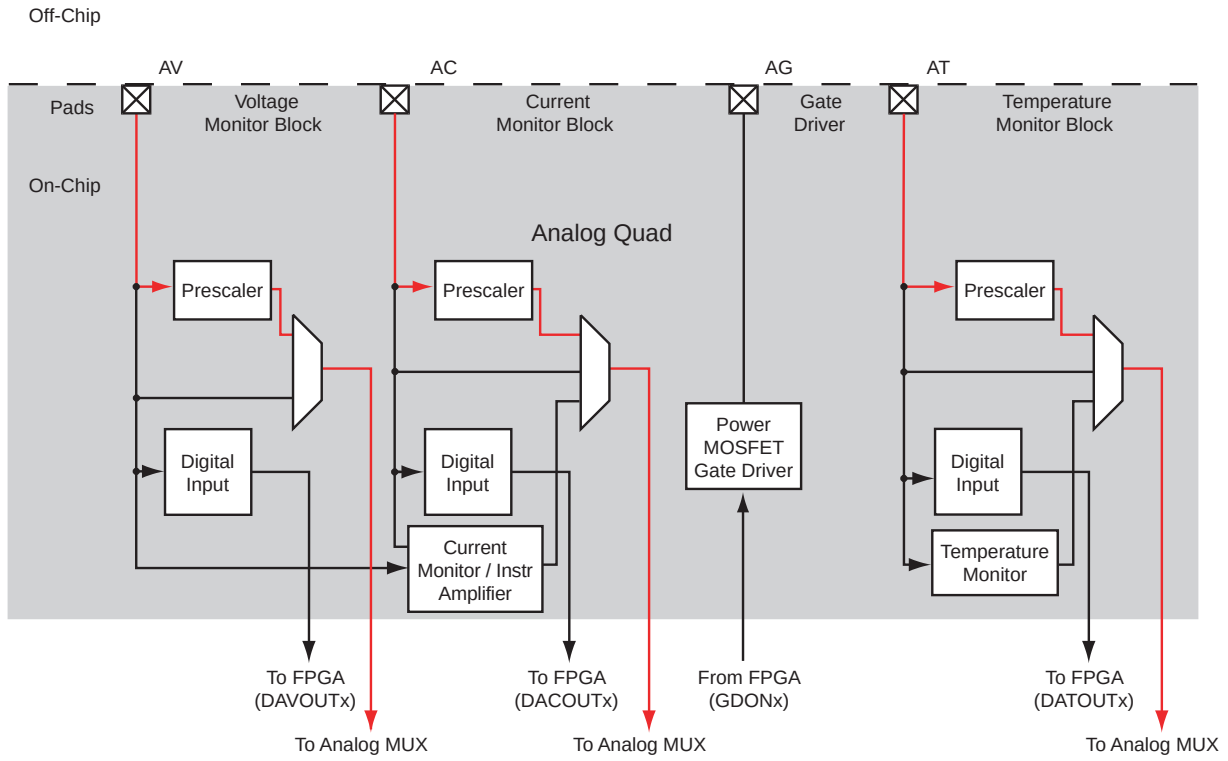


Figure 2-67 Analog Quad Prescaler Input Configuration

Terminology

BW Bandwidth

BW is a range of frequencies that a Channel can handle.

Channel

A channel is define as an analog input configured as one of the Prescaler range shown in Table 2-57 on page 2-130. The channel includes the Prescaler circuit and the ADC.

Channel Gain

Channel Gain is a measured of the deviation of the actual slope from the ideal slope. The slope is measured from the 20% and 80% point.

$$\text{Gain} = \frac{\text{Gain}_{\text{actual}}}{\text{Gain}_{\text{ideal}}}$$

EQ 1

Channel Gain Error

Channel Gain Error is a deviation from the ideal slope of the transfer function. The Prescaler Gain Error is expressed as the percent difference between the actual and ideal, as shown in EQ 2.

$$\text{Error}_{\text{Gain}} = (1\text{-Gain}) \times 100\%$$

EQ 2

Gain Error

The gain error of an ADC indicates how well the slope of an actual transfer function matches the slope of the ideal transfer function. Gain error is usually expressed in LSB or as a percent of full-scale (%FSR). Gain error is the full-scale error minus the offset error (Figure 2-84).

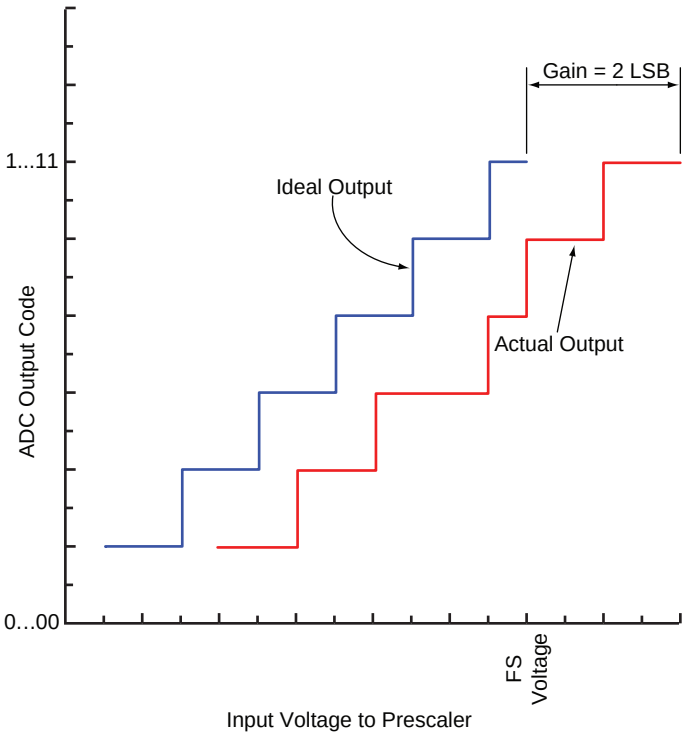


Figure 2-84 Gain Error

Gain Error Drift

Gain-error drift is the variation in gain error due to a change in ambient temperature, typically expressed in ppm/°C.

EQ 16 through EQ 18 can be used to calculate the acquisition time required for a given input. The STC signal gives the number of sample periods in ADCCLK for the acquisition time of the desired signal. If the actual acquisition time is higher than the STC value, the settling time error can affect the accuracy of the ADC, because the sampling capacitor is only partially charged within the given sampling cycle. Example acquisition times are given in Table 2-44 and Table 2-45. When controlling the sample time for the ADC along with the use of the active bipolar prescaler, current monitor, or temperature monitor, the minimum sample time(s) for each must be obeyed. EQ 19 can be used to determine the appropriate value of STC.

You can calculate the minimum actual acquisition time by using EQ 16:

$$VOUT = VIN(1 \quad e^{t/RC})$$

EQ 16

For 0.5 LSB gain error, VOUT should be replaced with (VIN (0.5 ×LSB Value)):

$$(VIN \quad 0.5 \times LSB\ Value) = VIN(1 \quad e^{t/RC})$$

EQ 17

where VIN is the ADC reference voltage (VREF)

Solving EQ 17:

$$t = RC \times \ln (VIN / (0.5 \times LSB\ Value))$$

EQ 18

where $R = Z_{INAD} + R_{SOURCE}$ and $C = C_{INAD}$.

Calculate the value of STC by using EQ 19.

$$t_{SAMPLE} = (2 + STC) \times (1 / ADCCLK) \text{ or } t_{SAMPLE} = (2 + STC) \times (ADC\ Clock\ Period)$$

EQ 19

where ADCCLK = ADC clock frequency in MHz.

t_{SAMPLE} = 0.449 µs from bit resolution in Table 2-44.

ADC Clock frequency = 10 MHz or a 100 ns period.

STC = (t_{SAMPLE} / (1 / 10 MHz)) 2 = 4.49 2 = 2.49.

You must round up to 3 to accommodate the minimum sample time.

Table 2-44 Acquisition Time Example with VAREF = 2.56 V

VIN = 2.56V, R = 4K (R_{SOURCE} ~ 0), C = 18 pF		
Resolution	LSB Value (mV)	Min. Sample/Hold Time for 0.5 LSB (µs)
8	10	0.449
10	2.5	0.549
12	0.625	0.649

Table 2-45 Acquisition Time Example with VAREF = 3.3 V

VIN = 3.3V, R = 4K (R_{SOURCE} ~ 0), C = 18 pF		
Resolution	LSB Value (m V)	Min. Sample/Hold time for 0.5 LSB (µs)
8	12.891	0.449
10	3.223	0.549
12	0.806	0.649

Sample Phase

A conversion is performed in three phases. In the first phase, the analog input voltage is sampled on the input capacitor. This phase is called sample phase. During the sample phase, the output signals BUSY and SAMPLE change from '0' to '1', indicating the ADC is busy and sampling the analog signal. The sample time can be controlled by input signals STC[7:0]. The sample time can be calculated by EQ 20. When controlling the sample time for the ADC along with the use of Prescaler or Current Monitor or Temperature Monitor, the minimum sample time for each must be obeyed.

Table 3-13 Summary of I/O Output Buffer Power (per pin) Default I/O Software Settings [1]

	C_{LOAD} (pF)	VCCI (V)	Static Power PDC8 (mW)[2]	Dynamic Power PAC10 (µW/MHz)[3]
Applicable to Pro I/O Banks				
Single-Ended				
3.3 V LVTTL/LVCMOS	35	3.3		474.70
2.5 V LVCMOS	35	2.5		270.73
1.8 V LVCMOS	35	1.8		151.78
1.5 V LVCMOS (JESD8-11)	35	1.5		104.55
3.3 V PCI	10	3.3		204.61
3.3 V PCI-X	10	3.3		204.61
Voltage-Referenced				
3.3 V GTL	10	3.3		24.08
2.5 V GTL	10	2.5		13.52
3.3 V GTL+	10	3.3		24.10
2.5 V GTL+	10	2.5		13.54
HSTL (I)	20	1.5	7.08	26.22
HSTL (II)	20	1.5	13.88	27.22
SSTL2 (I)	30	2.5	16.69	105.56
SSTL2 (II)	30	2.5	25.91	116.60
SSTL3 (I)	30	3.3	26.02	114.87
SSTL3 (II)	30	3.3	42.21	131.76
Differential				
LVDS		2.5	7.70	89.62
LVPECL		3.3	19.42	168.02
Applicable to Advanced I/O Banks				
Single-Ended				
3.3 V LVTTL / 3.3 V LVCMOS	35	3.3		468.67
2.5 V LVCMOS	35	2.5		267.48
1.8 V LVCMOS	35	1.8		149.46
1.5 V LVCMOS (JESD8-11)	35	1.5		103.12
3.3 V PCI	10	3.3		201.02
3.3 V PCI-X	10	3.3		201.02

Notes:

1. Dynamic power consumption is given for standard load and software-default drive strength and output slew.
2. PDC8 is the static power (where applicable) measured on VCCI.
3. PAC10 is the total dynamic power measured on VCC and VCCI.

Methodology

Total Power Consumption P TOTAL

Operating Mode, Standby Mode, and Sleep Mode

$P_{TOTAL} = P_{STAT} + P_{DYN}$

P_{STAT} is the total static power consumption.

P_{DYN} is the total dynamic power consumption.

Total Static Power Consumption P STAT

Operating Mode

$P_{STAT} = PDC1 + (N_{NVM\text{-}BLOCKS} * PDC4) + PDC5+ (N_{QUADS} * PDC6) + (N_{INPUTS} * PDC7) + (N_{OUTPUTS} * PDC8) + (N_{PLLS} * PDC9)$

$N_{NVM\text{-}BLOCKS}$ is the number of NVM blocks available in the device.

N_{QUADS} is the number of Analog Quads used in the design.

N_{INPUTS} is the number of I/O input buffers used in the design.

$N_{OUTPUTS}$ is the number of I/O output buffers used in the design.

N_{PLLS} is the number of PLLs available in the device.

Standby Mode

$P_{STAT} = PDC2$

Sleep Mode

$P_{STAT} = PDC3$

Total Dynamic Power Consumption P DYN

Operating Mode

$P_{DYN} = P_{CLOCK} + P_{S\text{-}CELL} + P_{C\text{-}CELL} + P_{NET} + P_{INPUTS} + P_{OUTPUTS} + P_{MEMORY} + P_{PLL} + P_{NVM} + P_{XTL\text{-}OSC} + P_{RC\text{-}OSC} + P_{AB}$

Standby Mode

$P_{DYN} = P_{XTL\text{-}OSC}$

Sleep Mode

$P_{DYN} = 0$ W

Global Clock Dynamic Contribution P CLOCK

Operating Mode

$P_{CLOCK} = (PAC1 + N_{SPINE} * PAC2 + N_{ROW} * PAC3 + N_{S\text{-}CELL} * PAC4) * F_{CLK}$

N_{SPINE} is the number of global spines used in the user design guidelines are provided in the "Spine Architecture" section of the Global Resources chapter in the Fusion and Extended Temperature Fusion FPGA Fabric User's Guide.

N_{ROW} is the number of VersaTile rows used in the design guidelines are provided in the "Spine Architecture" section of the Global Resources chapter in the Fusion and Extended Temperature Fusion FPGA Fabric User's Guide.

F_{CLK} is the global clock signal frequency.

$N_{S\text{-}CELL}$ is the number of VersaTiles used as sequential modules in the design.

Standby Mode and Sleep Mode

$P_{CLOCK} = 0$ W

Sequential Cells Dynamic Contribution P S-CELL

Operating Mode

FG676	
Pin Number	AFS1500 Function
A1	NC
A2	GND
A3	NC
A4	NC
A5	GND
A6	NC
A7	NC
A8	GND
A9	IO17NDB0V2
A10	IO17PDB0V2
A11	GND
A12	IO18NDB0V2
A13	IO18PDB0V2
A14	IO20NDB0V2
A15	IO20PDB0V2
A16	GND
A17	IO21PDB0V2
A18	IO21NDB0V2
A19	GND
A20	IO39NDB1V2
A21	IO39PDB1V2
A22	GND
A23	NC
A24	NC
A25	GND
A26	NC
AA1	NC
AA2	VCCIB4
AA3	IO93PDB4V0
AA4	GND
AA5	IO93NDB4V0
AA6	GEB2/IO86PDB4V0
AA7	IO86NDB4V0
AA8	AV0
AA9	GNDA
AA10	AV1

FG676	
Pin Number	AFS1500 Function
AA11	AV2
AA12	GNDA
AA13	AV3
AA14	AV6
AA15	GNDA
AA16	AV7
AA17	AV8
AA18	GNDA
AA19	AV9
AA20	VCCIB2
AA21	IO68PPB2V0
AA22	TCK
AA23	GND
AA24	IO76PPB2V0
AA25	VCCIB2
AA26	NC
AB1	GND
AB2	NC
AB3	GEC2/IO87PDB4V0
AB4	IO87NDB4V0
AB5	GEA2/IO85PDB4V0
AB6	IO85NDB4V0
AB7	NCAP
AB8	AC0
AB9	VCC33A
AB10	AC1
AB11	AC2
AB12	VCC33A
AB13	AC3
AB14	AC6
AB15	VCC33A
AB16	AC7
AB17	AC8
AB18	VCC33A
AB19	AC9
AB20	ADCGNDREF

FG676	
Pin Number	AFS1500 Function
AB21	PTBASE
AB22	GNDNVM
AB23	VCCNVM
AB24	VPUMP
AB25	NC
AB26	GND
AC1	NC
AC2	NC
AC3	NC
AC4	GND
AC5	VCCIB4
AC6	VCCIB4
AC7	PCAP
AC8	AG0
AC9	GNDA
AC10	AG1
AC11	AG2
AC12	GNDA
AC13	AG3
AC14	AG6
AC15	GNDA
AC16	AG7
AC17	AG8
AC18	GNDA
AC19	AG9
AC20	VAREF
AC21	VCCIB2
AC22	PTEM
AC23	GND
AC24	NC
AC25	NC
AC26	NC
AD1	NC
AD2	NC
AD3	GND
AD4	NC

FG676	
Pin Number	AFS1500 Function
AD5	IO94NPB4V0
AD6	GND
AD7	VCC33N
AD8	AT0
AD9	ATRTN0
AD10	AT1
AD11	AT2
AD12	ATRTN1
AD13	AT3
AD14	AT6
AD15	ATRTN3
AD16	AT7
AD17	AT8
AD18	ATRTN4
AD19	AT9
AD20	VCC33A
AD21	GND
AD22	IO76NPB2V0
AD23	NC
AD24	GND
AD25	NC
AD26	NC
AE1	GND
AE2	GND
AE3	NC
AE4	NC
AE5	NC
AE6	NC
AE7	NC
AE8	NC
AE9	GNDA
AE10	NC
AE11	NC
AE12	GNDA
AE13	NC
AE14	NC

FG676	
Pin Number	AFS1500 Function
AE15	GNDA
AE16	NC
AE17	NC
AE18	GNDA
AE19	NC
AE20	NC
AE21	NC
AE22	NC
AE23	NC
AE24	NC
AE25	GND
AE26	GND
AF1	NC
AF2	GND
AF3	NC
AF4	NC
AF5	NC
AF6	NC
AF7	NC
AF8	NC
AF9	VCC33A
AF10	NC
AF11	NC
AF12	VCC33A
AF13	NC
AF14	NC
AF15	VCC33A
AF16	NC
AF17	NC
AF18	VCC33A
AF19	NC
AF20	NC
AF21	NC
AF22	NC
AF23	NC
AF24	NC

FG676	
Pin Number	AFS1500 Function
AF25	GND
AF26	NC
B1	GND
B2	GND
B3	NC
B4	NC
B5	NC
B6	VCCIB0
B7	NC
B8	NC
B9	VCCIB0
B10	IO15NDB0V2
B11	IO15PDB0V2
B12	VCCIB0
B13	IO19NDB0V2
B14	IO19PDB0V2
B15	VCCIB1
B16	IO25NDB1V0
B17	IO25PDB1V0
B18	VCCIB1
B19	IO33NDB1V1
B20	IO33PDB1V1
B21	VCCIB1
B22	NC
B23	NC
B24	NC
B25	GND
B26	GND
C1	NC
C2	NC
C3	GND
C4	NC
C5	GAA1/IO01PDB0V0
C6	GAB0/IO02NDB0V0
C7	GAB1/IO02PDB0V0
C8	IO07NDB0V1

5 Datasheet Information

List of Changes

The following table lists critical changes that were made in each revision of the Fusion datasheet.

Revision	Changes	Page
Revision 6 (March 2014)	Note added for the discontinuance of QN108 and QN180 packages to the "Package I/Os: Single-/Double-Ended (Analog)" table and the "Temperature Grade Offerings" table (SAR 55113, PDN 1306).	II and IV
	Updated details about page programming time in the "Program Operation" section (SAR 49291).	2-46
	ADC_START changed to ADCSTART in the "ADC Operation" section (SAR 44104).	2-104
Revision 5 (January 2014)	Calibrated offset values (AFS090, AFS250) of the external temperature monitor in Table 2-49 Analog Channel Specifications have been updated (SAR 51464).	2-117
	Specifications for the internal temperature monitor in Table 2-49 Analog Channel Specifications have been updated (SAR 50870).	2-117
Revision 4 (January 2013)	The "Product Ordering Codes" section has been updated to mention "Y" as "Blank" mentioning "Device Does Not Include License to Implement IP Based on the Cryptography Research, Inc. (CRI) Patent Portfolio" (SAR 43177).	III
	The note in Table 2-12 Fusion CCC/PLL Specification referring the reader to SmartGen was revised to refer instead to the online help associated with the core (SAR 42563).	2-28
	Table 2-49 Analog Channel Specifications was modified to update the uncalibrated offset values (AFS250) of the external and internal temperature monitors (SAR 43134).	2-117
	In Table 2-57 Prescaler Control Truth Table AV (x = 0), AC (x = 1), and AT (x = 3), changed the column heading from 'Full-Scale Voltage' to 'Full Scale Voltage in 10-Bit Mode', and added and updated Notes as required (SAR 20812).	2-130
	The values for the Speed Grade (-1 and Std.) for FDDRIMAX (Table 2-180 Input DDR Propagation Delays) and values for the Speed Grade (-2 and Std.) for FDDOMAX (Table 2-182 Output DDR Propagation Delays) had been inadvertently interchanged. This has been rectified (SAR 38514).	2-220, 2-222
	Added description about what happens if a user connects VAREF to an external 3.3 V on their board to the "VAREF Analog Reference Voltage" section (SAR 35188).	2-225
	Added a note to Table 3-2 Recommended Operating Conditions1 (SAR 43429): The programming temperature range supported is $T_{ambient}$ = 0°C to 85°C.	3-3
	Added the Package Thermal details for AFS600-PQ208 and AFS250-PQ208 to Table 3-6 Package Thermal Resistance (SAR 37816). Deleted the Die Size column from the table (SAR 43503).	3-7
	Libero Integrated Design Environment (IDE) was changed to Libero System-on-Chip (SoC) throughout the document (SAR 42495). Live at Power-Up (LAPU) has been replaced with Instant On .	NA
Revision 3 (August 2012)	Microblade U1AFS250 and U1AFS1500 devices were added to the product tables.	I IV
	A sentence pertaining to the analog I/Os was added to the "Specifying I/O States During Programming" section (SAR 34831).	1-9