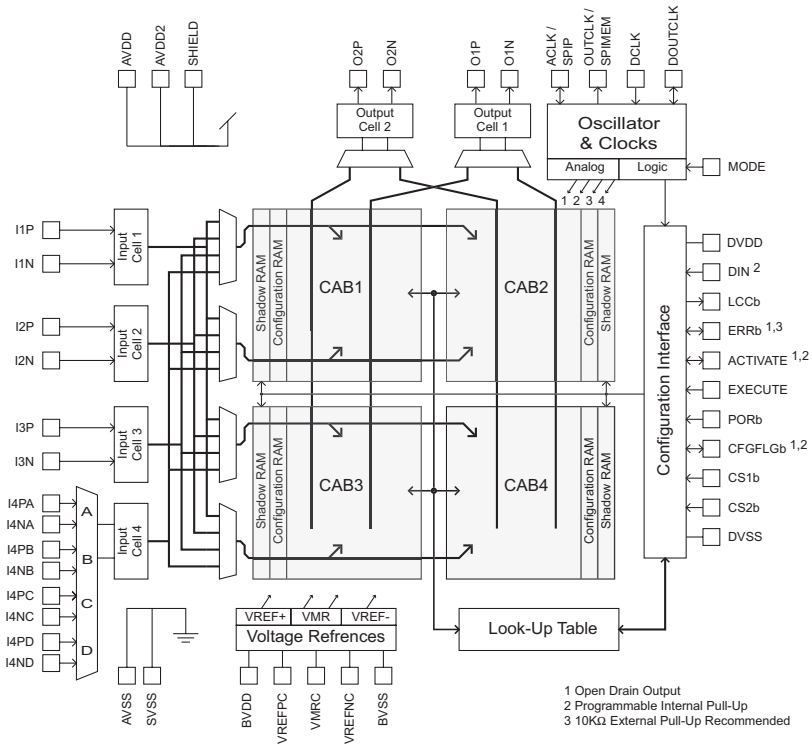
AN120E04
AN220E04

Field Programmable Analog Arrays - User Manual

Anadigmvortex is the second generation field programmable analog array (FPAA) device family from Anadigm®. Two members of the Anadigmvortex family are currently shipping - the low cost AN120E04 device that is geared towards high-volume applications requiring consolidation of discrete analog functionality; and the AN220E04 device that is further optimized to enable dynamic reconfiguration - a breakthrough capability that allows analog functions to be integrated within the system and controlled by the system processor.

The device consists of a 2 x 2 matrix of fully Configurable Analog Blocks (CABs), surrounded by a fabric of programmable interconnect resources. Configuration data is stored in an onchip SRAM configuration memory. Compared with our first-generation FPAAs, the Anadigmvortex architecture provides a significantly improved signal-to-noise ratio as well as higher bandwidth. The device also incorporates an 8-bit SAR-based Analog to Digital converter and a 256-byte look-up table. Combined, you can use these features to implement complex, non-linear analog functions such as sensor response linearization, arbitrary waveform synthesis, signal-dependent functions, analog multiplication, and signal companding. Based on a fully differential switched capacitor technology with an analog switch fabric, Anadigm®'s second-generation devices bring you a new level of device functionality and performance.

Furthermore, with the dynamic reconfigurability of the AN220E04, you can time-slice a single device to implement multiple analog functions or reconfigure the device on-the-fly to maintain precision operation despite system degradation and aging. The result is increased system functionality and longer system life.



Some of the notable features of the Anadigmvortex solution include:

- Analog design time reduction from months to minutes
- Faster time-to-solution compared to discretes or ASIC
- Ability to implement multiple chip configurations in a single device and to adapt functionality in the field
- High precision operation despite system degradation and aging
- Eliminating the need to source and maintain multiple inventories of product

The Anadigmvortex solution allows OEMs to deliver differentiated solutions faster and at lower overall system cost.

Anadigm® reserves the right to make any changes without further notice to any products herein. Anadigm® makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does Anadigm® assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including with out limitation consequential or incidental damages. "Typical" parameters can and do vary in different applications. All operating parameters, including "Typicals" must be validated for each customer application by customer's technical experts. Anadigm® does not in this manual convey any license under its patent rights nor the rights of others. Anadigm® software and associated products cannot be used except strictly in accordance with an Anadigm® software license. The terms of the appropriate Anadigm® software license shall prevail over the above terms to the extent of any inconsistency.

1 Architecture Overview 1

2 Analog Architecture Details 3

2.1 Input Cell 3
2.2 Muxed Analog Inputs 4
2.3 Output Cell 4
2.4 Configurable Analog Block 5
2.5 Look Up Table 6
2.6 SAR-ADC Operation 7
2.7 Voltage Reference and IBIAS Generators 7
2.8 System Clocks 8

3 Configuration Interface 9

3.1 Configuration and Clocks Pin List 10
3.1.1 MODE 11
3.1.2 DOUTCLK 11
3.1.3 DCLK (Data Clock) 12
3.1.4 ACLK/SPIP (Analog Clock/Serial PROM Clock) 12
3.1.5 OUTCLK/SPIMEM 12
3.1.6 PORb (Power On Reset) 12
3.1.7 ERRb (Error) 12
3.1.8 ACTIVATE 13
3.1.9 LCCb (Local Configuration Complete) 13
3.1.10 CFGFLGb (Configuration Flag) 13
3.1.11 DIN (Data In) 14
3.1.12 CS1b (Chip Select 1) 14
3.1.13 CS2b (Chip Select 2) 14
3.1.14 EXECUTE 15
3.2 Resets 15
3.3 Booting from EPROM 15
3.3.1 Boot from SPI EPROM 16
3.3.2 Single Device, Boot from Serial EPROM 17
3.3.3 Multiple Devices, Boot from Serial EPROM 18
3.4 Booting from a Host Processor 21
3.4.1 Synchronous Serial Interface Connections (SPI and SSI) 21
3.4.2 Typical Microprocessor Bus Connection 22
3.4.3 Advanced Feature - ACTIVATE 23
3.4.4 Advanced Feature - Return Read Data Path (AN220E04 Only) 23
3.5 Configuration Protocol 23
3.5.1 Primary Configuration Format & Byte Definitions 23
3.5.2 Header Block 25
SYNC BYTE 25
JTAG ID BYTE n 25
ID1 BYTE 25
3.5.3 Data Block 26
BYTE ADDRESS BYTE 27
BANK ADDRESS BYTE 27
DATA COUNT BYTE 28
DATA BYTE 28
ERR BYTE 28
3.5.4 Update Format (AN220E04 Only) 29
TARGET ID - Special Feature (AN220E04 Only) 30
3.5.5 Configuration Examples 30

Primary Configuration Format Example 31
Update Format Example (AN220E04 Only) 32
3.5.6 Configuration Clocking Considerations 32

4 Mechanical **33**
4.1 Package Pin Out 33
4.2 Recommended PCB Design Practices 34

Architecture Overview

Figure 1 – AN120E04 and AN220E04 Chip Overview 1

Analog Architecture Details

Figure 2 – A Regular Input Cell, there are three (See Figure 3 for the special muxed Input Cell.) 3
Figure 3 – Input Cell with a 4:1 Input Pair Multiplexer 4
Figure 4 – Analog Output Cell 4
Figure 5 – Overview of a Configurable Analog Block (CAB) 5
Figure 6 – Voltage Reference and Bias Current Generation 7
Figure 7 – Clock Features and Clock Domains 8

Configuration Interface

Figure 8 – Pins Associated with Device Configuration 10
Figure 9 – System Clocks when Configuration MODE = 0 11
Figure 10 – System Clocks when Configuration MODE = 1 11
Figure 11 – Known Compatible Crystals 12
Figure 12 – Known Compatible 25 Series SPI Memory Devices 16
Figure 13 – A Typical SPI EPROM Connection 16
Figure 14 – SPI EPROM, Initial Timing Sequence 16
Figure 15 – SPI EPROM, Completion Sequence 17
Figure 16 – Known Compatible 17 Series Serial EPROM Memory Devices 17
Figure 17 – A Typical Serial (FPGA) EPROM Connection 17
Figure 18 – Serial (FPGA) EPROM, Initial Timing Sequence 18
Figure 19 – Serial (FPGA) EPROM, Completion Sequence 18
Figure 20 – Connecting Multiple FPAAs to a Single Configuration Memory - Using a Crystal 19
Figure 21 – Multiple Device Serial EPROM Boot Handoff 19
Figure 22 – Connecting Multiple FPAAs to a Single Configuration Memory - Driving DCLK 20
Figure 23 – Using DOUTCLK in a Multi-Device System 20
Figure 24 – A Typical Host SPI Port Connection 21
Figure 25 – A Typical Host SSI Port Connection 21
Figure 26 – Typical Microprocessor External Bus Connection 22
Figure 27 – Timing Sequence for Booting from a Host Microprocessor 22
Figure 28 – Primary Configuration Data Stream Structure 24
Figure 29 – The Components of the 32 Bit JTAG Identification Number 25
Figure 30 – AN1/220E04 Memory Allocation Table 27
Figure 31 – Update Data Stream Structure 29
Figure 32 – Configuring Multiple Devices from a Host Processor 30

Mechanical

Figure 33 – Basic Guidelines for Optimal PCB Design (Ground Planes) 34

1 Architecture Overview

The AN120E04 and the AN220E04 devices are different in one key aspect. The AN220E04 device is dynamically reconfigurable. This device is optimized so that it can be updated partially or completely while operating.

The AN120E04 device can also be reprogrammed as many times as desired, however the device must first be reset before issuing another configuration data set..

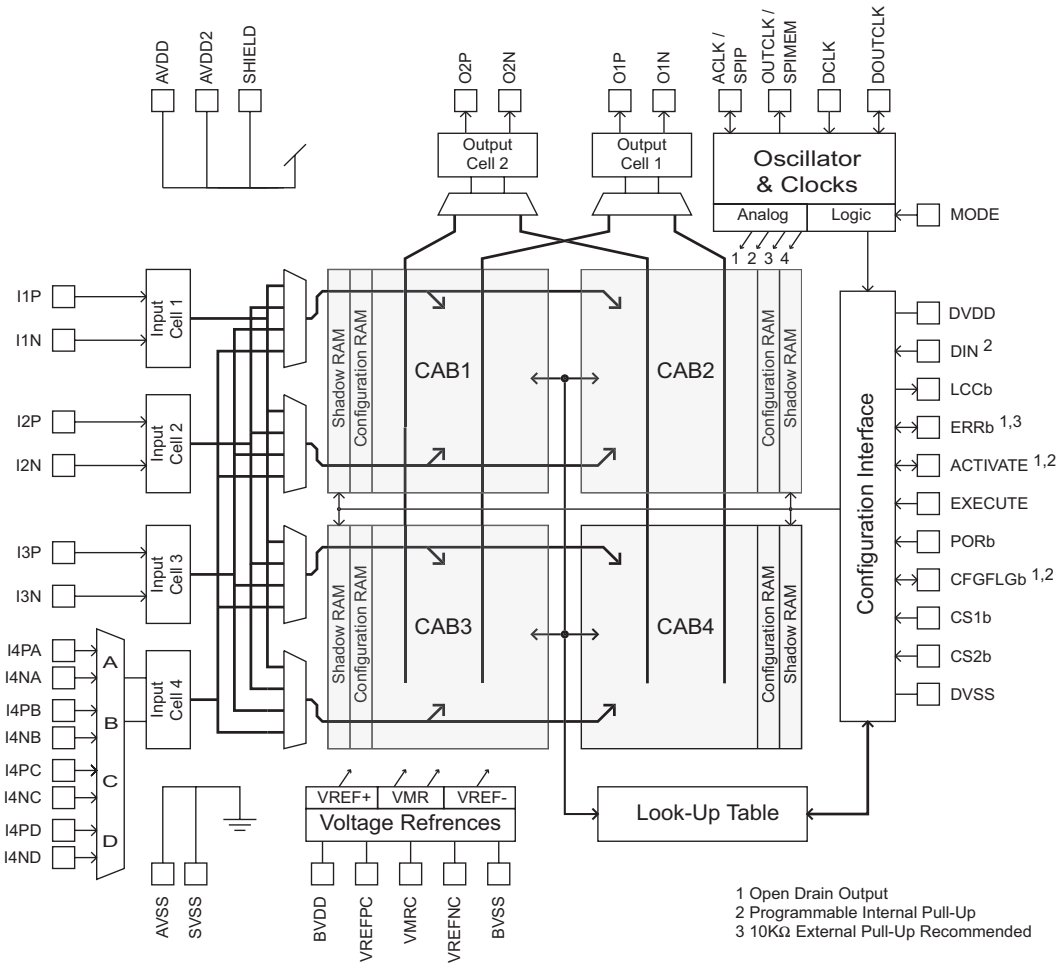


Figure 1 – AN120E04 and AN220E04 Chip Overview

The significant difference between the AN220E04 and AN120E04 devices is that once a Primary Configuration is complete; the configuration interface of the AN120E04 device ignores all further input. No further data writes are accepted unless a reset sequence is first completed.

Dynamic Reconfiguration available on the AN220E04 device, allows the host processor to send new configuration data to the FPAA while the old configuration is active and running. Once the new data load is complete, the transfer to the new analog configuration happens in a single clock cycle. Dynamic Reconfiguration in the AN220E04 device allows the user to develop innovative analog systems that can be updated (fully or partially) in real-time. The Field Programmable Analog Array (FPAA) contains 4 Configurable Analog Blocks (CABs) in its core. Most of the analog signal processing occurs within these CABs and is done with fully differential circuitry. The four CABs have access to a single Look Up Table (LUT) which offers a new method of adjusting any programmable element within the device in response to a signal or time base. It can be used to implement arbitrary input-to-output transfer functions (companding, sensor linearization), generate arbitrary signals, even perform voltage dependent filtering. A Voltage Reference Generator supplies reference voltages to each of the CABs within the device and has external pins for the connection of filtering capacitors.

Analog input signals can be connected from the outside world via the four Input Cells. The fourth Input Cell of the device has a special muxing feature which allows the connection of up to 4 unique signal sources. Each Input Cell can pass a differential signal pair directly into the array or process either single-ended or differential signals using combinations of: a unity gain buffer, a programmable gain amplifier, a programmable anti-alias filter and a special chopper stabilized amplifier. The chopper stabilized amplifier is especially designed for use with signals requiring significant gain and hence ultra low input offset voltages.

Output signals can be routed from within the array out through the output cells directly or through a programmable reconstruction filter and a pair of differential to single-ended converters. In either case, the external world signal is presented as a differential pair. The output cells can also be used to route out a logic level comparator output signal.

The device can accept either an external clock or generate its own clock using an on chip oscillator and an external crystal. Detection of the crystal is automatic. The resulting internal clock frequency can be divided down into four synchronized internal switched capacitor clocks of different frequencies by programmable dividers. The clock circuitry can also source any of these four clocks as a chip output.

The behavior of the CABs, clocks, signal routing, Input Cells, and Output Cells, is controlled by the contents of Configuration SRAM. Behind every Configuration SRAM bit is a Shadow SRAM bit. The Shadow SRAM of the AN220E04 device may be updated without disturbing the currently active analog processing. This allows for on-the-fly modification of one or more analog functions. This dynamic reconfiguration is not possible with the AN120E04 device.

The architecture includes a simple yet highly flexible digital configuration interface. The configuration interface is designed to work in stand-alone mode by connecting to either a common SPI or FPGA type serial EPROM. In this mode, after the device powers up, it will automatically load its configuration from the EPROM and begin functioning immediately thereafter.

The configuration interface is also designed to be connected directly to a host microprocessor's SPI master port where it presents itself as a SPI slave. It can also be accessed via a microprocessor's external data bus where the microprocessor's write strobe is recognized as a SPI clk and only a single data bit of the data bus is used for the serial SPI data. Reconfiguration of all or part of the device is supported in the AN220E04 device, allowing multiple configurations to be loaded over time if required. The configuration interface also allows multiple devices to be easily connected together to build up larger analog processing systems.

2 Analog Architecture Details

The following section explains some of the architecture details of the AN220E04 and the AN120E04 devices.

2.1 Input Cell

Each Input Cell contains a collection of resources which allow for high fidelity connections from the outside world with no need for additional external components. In order to maximize signal fidelity, all signal routing and processing within the device is fully differential. Accordingly each Input Cell accepts a differential signal input. A single ended signal may also be applied to the Input Cell. If a single ended source is attached, an internal switch will connect the negative side of the internal differential signal pair to Voltage Main Reference (VMR is the reference point for all internal signal processing and is set at 2.0 V above AVSS).

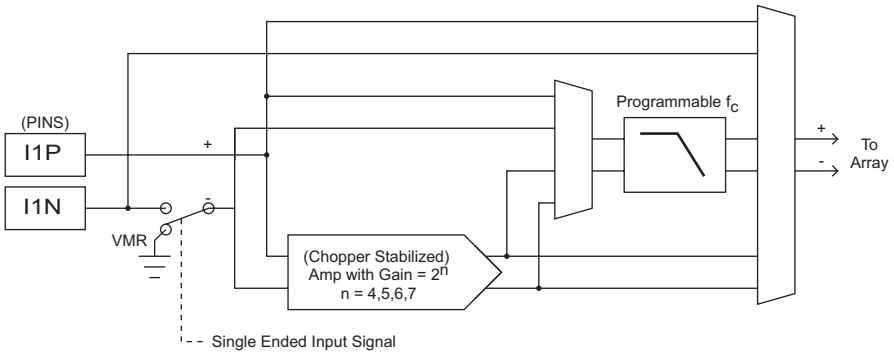


Figure 2 – A Regular Input Cell, there are three (See Figure 3 for the special muxed Input Cell.)

As with any sampled data system, it may sometimes be necessary to low pass filter the incoming signal to prevent aliasing artifacts. The input cell contains a second order programmable anti-aliasing filter. The filter may be bypassed, or set to selected corner frequencies.

When using the anti-aliasing filter, Anadigm recommends that the ratio of filter corner frequency to maximum signal frequency should be at least 30. These filters are a useful, integrated feature for low-frequency signals (signals with frequency up to 15kHz) only; and if high-order anti-aliasing is required, or where input signal frequencies are higher, Anadigm does recommend the use of external anti-aliasing.

A second unique resource available within each Input Cell is an amplifier with programmable gain and optional chopper stabilizing circuitry. The chopper stabilized amplifier greatly reduces the input offset voltage normally associated with op-amps. This can be very useful for applications where the incoming signal is very weak and requires a high gain amplifier at the input. The programmable gain of the amplifier can be set to 2^n where n = 4 through 7. The output of the amplifier can be routed through the programmable anti-aliasing input filter, or directly into the interior of the array (into a Configurable Analog Block). Single-ended input signals must use either the amplifier or the anti-alias filter in order to get the required single to differential conversion.

2.2 Muxed Analog Inputs

There is an input multiplexer available in front of one of the Input Cells. This allows the physical connection of 8 single ended or 4 differential pair input sources at once, though only one source at a time can be processed by the FPAA. As with the regular Input Cells, the optimal connection is from a differential signal source. If a single ended connection is programmed, the negative side of the internal differential pair will be connected to Voltage Main Reference.

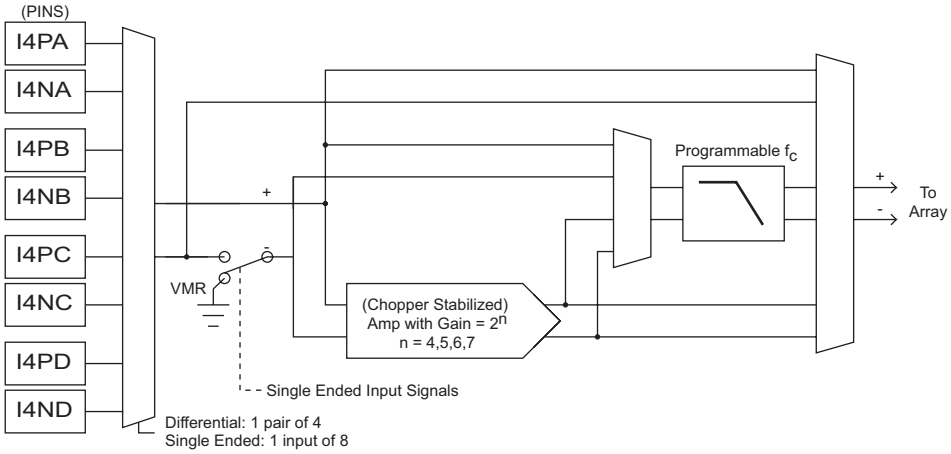


Figure 3 – Input Cell with a 4:1 Input Pair Multiplexer

2.3 Output Cell

Like the analog inputs, the analog Output Cells are loaded with features to ensure that your system's design can take full advantage of the fidelity and versatility that the core of the device offers. The outputs can serve to deliver digital data, or differential analog voltage signals.

Analog signal pairs sourced by CABs within the array are routed to an Output Cell via the Output Cell's input multiplexer.

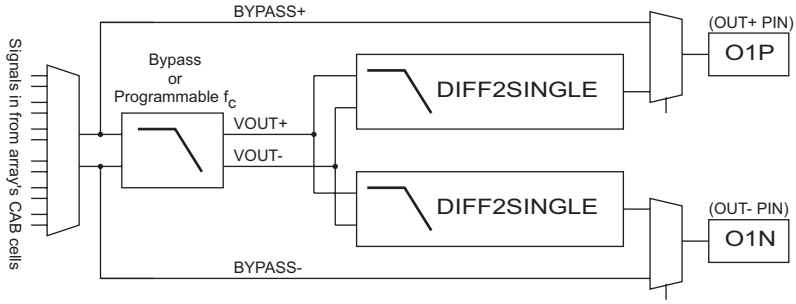


Figure 4 – Analog Output Cell

In some special circumstances, it may be desirable to route the core analog signals to the outside world with no additional buffering or filtering. The output cells have bypass paths which allow the core signals to come out with no further processing or buffering.

Each Output Cell contains a programmable filter identical to the one described for the Input Cells (see Section 2.1). The filter may be bypassed, or set to selected corner frequencies. Whereas the filter structure served as

an anti-aliasing filter for the input, in the Output Cell it serves as a 2nd order reconstruction filter. In this function, it smooths out the sampling induced stairstep nature of the output waveform.

A differential to single converter circuit follows the programmable filter. After the programmable filter and the DIFF2SINGLE conversion, the system designer may elect to utilize only one of the OUT signals, referencing it to Voltage Main Reference (VMR), or use them both (OUT+ and OUT-) as a differential pair. Remember that a single-ended output will have half the amplitude of a differential signal.

2.4 Configurable Analog Block

Within the device, there are four Configurable Analog Blocks (CABs). The functions available in the CAM library are mapped onto these programmable analog circuits. Figure 5 shows an overview of the Configurable Analog Block (CAB).

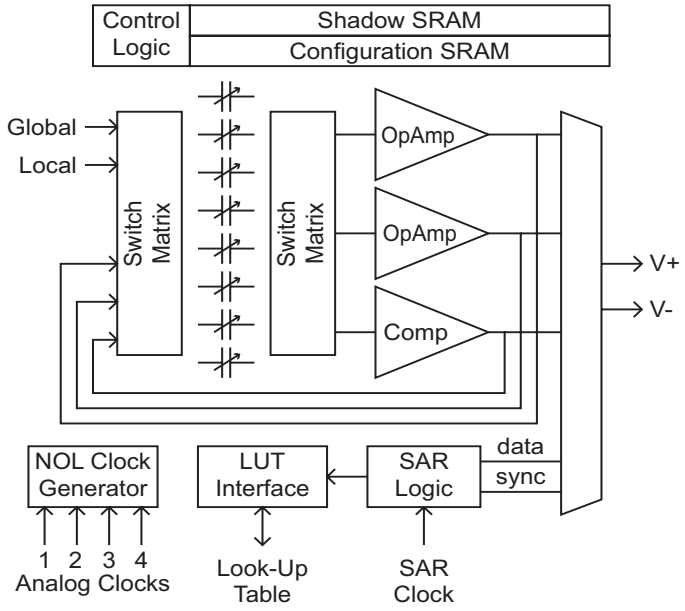


Figure 5 – Overview of a Configurable Analog Block

Among the many analog switches within the CAB, some are static and determine things like the general CAB circuit connections, capacitor values, and which input is active. Other switches are dynamic and can change under control of the analog input signal, the phase of the clock selected, and the SAR logic. Whether static or dynamic, all of the switches are controlled by the Configuration SRAM.

As part of the power-on reset sequence, SRAM is cleared to a known (safe) state. It is the job of the configuration logic to transfer data from the outside world into the Shadow SRAM and from there, copy it into the Configuration SRAM. The AN220E04 device allows reconfiguration. While an AN220E04 device is operating, the Shadow SRAM can be reloaded with values that will sometime later be used to update the Configuration SRAM. In this fashion, the FPAA can be reprogrammed on-the-fly, accomplishing anything from minor changes in circuit characteristics to complete functional context switches, instantaneously and without interrupting the signal path. The AN120E04 device must be reset between complete configuration loads and does not accept partial reconfigurations.

Analog signals route in from the cell's nearest neighbors using local routing resources. These input signals connect up to a first bank of analog switches. Feedback from the CAB's two internal opamps and single comparator also route back into this input switch matrix.

Next is a bank of 8 programmable capacitors. Each of these 8 capacitors is actually a very large bank of very small but equally sized capacitors. Each of these 8 programmable capacitors can take on a relative value between 0 and 255 units of capacitance. The actual value of capacitance is not all that important here. The CAM library elements do not depend on the absolute value of these capacitors, but rather on the ratio between them... which tracks to better than an 0.1% over process variations.

There is a second switch matrix to further establish the circuit topology and make the appropriate connections. There are two opamps and a single comparator at the heart of the CAB. Outputs of these active devices are routed back into the first switch matrix so feedback circuits can be constructed. These outputs also go to neighboring CABs.

Signal processing within the CAB is usually handled with a switched capacitor circuit. Switched capacitor circuits need non-overlapping (NOL) clocks in order to function correctly. The NOL Clock Generator portion of the CAB takes one of the four available analog clocks and generates all the non-overlapping clocks the CAB requires.

There is Successive Approximation Register (SAR) logic that, when enabled, uses the comparator within the CAB to implement an 8 bit Analog to Digital Converter (ADC). Routing the SAR-ADC's output back into its own CAB or to the Look Up Table enables the creation of non-linear analog functions like voltage multiplication, companding, linearization and automatic gain control.

2.5 Look Up Table

The device contains a single 256 byte Look Up Table (LUT). The 8 bit address input to the LUT can come from either the a SAR-ADC 8 bit output or from a special 8 bit LUT counter. The functional description of the SAR-ADC driving the LUT address inputs is given in the section below.

If the LUT counter is selected, the counter continuously counts up, resetting itself back to zero count each time that its programmable roll-over value is met. Each new count value is presented to the LUT as an address. The data read back from this address is then written into 1 or 2 target locations within Shadow SRAM. The target location(s) to be used and LUT contents are part of the device's configuration data set. The clock to the LUT counter is sourced by one of the 4 internal analog clocks (from one of the four clock dividers).

The subsequent transfer of these 1 or 2 bytes from Shadow SRAM into Configuration SRAM can occur as soon as the last configuration data byte is sent, or an internal zero crossing is detected, or a comparator trip point is met, or an external EXECUTE signal is detected.

With periodic clocking of the LUT counter, a LUT / CAB combination can form an arbitrary waveform generator, or temporally modulate a signal.

2.6 SAR-ADC Operation

Circuitry is included within the CAB which allows the construction of an 8-bit Successive Approximation Register (SAR) type Analog to Digital Converter (ADC). The SAR-ADC requires two clocks with a frequency ratio of 16 to 1. The slower clock (SAR clock, a.k.a. CLOCKA) determines the rate at which successive conversions will occur. The faster clock (SC clock, a.k.a. CLOCKB) is used to do the conversion itself. These clocks are generated by the normal clock divider circuitry.

The SAR result is in the sign magnitude format (1 bit sign, 7 bits magnitude). The SAR inputs should be limited to VMR ± 1.5 V. Inputs going above VREFPC (VMR + 1.5 V) and below VREFMC (VMR - 1.5 V) will result in the output railing to either 7F or FF as appropriate.

The SAR-ADC result can be routed to either the LUT's address port or back into its host CAB. The most common use of the SAR-ADC is to serve as an address generator for the Look Up Table. At the end of every conversion, the 8 bit result is recognized by the LUT as a new address. The configuration circuitry takes the LUT contents pointed to by this address and loads it into one or two specific locations in the Shadow SRAM.

A typical use scenario is where an input signal needs to be linearized and or calibrated. A signal comes in from the outside world and is presented to the CAB configured to do a SAR-ADC conversion. The SAR-ADC result is routed to the LUT where a linearization table was stored as part of the device's configuration image. Using the same mechanism as described for the LUT counter in section 2.5, the configuration circuitry takes the LUT contents pointed to by this address (the SAR-ADC result byte) and loads it into 1 or 2 specific Shadow SRAM locations. For this example, these locations would likely adjust the gain of an amplifier, thus achieving the desired linearization.

When the SAR-ADC conversion byte is routed directly back into the Configuration SRAM of its host CAB, self modifying circuits can be constructed.

2.7 Voltage Reference and IBIAS Generators

All analog signal processing within the device is done with respect to Voltage Main Reference (VMR) which is nominally 2.0 V. The VMR signal is derived from a high precision, temperature compensated bandgap reference source. In addition to VMR, VREF+ (1.5 V above VMR), and VREF- (1.5 V below VMR) signals are also generated for the device as shown in the figure below

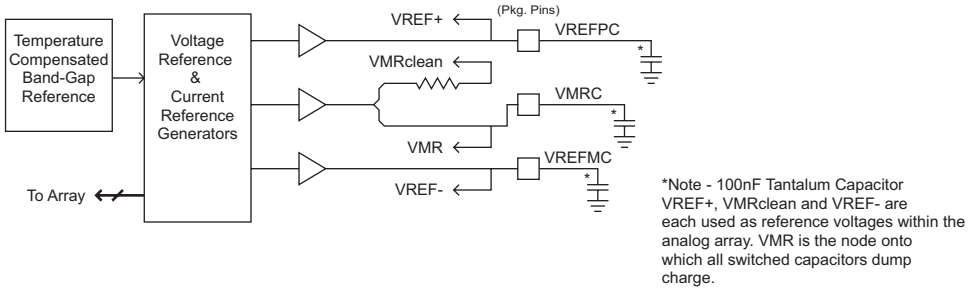


Figure 6 – Voltage Reference and Bias Current Generation

There are two versions of VMR routed to the CABs. VMR is the node onto which all switched capacitor charges get dumped and can be relatively noisy. VMRclean is also routed to the opamps within the CABs. This quiet version of VMR is used by the opamps as the ground reference in order to improve their settling times. It is required that external filtering caps be provided on VREFPC, VMRC, and VREFMC to ensure optimal chip performance. The recommended value for each is 75 to 100 nF. Higher values will have an adverse affect on settling time, lower values will reduce node stability. For highest possible performance, capacitors with a low series inductance, such as Tantalum, should be used. In most cases however, standard ceramic capacitors will be sufficient. VREF+ and VREF- are most often used by CAMs which utilize the comparator. In particular, these signals bound the recommended input range of SAR conversion CAMs.

2.8 System Clocks

Figure 7 provides a good high level overview of the various clock features and clock domains of the device. Not all of the features shown in this diagram are available in configuration MODE 1. See sections 3.1.1 through 3.1.5 for a complete explanation of these restrictions. The clock going to the configuration logic is always sourced at the DCLK pin. The DCLK pin may have an external clock applied to it up to 40 MHz. The DCLK pin may otherwise be connected to a series resonant crystal, in which case special circuitry takes over to form a crystal controlled oscillator. No programming is required. Connection of a crystal will result in a spontaneously oscillating DCLK. Please see section 3.1.3 for complete details on this feature.

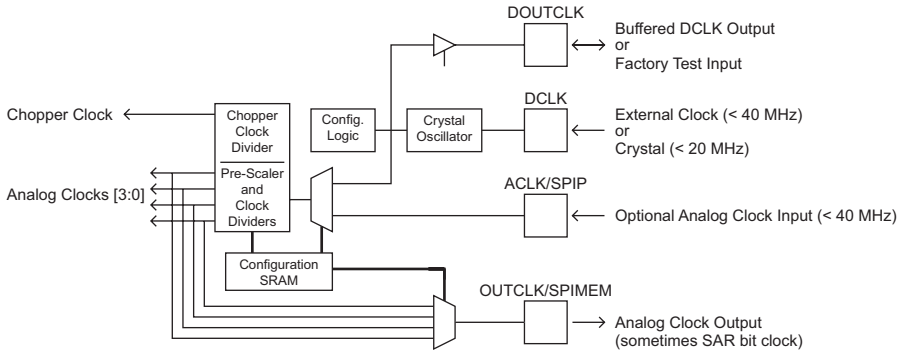


Figure 7 – Clock Features and Clock Domains

The analog clock domains are all sourced from a single master clock, either ACLK or DCLK. The device configuration determines which clock input will be used as the master clock. This master clock is divided into 5 unique domains. The first domain sources only the chopper stabilized amplifiers within the Input Cells. The other four domains are sourced by a user programmable prescaler feeding four user programmable dividers. Each of these domains can be used to drive either the SAR logic of a CAB, or the switched capacitor circuitry within the CAB itself. The clock generation circuitry ensures that all clocks derived from a single master clock signal will synchronize their rising edges (so that there is never any skew between 2 clocks of the same frequency). Importantly, this holds true for all clocks in a multi-device system as well.

3 Configuration Interface

The configuration interface provides a flexible solution for transferring configuration data into the configuration memory of the AN120E04 or AN220E04 devices.

The interface supports automatic standalone configuration from EPROM, with both SPI and FPGA serial EPROMS supported. The interface also supports configuration from an intelligent host via a standard SPI or SSI interface or via a typical microprocessor bus interface. Selection between these two configuration modes is accomplished with the MODE pin. Configuration speeds of up to 40MHz are supported.

Configuration from either a host or EPROM as above is possible with both the AN120E04 and the AN220E04. The AN220E04 however offers the additional feature of allowing reconfiguration of the device via the host. This feature, which is not supported by the AN120E04, allows the reconfiguration of all or any part of the device repeatedly and at will using the reconfiguration protocol. Thus the FPAA's behavior can be adjusted on-the-fly to meet the dynamic requirements of the application.

The configuration data is stored in SRAM based configuration memory distributed throughout the FPAA. There are two SRAM memories on the chip: Shadow SRAM and Configuration SRAM. Configuration data is first loaded into Shadow SRAM, and then on a single user-controllable clock edge, is loaded into Configuration SRAM. The device's analog functionality behaves according to the data in Configuration SRAM. This method allows configuration data to be loaded into the device in the background and take effect instantly when required.

Read out of the Configuration SRAM is supported allowing users to check data integrity if required. Read back applies only to the AN220E04. The AN120E04 does not support readback.

The device also features a Look Up Table (LUT). The LUT exists as part of the Configuration SRAM and can be read and written to as normal, but Shadow SRAM for the LUT is not supported. Thus data written to the LUT becomes effective as it is written.

On power up, internal power on reset circuitry is activated which resets the device's Configuration SRAM and prepares the device for a first or Primary Configuration. Primary Configuration then proceeds according to the protocol described later in this document. Once completed, reconfigurations can be executed as described above. In the case of the AN120E04, reconfiguration data is ignored.

Multiple device systems are supported through a daisy chain connection of configuration interface signals, and logical addressing via the host using the reconfiguration protocol. Devices can be reconfigured concurrently or singly. Thus groups of devices can be updated together or devices can be updated separately using exactly the same connections to the host.

3.1 Configuration and Clocks Pin List

The device has many advanced configuration features and as a consequence many of the pins of the configuration interface have multiple functions. Subsequent sections describe the typical connection schemes.

Name	Type	Functions
DOUTCLK	Output	Buffered version of DCLK. Inactive (floats) until its associated configuration bit is set. If unused this pin must be left floating.
	Input	(Factory reserved test input. Float if unused.)
MODE	Input	0, select clock support for synchronous serial interface 1, select clock support for SPI & FPGA EPROM interface
DCLK	Input	drive with < 40 MHz external configuration clock, or attach a 12, 16, 20, or 24 MHz crystal
ACLK / SPIP	Input	MODE = 0, Analog Clock (Switched Capacitor Clock) < 40 MHz
	Output	MODE = 1, SPI EPROM or Serial EPROM Clock
OUTCLK / SPIMEM	Output	During power-up, sources SPI EPROM initialization command string . After power-up, sources selected internal analog clock or comparator output.
PORb	Input	0, Chip Held in reset state Rising Edge, re-initiates power on reset sequence - 30 mS to complete
ERRb	Input	0, Initiate Reset (hold low for 15 clocks) 1, No Action
	O.D. Output 10 kΩ p/u reqd.	0, Error Condition Z, No Error Condition
ACTIVATE	Input	0, Hold off completion of configuration Rising Edge, Complete configuration
	O.D. Output	0, Device has not yet completed Primary Configuration Z, Device has completed Primary Configuration
LCCb	Output	0, Local configuration complete 1, Local configuration is not complete Once configuration is completed, it is a delayed version (8 clock cycles) of CS1b or if the device is addressed for read, it serves as serial data read output port.
CFGFLGb	Input	In multi-device systems... 0, Ignore incoming data (unless currently addressed) 1, Pay attention to incoming data (watching for address)
	O.D. Output	0, Device is being reconfigured Z, Device is not being reconfigured
DIN	Input	Serial Configuration Data Input
CS1b	Input	(Prior to completion of a Primary Configuration) 0, Allow configuration to proceed 1, Hold off configuration
	Input	(After completion of a Primary Configuration) Data input pin, serves as a serial data pass through port for a multi-device chain.
CS2b	Input	0, Chip is selected 1, Chip is not selected
EXECUTE	Input	0, No Action. This pin should normally be tied low. 1, Transfer Shadow SRAM into Configuration SRAM, depending on configuration settings

Figure 8 – Pins Associated with Device Configuration

3.1.1 MODE

MODE controls the behavior of the analog and configuration clocks portion of the device. The state of the MODE pin establishes a unique configuration for the device's clock pins as shown in the Figures 9 and 10.

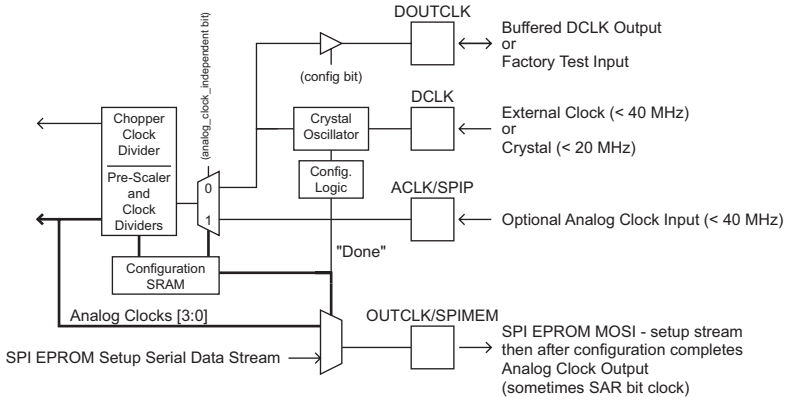


Figure 9 – System Clocks when Configuration MODE = 0

In MODE 0, ACLK/SPIP is an optional clock input that can be used to serve as the master clock for the analog clock domains within the device.

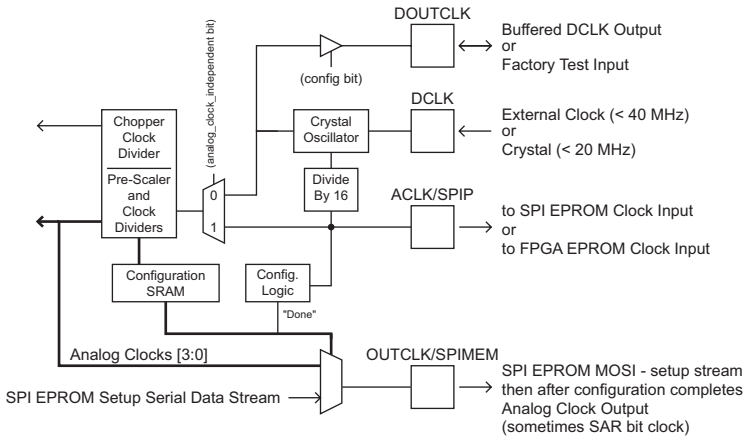


Figure 10 – System Clocks when Configuration MODE = 1

In MODE 1, ACLK/SPIP is an output which is a divided down version of the DCLK input. The intended connection is to a serial memory device's clock input.

Regardless of MODE setting, at the beginning of configuration, OUTCLK/SPIMEM sources a serial data bitstream designed to set up a 25 series SPI EPROM for read. The intended connection is to the MOSI (Master Out Slave In) pin of a SPI EPROM. FPGA type serial EPROMs need no such initialization. In this case the OUTCLK/SPIMEM pin is typically not used. Once configuration completes, OUTCLK/SPIMEM reverts to serving as an analog clock or comparator output port.

3.1.2 DOUTCLK

When enabled by configuration data, the DOUTCLK output provides a buffered version of DCLK. This is useful when using the oscillator feature of DCLK in the master device of a multi-device system. In this scenario, a crystal is attached to master device's DCLK input (rather than driving DCLK with a conventional clock source). See Figure 23 for further details.

Note that the clocks of this master device will be running about 10 ns ahead of all the slaves. For this reason, the clock master should be the last device in the analog chain. i.e. Analog outputs from the clock master device should not be connected to analog inputs of clock slaves.

When this feature is not used, DOUTCLK should be floated. When this configuration data bit is not set, DOUTCLK becomes a factory reserved test input with an internal pull-down. Setting this bit disables both the input and the pull-down device.

3.1.3 DCLK (Data Clock)

The rising edge of the input on the DCLK pin is used to drive the configuration logic. Until a clock is supplied, the internal power-up procedure cannot be completed. The maximum DCLK frequency is 40MHz. The supplied clock can be free running or a strobe.

An interesting feature of the DCLK input is that it may be driven with a standard logic signal, or a series resonant crystal can be connected (to DVSS). The device's on chip oscillator automatically detects an attached crystal and uses it to establish a self generated internal clock that can be used by both the configuration logic and analog portions of the device. The allowable frequency range for an attached crystal is between 12 MHz and 24 MHz, with 16 MHz being the optimal choice.

Mfr.	Frequency	Part Number
NDK	12, 16, 20 MHz	NX8045GB
C-MAC	12, 16, 20 MHz	12 SMX
AeL	12, 16, 20, 24 MHz	SXH

Figure 11 – Known Compatible Crystals

3.1.4 ACLK/SPIP (Analog Clock/Serial PROM Clock)

In MODE 0, ACLK/SPIP is an optional analog master clock input to drive the switched capacitor circuitry within the device.

In MODE 1, ACLK/SPIP is a clock output which is normally used to clock data out of a Serial FPGA EPROM or provide the SPI master clock to an attached SPI EPROM. See sections 3.3.1 and 3.3.2 for listing of compatible EPROMs. It is a divide down (by16) version of DCLK.

3.1.5 OUTCLK/SPIMEM

During power-up, the OUTCLK/SPIMEM pin transmits control words to the attached SPI memory device (if any). A SPI EPROM requires a control word to be sent followed by a 16 bit start address. After configuration, the OUTCLK/SPIMEM pin routes out any 1 of the 4 internal analog clocks as enabled by the configuration data.

3.1.6 PORb (Power On Reset)

When PORb is asserted low, the device's internal power on reset circuitry is re-activated as if the device were being powered up for the first time. When utilized as a control signal, PORb is normally just pulsed low, but it can be held low for an indefinitely long period of time. Once PORb is released high, the POR circuit completes a normal power-on reset sequence and control is handed over to the configuration state machine.

3.1.7 ERRb (Error)

This is an open drain input/output pin. An external pull-up resistor should be attached to this pin, typically 10 kΩ. In large multi-device systems, this pull-up may be reduced to 5K to overcome loading effects. Initially during power-up, this pin serves as an output and is asserted low by the device. As the power-on reset sequence progresses the ERRb pin is released and is pulled up by the external resistor, allowing the configuration sequence to commence. If there is more than one FPAA in a system, then the ERRb pins should be tied together. This forces power-up to be delayed until ERRb has been released in all devices. Different device types will take different times to power-up. The rising edge of ERRb is therefore used to synchronize all FPAAs in the system to the same incoming clock cycle. Once ERRb goes high configuration can begin.

A user can manually delay the start of configuration by externally pulling ERRb low from power-up.

An alternative method for delaying configuration is to hold the CS2b pin high during power-up. ERRb will not be released until CS2b is taken low. ERRb remains high during configuration and reconfiguration unless an error occurs. An error condition is indicated by ERRb being asserted low by the device in which the error occurs.

As controlled by the configuration data set, it is possible to set the length of the error pulse to be short or long in the device which generates it. Short pulses are ignored by all other devices in the system, and the device which generated the error resets to the point where a simple Update is required. If a long ERRb pulse is generated, then for both single and multiple device systems a Primary Configuration can begin immediately. Long pulses are detected by all other devices in the system, which reset to the point where a complete Primary Configuration is required. The device which generated the error also resets to this state.

As an output, a long ERRb pulse is asserted low for 15 DCLK clocks. A short ERRb output pulse is 1 DCLK clock long. As an input, ERRb is recognized asserted when held low for 15 or more DCLK periods.

The ERRb pin may be used to force the device to do a Primary Configuration. If ERRb is pulled low externally after power up completes, then the device is reset and Primary Configuration will begin again once ERRb is released. If used as an input for this purpose, the input low period should be at least 15 DCLK periods long.

3.1.8 ACTIVATE

The ACTIVATE pin is an open drain input/output with an internal pull-up resistor selectable via configuration. It asserts low during power-up and remains low until Primary Configuration is complete when it is released and pulls high using only the pull-up resistor. It remains de-asserted (tri-stated and pulled high) thereafter. Once ACTIVATE pulls high configuration is allowed to complete if the ENDEXECUTE bit is set. See section 3.5.1 for further detail.

If there is more than one FPAA in a system, all ACTIVATE signals should be tied together to ensure that all devices conclude their configuration at exactly the same time. The ACTIVATE signal is also intended to be used to disable a standard FPGA Serial EPROM, if used, once configuration is complete. (See section 3.4.3 for further detail.)

The internal pull-up associated with the ACTIVATE pin is selectable through a control byte bit and becomes active immediately after the control byte is latched in.

3.1.9 LCCb (Local Configuration Complete)

During power-up the LCCb output drives high. So long as the Primary Configuration is incomplete the LCCb pin will continue to drive high. Just before configuration completes the LCCb pin asserts low. In multi-device systems, this output is normally connected to the CS1b (input) pin of the next device in the configuration chain, allowing that device's configuration sequence to commence. See section 3.3.3 for further details on the LCCb to CS1b connection.

Once configuration completes (two clocks after ACTIVATE asserts high), the LCCb pin becomes a data output. If the device is being read from, then LCCb serves as the serial data output pin for the read data. If the device is not being read from, then LCCb is simply a registered version of CS1b, allowing serial data to pass through the device for a multi-device configuration serial data chain. See Figure 32 for a detailed look at this configuration.

3.1.10 CFGFLGb (Configuration Flag)

The CFGFLGb pin is an open drain input/output with an internal pull-up resistor selectable via configuration. CFGFLGb is first driven high then driven low during power-up and remains low until Primary Configuration is complete when it is released and pulls high using the internal or external pull-up. The pin will drive low again at the beginning of reconfiguration and remain low until the end of reconfiguration when it is released and allowed to pull high once again.

In a multi-device system the CFGFLGb pins should all be tied together. Devices in a multi-device system that are not being addressed for reconfiguration ignore input data until CFGFLGb pulls high. The CFGFLGb pin can be monitored by the user to indicate when configurations are in progress.

The CFGFLGb is also used to initialize and chip select a SPI memory, if used, as these memories require a falling edge on their chip select input to reset. This edge is provided when CFGFLGb is driven low during power-up. The instruction and address data subsequently output by the OUTCLK pin to initialize the SPI memory is synchronized to this falling edge.

The internal pull-up is selectable through a control byte bit and becomes active immediately after the control byte is latched in.

3.1.11 DIN (Data In)

DIN is the serial data input pin. During power-up, this pin is ignored. During configuration (or reconfiguration), DIN is the serial data input for configuration data. There is a weak internal pull-up on DIN which if configured ensures a valid logic state after an attached serial EPROM goes tri-state.

3.1.12 CS1b (Chip Select 1)

Prior to Primary Configuration, while CS1b and CS2b are both low, DCLK is used to clock the configuration state machine. Once Primary Configuration is complete, signals on CS1b are delayed by 8 clocks and passed to LCCb.

CS1b therefore behaves as an active low chip select. CS1b should be synchronous to the configuration clock (DCLK).

Note that CS1b is actually an intelligent polling I/O, although this function is entirely transparent to the user. It operates on a cyclic basis, writing a weak logic 1 out during the high period of the clock and reading the logic state on the pin during the low period of the clock. On the next rising edge of the clock the logic state on the CS1b pin is latched internally.

See section 3.3.3 for further details on the LCCb to CS1b connection.

3.1.13 CS2b (Chip Select 2)

CS2b is an active low chip select input pin. CS2b should be synchronous to the configuration clock. It is sampled at the end of the power-up period, and if high it stops the ERRb pin from going high. Completion of power-up is delayed until CS2b goes low. A user can therefore delay configuration by holding CS2b high from power-up. Once CS2b goes low, ERRb goes high and configuration begins.

CS2b is logically NOR'ed with CS1b and the output is used to gate the incoming clock to the configuration circuitry. In other words, when CS1b and CS2b are both low, the clock is enabled. To ensure that this gating always results in a clean clock to the main configuration state machine, CS2b should be synchronous to the configuration clock.

It is important to note that when booting from either a Serial EPROM or SPI EPROM, CS2b must remain low throughout configuration, otherwise the data clocked out of the EPROM will not be clocked into the device. This happens because CS2b does not stop data being clocked out of the EPROM, but does stop it from being clocked into the device. After configuration CS2b continues to act with CS1b as a clock enable for the device. A user can therefore hold CS2b high after configuration to reduce power consumption in the device.

3.1.14 EXECUTE

The EXECUTE input pin should normally be tied low.

3.2 Resets

The available sources of reset include: a hard power cycle, asserting ERRb low for not less than 15 DCLK's, pulsing PORb low, issuing a software reset during the Primary Configuration, or after an error occurs during a Primary Configuration and ERRb asserts low.

Assume ERRb is not actively driven and only pulled high through an external pull-up resistor and PORb is similarly pulled high. After power is applied to the device an internally generated power-on reset pulse resets the power-up and configuration state machines. The power-on reset pulse also resets the configuration memory.

The power-up state machine does not start until a clock becomes valid and has clocked 5 times. This help protect the system from functioning until the clock is stable. Thereafter, the power-up state machine takes control. Once power-up is complete, the configuration state machine becomes active and configuration proceeds.

Assume the device is configured and operating normally. Driving ERRb low for longer than 15 configuration clock cycles, will cause the device to reset. The device will become active and configuration can occur once again.

Assume the device is configured and operating normally with ERRb pulled up high. If PORb is driven low the device will reset. Holding PORb low, keeps the device in a power-on reset condition. When PORb is finally released to a high state, the power-on reset circuitry will recognize a rising edge on PORb and be tricked into thinking that the device is powering up. The normal power-up sequence will repeat.

3.3 Booting from EPROM

The simplest method of configuring the device is to let it self boot from a non-volatile serial memory. The FPAA is directly compatible with both industry standard 25 series SPI EPROMs and 17 series Serial EPROMs (most commonly used as FPGA configuration memories).

As system power comes up, the device first completes its internal power-on reset, checks the state of the CS1b and CS2b pins and if set correctly will then provide the necessary signalling to read data out of either of these two common EPROM types. Once the read of configuration data is complete, the device will automatically activate the analog circuitry. The entire power-on reset and configuration process is automatic.

3.3.1 Boot from SPI EPROM

A typical connection scheme for an industry standard 25 series SPI EPROM is shown in Figure 13. Once the device's internal power-on reset sequence completes, CFGFLGb will assert low selecting the memory device. The OUTCLK/SPIMEM pin sends serial command words to the SPI memory instructing it to begin delivering data starting from its internal address 0. At the appropriate time, the FPAA will start paying attention to its DIN pin, expecting a SYNC byte followed by configuration data. If there is any error encountered during this process, ERRb will assert low and the device will ignore all subsequent serial data.

Mfr.	Part Number
Atmel	AT25080
Xicor	X5043
Microchip	25AA160
Fairchild	NM25C640

Figure 12 – Known Compatible 25 Series SPI Memory Devices

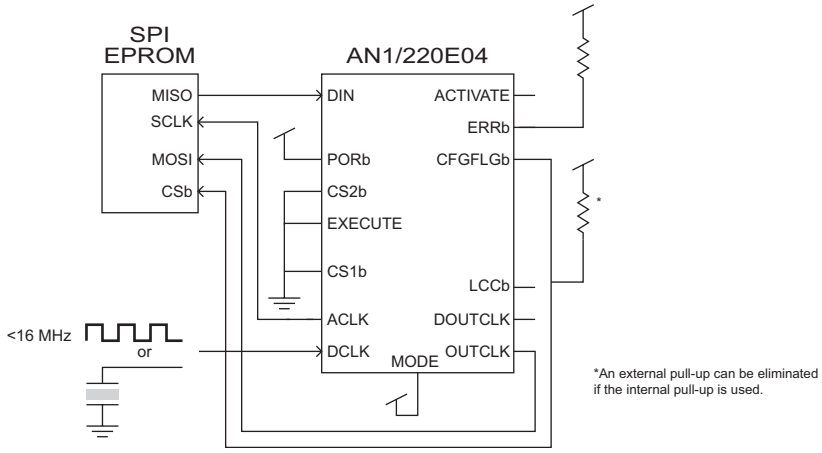


Figure 13 – A Typical SPI EPROM Connection

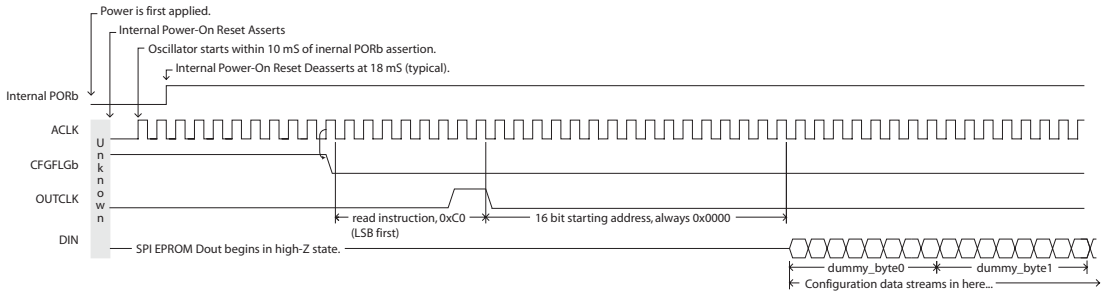


Figure 14 – SPI EPROM, Initial Timing Sequence

As the system's power supply first begins to ramp up, the ACLK, CFGFLGb, and OUTCLK signals are unknown. Soon though, the device's internal Power-On Reset circuitry asserts and gets everything in a known state. The FPAA's oscillator has a typical start up time of less then 10 mS, and the POR circuitry will conclude within 30 mS. (Note- This diagram is not to scale. At an ACLK rate of 2 MHz, the entire Primary Configuration occurs within 3 mS of the internal PORb de-assertion.)

SPI data is sourced by the SPI EPROM on the falling edge of ACLK. Setup time of DIN to ACLK should be greater than 2 nS. Hold time is 0 nS.

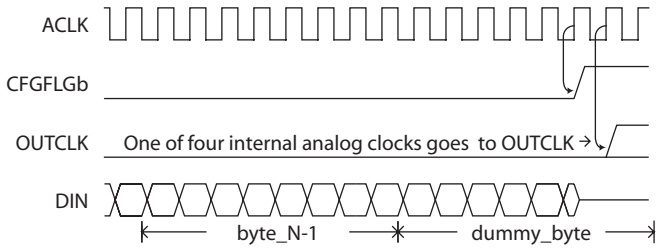


Figure 15 – SPI EPROM, Completion Sequence

As the last configuration data byte (a dummy byte) is being clocked into the device CFGFLGb de-asserts. One ACLK later, OUTCLK will drive out one of the four internal analog clocks or one of the four comparator outputs if the configuration data set directs it to do so. If no clock is selected to be routed to OUTCLK, the pin will be driven low instead.

3.3.2 Single Device, Boot from Serial EPROM

A second readily available serial memory type is the 17 series. These Serial EPROMs are typically used as configuration memories for FPGAs. Unlike SPI EPROMs there is no command input path to these devices (OUTCLK/SPIMEM is typically left unconnected). Once both CEb is asserted and RESETb released, these memories simply stream serial data out with each clock. As with the SPI EPROM, all of this happens automatically as the power-on reset sequence concludes.

Mfr.	Part Number
Xilinx	XC1700E
Atmel	AT17 series
Altera	ECP1, ECP2

Figure 16 – Known Compatible 17 Series Serial EPROM Memory Devices

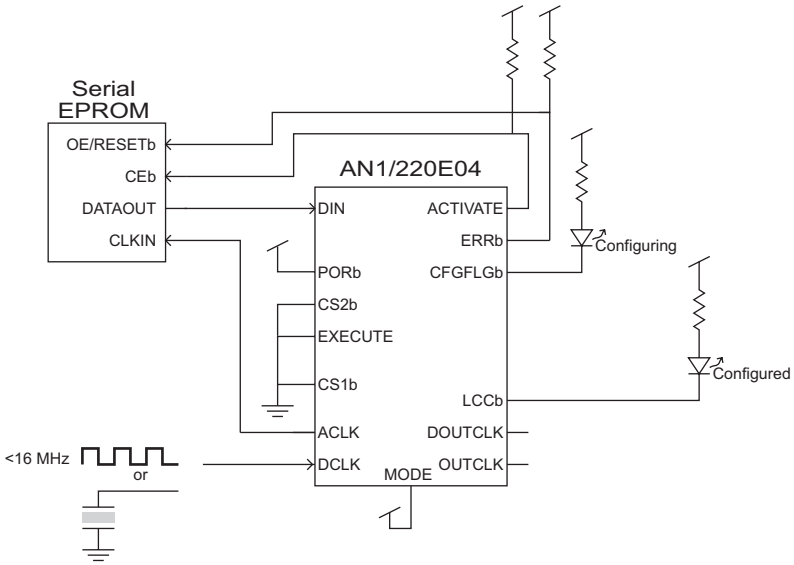


Figure 17 – A Typical Serial (FPGA) EPROM Connection

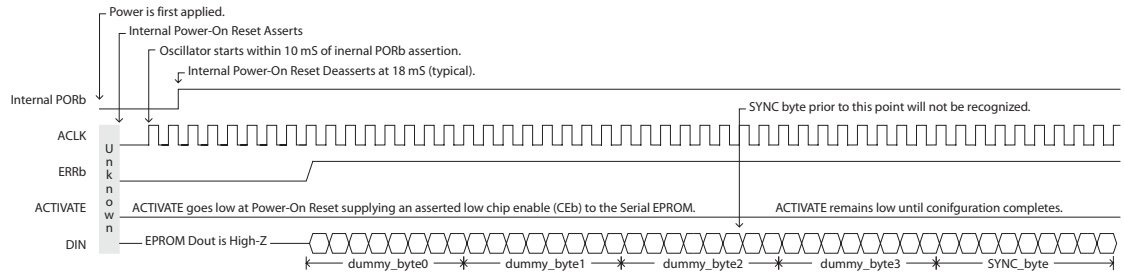


Figure 18 – Serial (FPGA) EPROM, Initial Timing Sequence

The start up sequence is shown in Fig 18. The only significant difference between using a SPI EPROM and an FPGA type Serial EPROM is that the serial command words being sourced by the (unconnected) OUTCLK/SPIMEM pin are ignored by the FPGA EPROM. Because the EPROM will start sourcing data before the FPAA is ready to accept it, the configuration data should be prefixed with 4 dummy bytes prior to the SYNC byte and the meaningful configuration data. AnadigmDesigner®2 design software handles this insertion of the requisite prefix data. As described previously, a few clocks are need at the end to complete the configuration. A postfix of one dummy byte is also added to the meaningful configuration data.

17 series FPGA serial EPROMs output data on the rising edge of ACLK. Setup time of DIN to ACLK should be greater than 2 nS. Hold time is 0 nS.

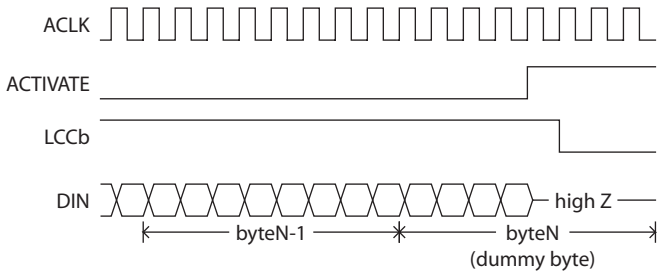


Figure 19 – Serial (FPGA) EPROM, Completion Sequence

3.3.3 Multiple Devices, Boot from Serial EPROM

Multiple FPAAs may be configured from a single EPROM (either Serial or SPI type). The first FPAA in the chain has both of its chip selects pulled low and so it begins configuring immediately after power up. All down-stream devices are stopped from configuring because their CS1b inputs are held high. As the first FPAA in the chain completes its self configuration, it de-asserts LCCb. This flags the next device in the chain to begin its configuration sequence and so on down the chain.

Tying all the open drain ERRb bi-directional pins together ensures that if any one of the devices in the configuration chain detects an error and fails to configure, then all of the devices in the chain will be reset and Primary Configuration will start again.

Likewise, all of the devices in the configuration chain have their open drain ACTIVATE bi-directional pins also tied together. As each device completes it's configuration, it ceases to drive the ACTIVATE line low. As the last device in the chain completes its configuration it too will cease to drive ACTIVATE low. In this manner, all the analog circuitry will become active on the next configuration clock after the ACTIVATE line pulls high.

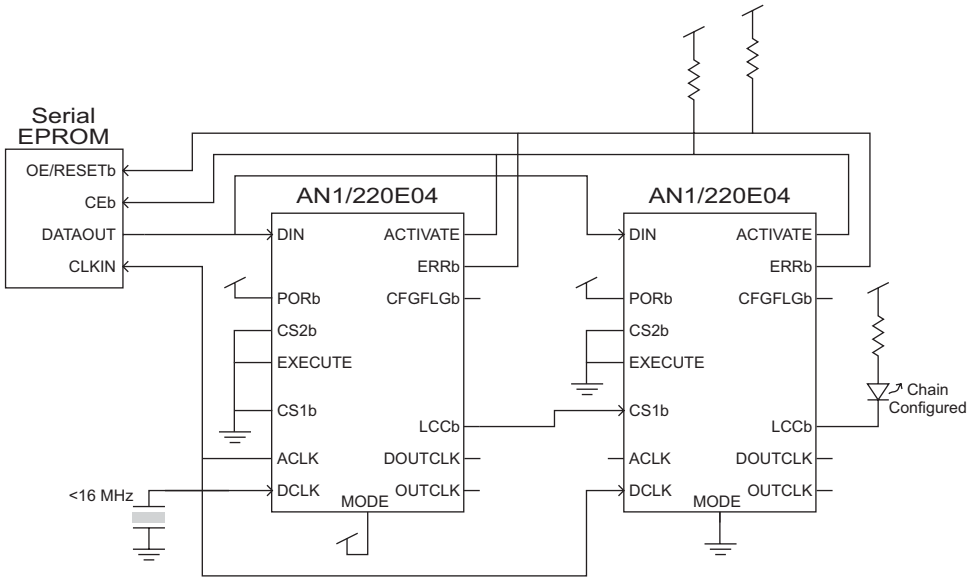


Figure 20 – Connecting Multiple FPAAs to a Single Configuration Memory - Using a Crystal

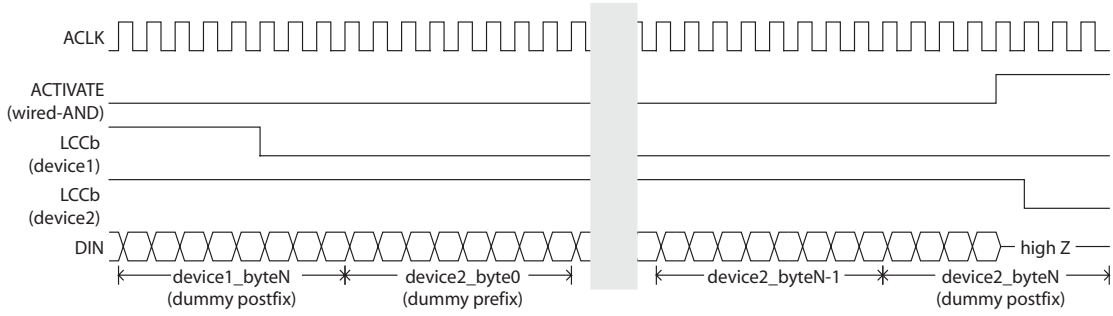


Figure 21 – Multiple Device Serial EPROM Boot Handoff

Figure 21 shows just two FPAA's in the configuration chain. The common (wired-AND) ACTIVATE signal pulls high only when the second device completes its configuration and releases it.

The initial sequence is the same as shown in Figure 18. During a multi-device configuration, the first several clocks of the configuration sequence for the second device provide the edges needed for the first device to complete its configuration. The first device therefore does not require a dummy postfix byte, but it is part of a standard configuration data set. The clocks associated with a dummy post-fix byte are however required at the end of the configuration load for the second device. AnadigmDesigner®2 provides these extra "for clocking only" prefix and postfix dummy bytes.

In Figure 20, the first device generates the analog clock (DCLK/16) which is used by both devices. Note the different settings of the mode pins to achieve this. (Please reference Figures 9 and 10 for further detail.)

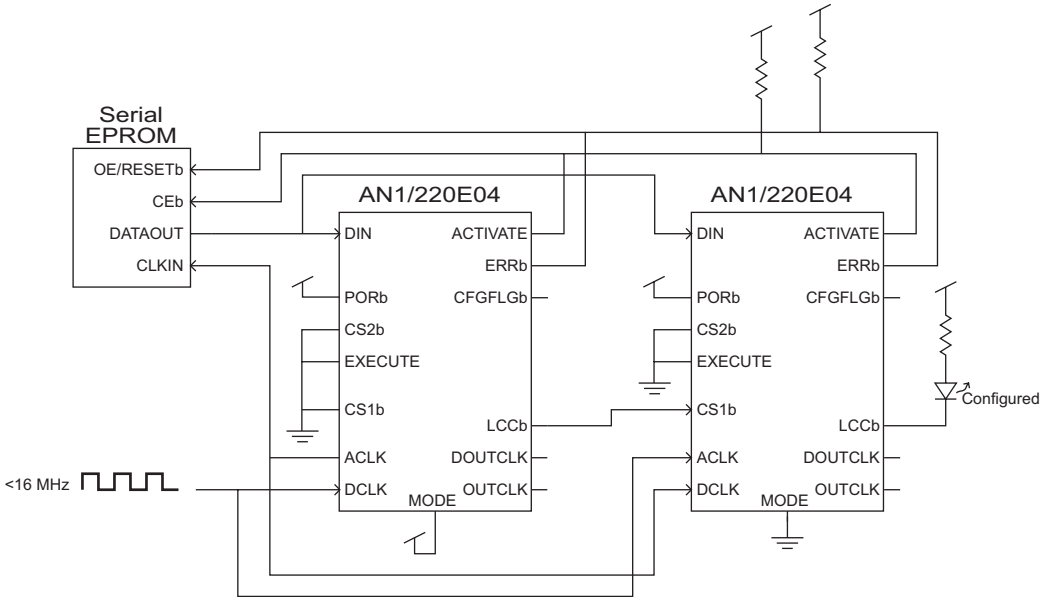


Figure 22 – Connecting Multiple FPAAs to a Single Configuration Memory - Driving DCLK

In Figure 22, the cross wiring of ACLK and DCLK allows both devices to configure using the same configuration clock (input clock/16), and allows both devices to use the higher frequency input clock to the first device as the analog master clock. For this to work, the analogue_clk_independent configuration data bit should be set high in the second device and low in the first device. Additional devices should be wired up as the second device and also have the analogue_clk_independent bit set high. In order to implement a system which requires a fast analog clock and which uses the on-chip crystal oscillator, the connection shown in Figure 23 should be used. The analogue_clk_independent bit in the second device should be set to allow the device to use the input on ACLK as the analog clock. Configuration data in the first device should be set to enable a buffered version of DCLK to be output on DOUTCLK. See Figure 9 for further detail.

For more information on how to set these configuration bits and other advanced features, please contact Anadigm technical support.

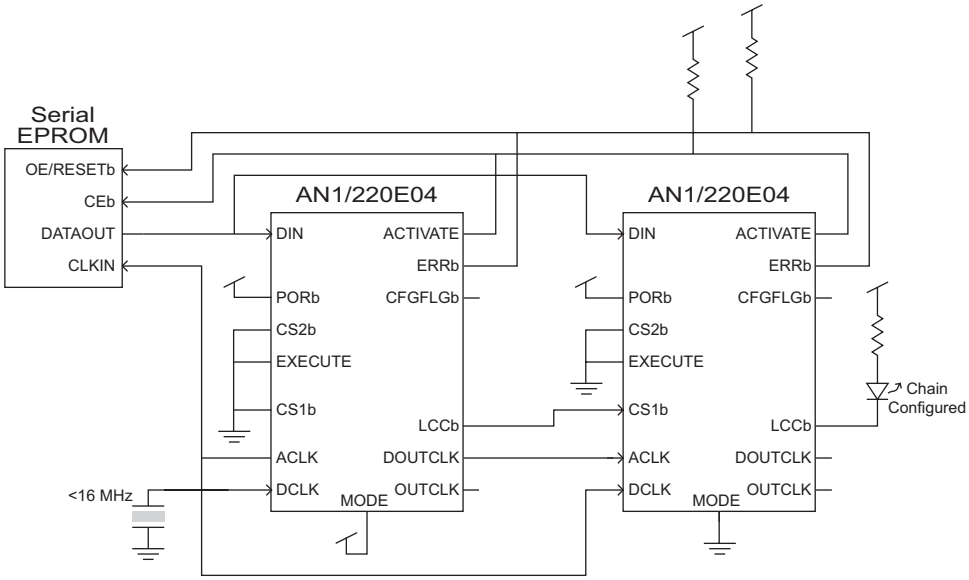


Figure 23 – Using DOUTCLK in a Multi-Device System

3.4 Booting from a Host Processor

In applications demanding on-the-fly adjustments to the analog circuitry, there will be a host microprocessor available to: perform the calculation of new circuit values, assemble these new values into a configuration data block and transfer that data block into the FPAA. The device's flexible configuration interface is designed to accept input from either serial memories or any of three major microprocessor interface types: Synchronous Serial Interface (SSI), Serial Peripheral Interface (SPI), or a conventional external peripheral bus interface.

3.4.1 Synchronous Serial Interface Connections (SPI and SSI)

As far as the device is concerned, the SPI and SSI interfaces are one in the same. There is a signal to indicate data is coming, a serial data line, and a serial data clock. The only real differences between SSI and SPI connections are the names various hosts assigns these signals and the frequencies with which the host can drive them. Functionally, these two connection schemes are the same.

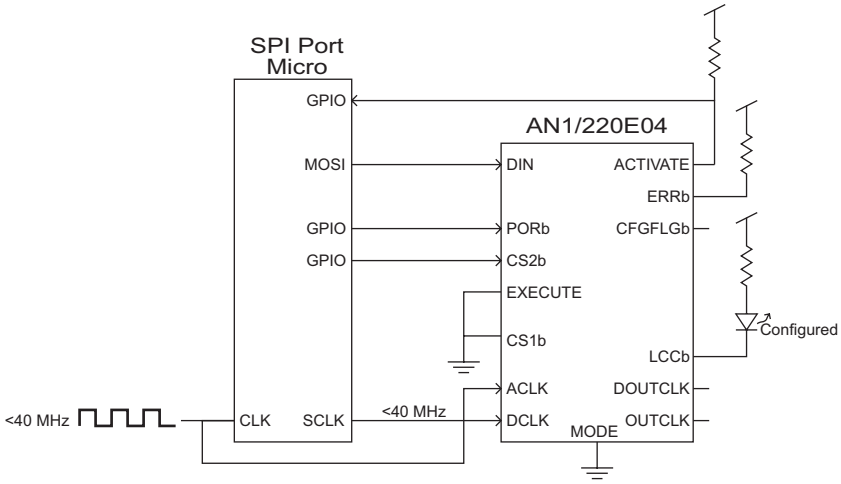


Figure 24 – A Typical Host SPI Port Connection

Microprocessors with SPI ports are quite a bit more common than those with SSI ports. The only down sides to using a SPI port is that they are considerably slower than SSI and as such are rarely supported by DMA transfer capability. SSI ports on the other hand often run at or near the microprocessor's bus speed and are sometimes supported by DMA channels.

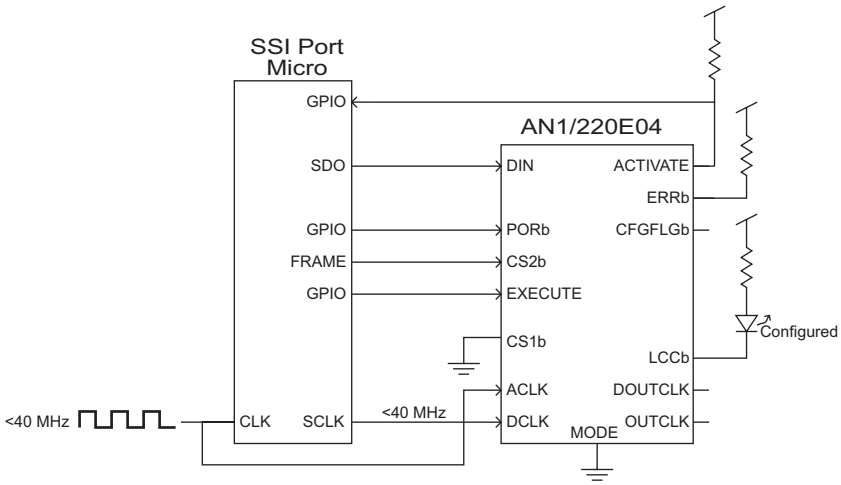


Figure 25 – A Typical Host SSI Port Connection

In each of these connection scenarios, the device's ACTIVATE line is fed back to the host's general purpose input/output (GPIO) pin to provide an indication that the configuration was successful. Figure 25, also shows an optional connection from host GPIO to the FPAA's EXECUTE input. More detailed discussion of EXECUTE is given in Section 3.1.14.

3.4.2 Typical Microprocessor Bus Connection

The configuration interface is synchronous but there is no requirement for the configuration clock to be uninterrupted. It is therefore possible to "clock" the configuration interface using the write strobe signal typical of most microprocessor's external data buses.

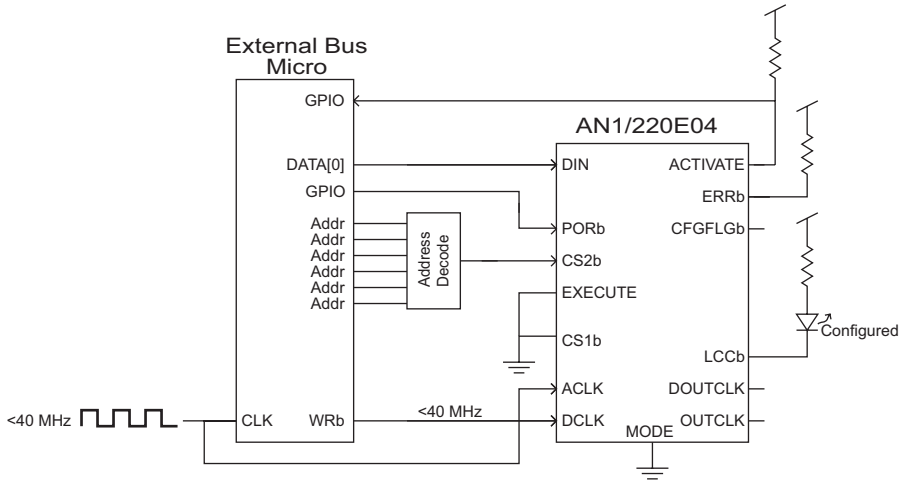
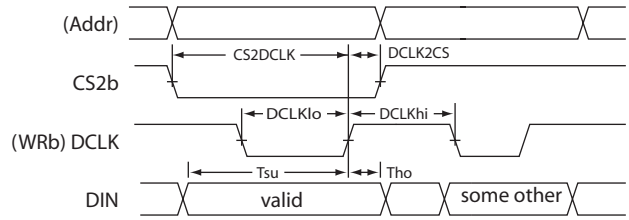


Figure 26 – Typical Microprocessor External Bus Connection



Symbol	Minimums	Description
CS2DCLK	2 nS	CS2b assertion prior to rising edge of DCLK
DCLK2CS	0 nS	CS2b de-assertion after rising edge of DCLK
DCLKlo	12.5 nS	minimum low period for DCLK
DCLKhi	12.5 nS	minimum high period for DCLK
Tsu	2 nS	Set up time for DIN with respect to DCLK rising edge
Tho	0 nS	Set up time for DIN with respect to DCLK falling edge

Figure 27 – Timing Sequence for Booting from a Host Microprocessor

The timing restrictions for SPI and SSI connections are the same as given above. The only significant difference in the peripheral bus connections is that DCLK is a strobe rather than a continuous clock.

There are several options available to drive CS2b. In the example shown above, some level of address decoding is accomplished with a third device (typically a PAL). Another option is to use a single high order address line. While this may not be an efficient use of your processor's external memory address space, it may be sufficient for your particular design. Many microprocessors provide chip select outputs and these too are usually suitable to the task of driving the device's CS2b input.

As with the SPI and SSI connections, ACTIVATE or ERRb can be monitored to confirm the configuration data transfer.

3.4.3 Advanced Feature - ACTIVATE

In multi-FPAA systems, it may be beneficial to prevent any of the FPAAs from going active until they have all received their configuration data. ACTIVATE is an open drain bi-directional pin controlling logic which achieves this function.

Consider Figure 20. Out of reset, each FPAA drives ACTIVATE low. Each device will continue to drive ACTIVATE low until its configuration is completed at which time it is released. Only when the ACTIVATE net pulls high (which in our example will only happen after the both devices receive their configuration) will the analog circuitry be allowed to go active.

ACTIVATE has an optional internal pull-up resistor that may be enabled via the device's configuration data set. In a multi-FPAA system, it is recommended that a single external pull-up be used.

3.4.4 Advanced Feature - Return Read Data Path (AN220E04 Only)

In most applications, the function of the LCCb pin is to signal that a configuration has been completed. For more advanced applications, a read back data path may be beneficial. These applications take advantage of the fact that once a Primary Configuration is complete, LCCb becomes a serial data output pin.

If the device is addressed for a read, LCCb sources the data out. If the device is not addressed for a read, then LCCb registers out the state of the CS1b input pin (with an 8 clock latency). Read data is presented on the LCCb pin of the addressed device after a delay of 16 clocks. Figure 32 shows several devices connected correctly for serial configuration and subsequent read back.

3.5 Configuration Protocol

Serial data, no matter how it comes into the device, must adhere to the configuration protocol defined in this section. AnadigmDesigner®2 constructs a configuration data file which adheres to this protocol so that even for the simplest case of self-booting from a serial EPROM, all the requisite information is contained in the serial data stream delivered to the device during configuration.

In dynamic applications, the host processor must not only determine the appropriate configuration data but also transfer that data to the device using the protocol defined herein.

There are two data formats which comprise the configuration protocol: Primary Configuration Format, and Update Format. Each is explained in detail in the following sections.

3.5.1 Primary Configuration Format & Byte Definitions

The Primary Configuration format is the format of the data that is generated by AnadigmDesigner®2 and is the format that must be used exactly once to configure the device for the first time after reset. Out of reset, all Shadow SRAM locations are reset to "zeros". A Primary Configuration is therefore only required to send data to Shadow SRAM locations requiring "ones". The LUT SRAM is not expressly reset to zero. The Primary Configuration is therefore also required to initialize the LUT SRAM if the LUT is intended to be used.

The Primary Configuration format is comprised of a Header Block followed by one or more Data Blocks. A header block contains a Sync byte, JTAG ID, ID1 and Configuration Control bytes. A Data Block contains data addressing information, a configuration data byte count and from 1 to 256 configuration data bytes, followed by error check byte.

	Data	Byte Name	Description
Header Block	11010101 D5	SYNC	Synchronization byte, always D5
	10110111 B7	JTAG ID BYTE 0	Bits [7:0] of JTAG Device ID - 0x300022B7 (or 0x300012B7)
	00100010 22	JTAG ID BYTE 1	Bits [15:8] of JTAG Device ID
	00000000 00	JTAG ID BYTE 2	Bits [23:16] of JTAG Device ID
	00110000 30	JTAG ID BYTE 3	Bits [31:24] of JTAG Device ID
	XXXXXXXX	ID1	ID1 Byte
	XXXXXXXX	CONTROL	Configuration Control Byte
Data Block (first)	11XXXXXX	BYTE ADDRESS	Starting Byte Address (DATA_FOLLOWS = 1)
	XXXXXXXX	BANK ADDRESS	Starting Bank address
	XXXXXXXX	DATA COUNT	Data byte count, a value of 00 instructs 256 bytes
	XXXXXXXX	DATA 0	Data byte to write to starting address + 0
	XXXXXXXX	DATA 1	Data byte to write to starting address + 1
	Remaining data bytes go in this region...		
	XXXXXXXX	DATA n	Data byte to write to starting address + n
	00101010 2A	ERR	Error check byte
Remaining data blocks go in this region...			
Data Block (last)	10XXXXXX	BYTE ADDRESS	Starting Byte Address (DATA_FOLLOWS = 0)
	XXXXXXXX	BANK ADDRESS	Starting Bank address
	XXXXXXXX	DATA COUNT	Data byte count, a value of 00 instructs 256 bytes
	XXXXXXXX	DATA 0	Data byte to write to starting address + 0
	XXXXXXXX	DATA 1	Data byte to write to starting address + 1
	Remaining data bytes go in this region...		
	XXXXXXXX	DATA n	Data byte to write to starting address + n
	00101010 2A	ERR	Error check byte

Figure 28 – Primary Configuration Data Stream Structure

3.5.2 Header Block

SYNC BYTE

The configuration logic always expects a synchronization header. For the Primary Configuration and Update formats, this sync header is always 11010101 (D5).

JTAG ID BYTE n

Every Anadigm® device type has a unique 32 bit JTAG ID associated with it, though not all of our products have a JTAG interface port. Requiring the JTAG ID to match during Primary Configuration is a way of ensuring that configuration data intended for another device does not get accidentally loaded. If a Primary Configuration is attempted in which the JTAG ID is not as expected, the device will assert ERRb and no data will be loaded into the array. Incorrect data can cause high stress conditions to exist within the device, possibly causing damage. The hex JTAG ID for the AN220E04 device is 0x300022B7. The hex JTAG ID for AN120E04 device is 0x300012B7.

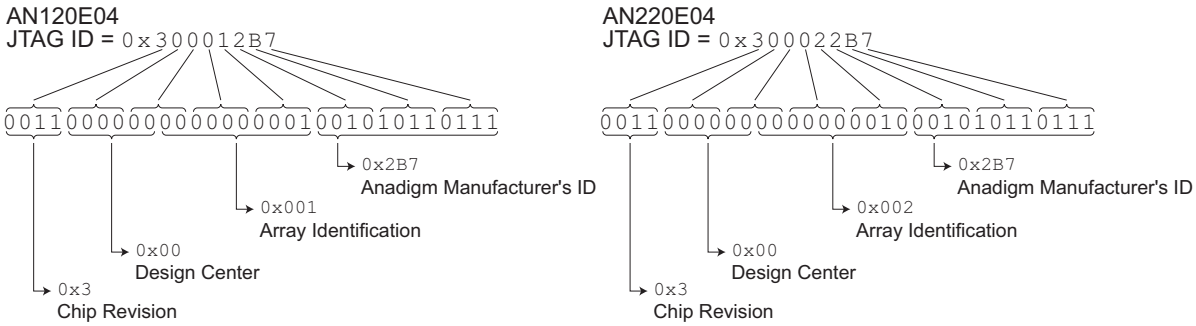


Figure 29 – The Components of the 32 Bit JTAG Identification Number

ID1 BYTE

The ID1 field establishes one of the two logical addresses for the device. ID1 is considered the primary logical address for the device. The alternate logical address (ID2) is not part of the Header Block, but rather it is established within the device's configuration data and is therefor delivered within a Data Block. Having logical addresses for every FPAA in the system allows the connection of many FPAAs in series (consuming no extra physical host connections) and once configured, communication only with the specifically addressed device(s). See section 3.5.5 for further details.

3.5.3 Data Block

CONTROL BYTE

Bit Number									
7	6	5	4	3	2	1	0		
0	0	0	0	0	1	0	1	Default bit values as generated by AnadigmDesigner®2	
								PULLUPS	**1:** Enable internal pull-ups. **0:** Disable internal pull-ups. This bit is used to enable internal pull-ups on the CFGFLGb and ACTIVATE pins. PULLUPS is sticky, i.e. Once set, it stays set until a device reset. If the pin is externally loaded then it is recommended that an external pull-up resistor be used instead of the internal. (Note - DIN pull-up is controlled by configuration data only.) (Note - ERRb always requires an external pull-up resistor. 10KΩ is the recommended value.)
								(Factory Reserved)	**1:** Factory Reserved setting. **0:** Normally set to “0”.
								ENDEXECUTE	**1:** At the end of the current transfer cycle, Shadow SRAM will be copied into Configuration SRAM. **0:** No action. (See below and section 3.1.14 further explanation.)
								SRESET	**1:** The device will perform a reset. **0:** No action. This bit allows the host to initiate a soft reset. The device will reset as soon this bit is latched.
								READ	**1:** Sets the device in read mode, Configuration SRAM and LUT only. **0:** Sets the device in write mode.
								STOP_READBACK	**1:** Stop any data read back from the device. **0:** Allow data read back from the device. This bit can be set during Primary Configuration or Update. If set, an internal flag is set which prevents all further data read backs. This internal flag can only be reset by re-powering the device, thereby destroying the SRAM contents. If any attempt to do a read back is made after this bit is set, then ERRb will drive low for 14 DCLK cycles and the device will be reset to a point where a Primary Configuration is required. In the AN120E04 device, this feature is superfluous.
								RESET_ALL	**1:** On an error, the ERRb output will pull low for 15 DCLK cycles and the device will be reset to a point where a Primary Configuration is required. **0:** On an error, the ERRb output will be pulsed low for a single DLCK cycle and the device will be reset to a point where only an Update is required. (Review section 3.1.7 for further explanation.)
								(Factory Reserved)	**1:** Factory Reserved setting. **0:** Normally set to “0”.

It is recommended that the default bit values of the Control Byte always be used. For more information on Configuration SRAM and LUT read back and other advanced features, please contact Anadigm technical support.

Data downloaded into the device is placed into Shadow SRAM. In order to keep any disruption of analog processing to a minimum, the transfer from Shadow SRAM to Configuration SRAM occurs in a single clock cycle. Using the default control byte value (in particular ENDEXECUTE = 1), this transfer will happen automatically at the completion of any configuration data download.

BYTE ADDRESS BYTE

Bit Number								
7	6	5	4	3	2	1	0	
			BYTE ADDRESS					Starting byte address of Shadow SRAM to be loaded.
		(Factory Reserved)						**1:** Factory Reserved setting. **0:** Should normally be set to “0”.
	DATA_FOLLOWS							**1:** A subsequent Data Block will be expected by the configuration logic. **0:** This block is presumed to be the final block of configuration data.
CONSTANT 1								**1:** Must always be set to “1”. **0:** Undefined operation.

BANK ADDRESS BYTE

Bit Number								
7	6	5	4	3	2	1	0	
BANK ADDRESS								Starting bank address of Shadow SRAM to be loaded.

The BYTE and BANK Address bytes taken together form the starting Shadow SRAM (or LUT SRAM) load address for the subsequent block of configuration data. The memory within the device is organized as 18 rows (banks) by 32 columns (bytes). No special handling is required to cross bank or byte boundaries, this is automatic. The address allocation of the device’s Shadow and LUT SRAM is shown in the figure below.

BANK ADDRESS	BYTE ADDRESS 1F 1E 1D 1C 1B 1A 19 18 17 16 15 14 13 12 11 10 0F 0E 0D 0C 0B 0A 09 08 07 06 05 04 03 02 01 00
00	Lower Auxiliary Shadow SRAM Bank
01	Upper Auxiliary Shadow SRAM Bank
02	CAB 1 Lower Shadow SRAM Bank
03	CAB 1 Upper Shadow SRAM Bank
04	CAB 2 Lower Shadow SRAM Bank
05	CAB 2 Upper Shadow SRAM Bank
06	CAB 3 Lower Shadow SRAM Bank
07	CAB 3 Upper Shadow SRAM Bank
08	CAB 4 Lower Shadow SRAM Bank
09	CAB 4 Upper Shadow SRAM Bank
0A-0F	(Factory Reserved Address Range)
10	Look Up Table SRAM Bank 0
11	Look Up Table SRAM Bank 1
12	Look Up Table SRAM Bank 2
13	Look Up Table SRAM Bank 3
14	Look Up Table SRAM Bank 4
15	Look Up Table SRAM Bank 5
16	Look Up Table SRAM Bank 6
17	Look Up Table SRAM Bank 7
18-FF	(Factory Reserved Address Range)

Figure 30 – AN1/220E04 Memory Allocation Table

DATA COUNT BYTE

Setting this field to a value of 0x00 signifies that 256 data bytes follow in this data block. Setting this field to any integer value between 1 and 255 signifies that exactly that many data bytes follow. This byte count only represents the number of configuration data bytes that follow (data bytes destined for the Shadow SRAM or LUT SRAM); the count does not include the Error check byte.

DATA BYTE

Configuration data bytes. This is the data that gets loaded into the Shadow SRAM or LUT SRAM, starting at the address defined in BYTE and BANK address bytes defined just above. There may 1 up to 256 data bytes per block.

ERR BYTE

The only byte expected after the DATA bytes is the ERR constant 2A. If any other value is read in, ERRb will assert and the configuration process will be aborted. Reconfiguration will be required as described in section 3.1.7.

3.5.4 Update Format (AN220E04 Only)

Once a Primary Configuration has been completed, there is no longer any requirement for the host to transmit out JTAG ID information or reprogram the ID1's of the devices along the communication path. (A configuration data stream doing so would be considered an error, and devices would assert ERRb.) The host now only needs to send out a sync header and a valid ID for the target device or devices. The remainder of the information is just as described above in section 3.5.1.

As with Primary Configuration, it is not a requirement of the Update format to contain a complete data set for the device. Partial reconfiguration of the device is supported. It is most often the case that only a few Shadow SRAM or LUT SRAM addresses need new data. The Update format provides a quick and compact method for moving this new data into the device.

	Data	Byte Name	Description
Header Block	11010101 D5	SYNC	Synchronization byte, always D5
	XXXXXXXX	TARGET ID	ID1, ID2, or 0xFF - Logical address of the target device(s).
	XXXXXXXX	CONTROL	Configuration Control Byte
Data Block (first)	11XXXXXX	BYTE ADDRESS	Starting Byte Address (DATA_FOLLOWS = 1)
	XXXXXXXX	BANK ADDRESS	Starting Bank address
	XXXXXXXX	DATA COUNT	Data byte count, a value of 00 instructs 256 bytes
	XXXXXXXX	DATA 0	Data byte to write to starting address + 0
	XXXXXXXX	DATA 1	Data byte to write to starting address + 1
	Remaining data bytes (if any) go in this region...		
	XXXXXXXX	DATA n	Data byte to write to starting address + n
	XXXXXXXX or 00101010 2A	CRC_MSB or ERR	Most significant byte of CRC16 error code or (depending on Bit 5 of BYTE ADDRESS) Basic error check byte
	XXXXXXXX	CRC_LSB	Least significant byte of CRC16 error code (if used)
Remaining data blocks (if any) go in this region...			
Data Block (last)	10XXXXXX	BYTE ADDRESS	Starting Byte Address (DATA_FOLLOWS = 0)
	XXXXXXXX	BANK ADDRESS	Starting Bank address
	XXXXXXXX	DATA COUNT	Data byte count, a value of 00 instructs 256 bytes
	XXXXXXXX	DATA 0	Data byte to write to starting address + 0
	XXXXXXXX	DATA 1	Data byte to write to starting address +1
	Remaining data bytes (if any) go in this region...		
	XXXXXXXX	DATA n	Data byte to write to starting address + n
	XXXXXXXX or 00101010 2A	CRC_MSB or ERR	Most significant byte of CRC16 error code or (depending on Bit 5 of BYTE ADDRESS) Basic error check byte
	XXXXXXXX	CRC_LSB	Least significant byte of CRC16 error code (if used)

Figure 31 – Update Data Stream Structure

TARGET ID - Special Feature (AN220E04 Only)

ID1 (the primary logical address) is established for the device during the Primary Configuration. The configuration data itself establishes ID2 (the alternate logical address) for the device. Once the Primary Configuration is complete, a device will respond to any Update format configuration data stream which contains either: a matching ID1 value, a matching ID2 value or 0xFF in the TARGET ID byte field. The hex address of 0xFF serves as a global ID; all devices respond to this ID.

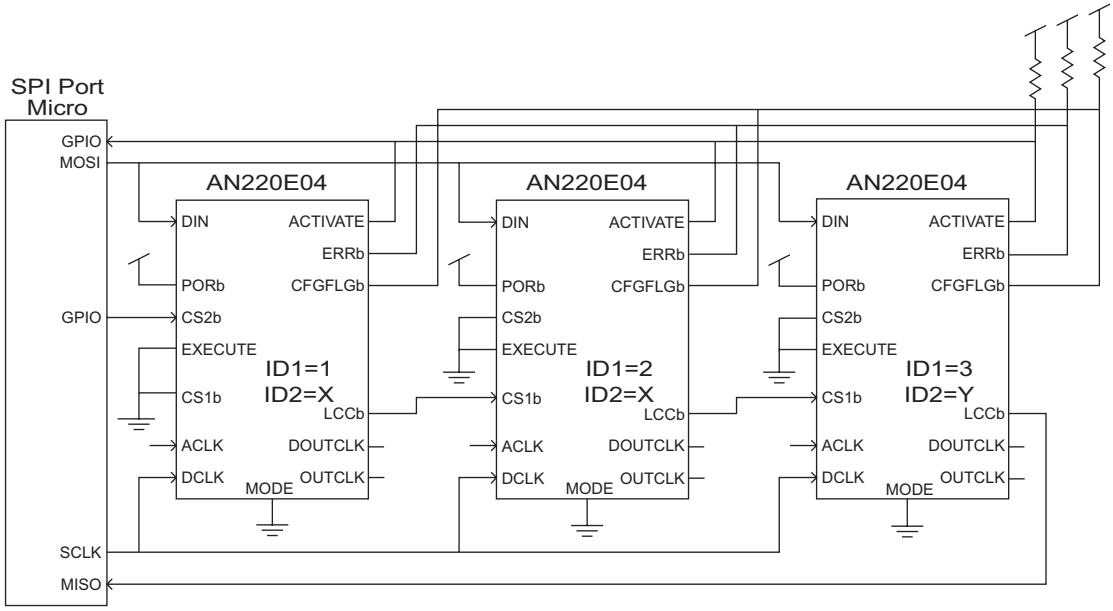


Figure 32 – Configuring Multiple Devices from a Host Processor

Figure 32 shows a valid connection and configuration example for multiple FPAA's being hosted from a single SPI port. During Primary Configuration, each device in the chain received a unique ID1 and a non-unique ID2. The ID1's were assigned via the ID1 fields of the Primary Configuration data streams. The ID2's were established within the configuration data sets downloaded into each of the devices. Once Primary Configurations are complete for all of these devices, each will respond to the host SPI port only if the TARGET ID field of the Update configuration data stream contains either its ID1 or ID2 identifiers (or 0xFF).

Assume that two of the devices perform identical analog functions "X", and the third a unique function "Y". Also assume that the X and Y configuration data sets contain ID2 values of X and Y respectively. During the Primary Configuration, the host processor assigned the first device an ID1 of 1 and filled that device with the X configuration. Likewise, the ID1=2 device also got the X configuration. The ID1=3 device got the Y configuration. Once these Primary Configurations are complete and analog operations go active, the host processor may alter via Update both X devices concurrently by using TARGET ID = X rather than sequentially by addressing TARGET ID = 1 then TARGET ID = 2. If the host instead uses 0xFF in the TARGET ID field, then all three devices will concurrently accept the subsequent reconfiguration data.

3.5.5 Configuration Examples

The following examples assume a hosted interface. Data must be shifted into the FPAA most significant bit first. White space and comments are included only to improve readability for these examples.

Primary Configuration Format Example

```
00000000    //40 clocks are required to be sent to complete the power-up
00000000    //reset sequence. This is usually accomplished by sending out 5
00000000    //bytes of "don't care" prefix data. After the 40th clock, the
00000000    //configuration logic is ready.
00000000

11010101    //0xD5 is the required sync header.
10110111    //0xB7 is Least Significant Byte of JTAG ID word. (AN220E04)
00100010    //0x22 is byte 2 of JTAG ID word.
00000000    //0x00 is byte 3 of JTAG ID word.
00110000    //0x30 is Most Significant Byte of JTAG ID word.

00000001    //User assigns any Chip ID except 0xFF.

00000101    //Control Byte - PULLUPS are enabled.
            //ENDEXECUTE - Transfer Shadow SRAM to Configuration
            //SRAM as soon as this download is complete.

11000000    //Constant 1, DATA_FOLLOWS 1, start BYTE address is 0
00000000    //The starting BANK address is 0

00000000    //0x00 byte count field means 256 data bytes follow.

datadata    //The first configuration data byte.
datadata    //The second configuration data byte.
...
datadata    //the 256th configuration data byte.

00101010    //0x2A is the Basic error checking constant expected.

........    //This is the region that the other blocks of data
            //need to be sent to completely fill the Shadow SRAM.
            //These blocks do not need to be prefaced by additional
            //clocks, nor do they require a JTAG ID, ID1
            //or Control bytes. These intermediate blocks all have
            //the same form as the final block shown below. The
            //important point to note is that only the final
            //block of Primary Configuration has the DATA_FOLLOWS
            //bit cleared in the BYTE ADDRESS byte.

10011110    //DATA_FOLLOWS is cleared to 0, this means that at
            //the conclusion of the transfer of this final block,
            //Shadow SRAM will get copied into Configuration SRAM,
            //with no additional action required by the host.
            //0x1E is the starting BYTE address.

00010111    //0x17 is the starting BANK address.

00000010    //0x02 is byte count for this particular last block.

datadata    //Second to the last configuration data byte.
datadata    //The Last configuration data byte.

00101010    //0x2A is the Basic error checking constant expected.
00000000    //8 clocks are required by the configuration state
            //machine to finish the transfer. This is usually
            //accomplished by sending out a single byte of "don't
            //care" data.
```

Update Format Example (AN220E04 Only)

```
11010101    //0xD5 is the required sync header.

00000001    //TARGET ID - The ID1 or ID2 value of the target device
            //or the universal target ID of 0xFF.

00000101    //Control Byte - ENDEXECUTE and PULLUPS are enabled.

10011110    //DATA_FOLLOWS is cleared to 0, so the configuration
            //logic will expect no more data after this final block
            //and because ENDEXECUTE is set = 1 in the Control Byte
            //Shadow SRAM will get copied into Configuration SRAM
            //as soon as the data block is download with no
            //additional action required by the host.
            //0x1E (decimal 30)is the starting BYTE address.

00000011    //0x03 is the starting BANK address.

00000011    //0x03 byte count field means 3 data bytes follow.

datadata    //The 1st updated data byte goes to bank 3 byte 30
datadata    //The 2nd updated data byte goes to bank 3 byte 31
datadata    //The 3rd updated data byte goes to bank 4 byte 0

00101010    //0x2A is the Basic error checking constant expected.

00000000    //8 "don't care" bits to provide the necessary clocks
            //to complete the load. Because the EXECUTE bit was
            //set on this block's control byte, the immediate
            //transfer from Shadow RAM to the Configuration RAM
            //will occur here.
```

The 8 clocks at the end of each of these configurations are necessary only to complete the transfer at that time. If it is not critical to complete the transfer at that particular moment, then the clocking associated with any subsequent Update block will be sufficient to complete the transfer. With no clocks, the configuration state machine simply freezes in place. There are no "unsafe" states.

3.5.6 Configuration Clocking Considerations

The state machines within the device's configuration logic require a clock in order to function correctly. In most of the configuration connection examples above, the clock is not free running so in addition to adhering to the protocols described, the host processor must also provide clocks to allow the state machines to complete their reset and data transfer sequences. If the appropriate clocks are not provided, the state machines will idle and the reset or configuration will not complete as expected.

Out of power-up, the configuration logic requires 40 clocks to complete its reset sequence. After any configuration or reconfiguration, the configuration logic requires 8 clocks to complete the data transfer.

4 Mechanical

4.1 Package Pin Out

1	I4PA	Analog IN+	Analog multiplexer input signals. The multiplexer can accept 4 differential pairs or 4 single ended input connections.
2	I4NA	Analog IN-	
3	O1P	Analog OUT+	
4	O1N	Analog OUT-	
5	AVSS	Analog VSS	Analog Power
6	AVDD	Analog VDD	Analog Power
7	O2P	Analog OUT+	
8	O2N	Analog OUT-	
9	I1P	Analog IN+	
10	I1N	Analog IN-	
11	I2P	Analog IN+	
12	I2N	Analog IN-	
13	SHIELD	Analog VDD	Low noise VDD bias for capacitor array n-wells
14	AVDD2	Analog VDD	Analog Power
15	VREFMC	Vref	Attach filter capacitor for VREF-
16	VREFPC	Vref	Attach filter capacitor for VREF+
17	VMRC	Vref	Attach filter capacitor for VMR (Voltage Main Reference)
18	BVDD	Analog VDD	Analog Power for Bandgap Vref Generators
19	BVSS	Analog VSS	Analog Ground for Bandgap Vref Generators
20	CFGFLGb	Digital IN	In multi-device systems... 0, Ignore incoming data (unless currently addressed) 1, Pay attention to incoming data (watching for address)
		Digital OUT (Open Drain)	0, Device is being reconfigured Z, Device is not being reconfigured
21	CS2b	Digital IN	0, Chip is selected 1, Chip is not selected
22	CS1b	Digital IN	(Prior to completion of a Primary Configuration) 0, Allow configuration to proceed 1, Hold off configuration
		Digital IN	(After completion of a Primary Configuration) Data input pin, read back pass through port for multi-device system
23	DCLK	Digital IN	drive with < 40 MHz external configuration clock, or attach a 12, 16, 20, or 24 MHz crystal
24	SVSS	Digital VSS	Digital Ground - Substrate Tie
25	MODE	Digital IN	0, select clock support for synchronous serial interface 1, select clock support for SPI & FPGA EPROM interface
26	ACLK / SPIP	Digital IN	MODE = 0, Analog Clock (Switched Capacitor Clock) < 40 MHz
		Digital OUT	MODE = 1, SPI EPROM or Serial EPROM Clock
27	OUTCLK / SPIMEM	Digital OUT	During power-up, sources SPI EPROM initialization command string. After power-up, sources selected internal analog clock or comparator output.
28	DVDD	Digital VDD	Digital Power
29	DVSS	Digital VSS	Digital Ground
30	DIN	Digital IN	Serial Configuration Data Input
31	LCCb	Digital OUT	0, Local configuration complete 1, Local configuration is not complete Once configuration is completed, it is a delayed version (8 clock cycles) of CS1b or if the device is addressed for read, it serves as serial data read output port.
32	ERRb (10 kΩ p/u required)	Digital IN	0, Initiate Reset (hold low for 15 DCLK's) 1, No Action
		Digital OUT (Open Drain)	0, Error Condition Z, No Error Condition
33	ACTIVATE	Digital IN	0, Hold off completion of configuration Rising Edge, Allow completion of configuration
		Digital OUT (Open Drain)	0, Device has not yet completed Primary Configuration Z, Device has completed Primary Configuration
34	DOUTCLK	Digital OUT	Buffered version of DCLK. Inactive (floats) until its associated configuration bit is set. If unused this pin must be left floating.
		Digital IN	(Factory reserved test input. Float if unused.)
35	PORb	Digital IN	0, Chip Held in reset state Rising Edge, re-initiates power on reset sequence, 30 mS to complete
36	EXECUTE	Digital IN	0, No Action. This pin should normally be tied low. 1, Transfer Shadow SRAM into Configuration SRAM, depending on configuration settings
37	I3P	Analog IN+	
38	I3N	Analog IN-	
39	I4PD	Analog IN+	Analog multiplexer input signals. The multiplexer can accept 4 differential pairs or 4 single ended input connections.
40	I4ND	Analog IN-	
41	I4PC	Analog IN+	
42	I4NC	Analog IN-	
43	I4PB	Analog IN+	
44	I4NB	Analog IN-	

4.2 Recommended PCB Design Practices

The device is designed to perform with very few external components required. However, there is no fighting physics and some filtering capacitors are required for both the supply rails as well as the internally generated voltage references.

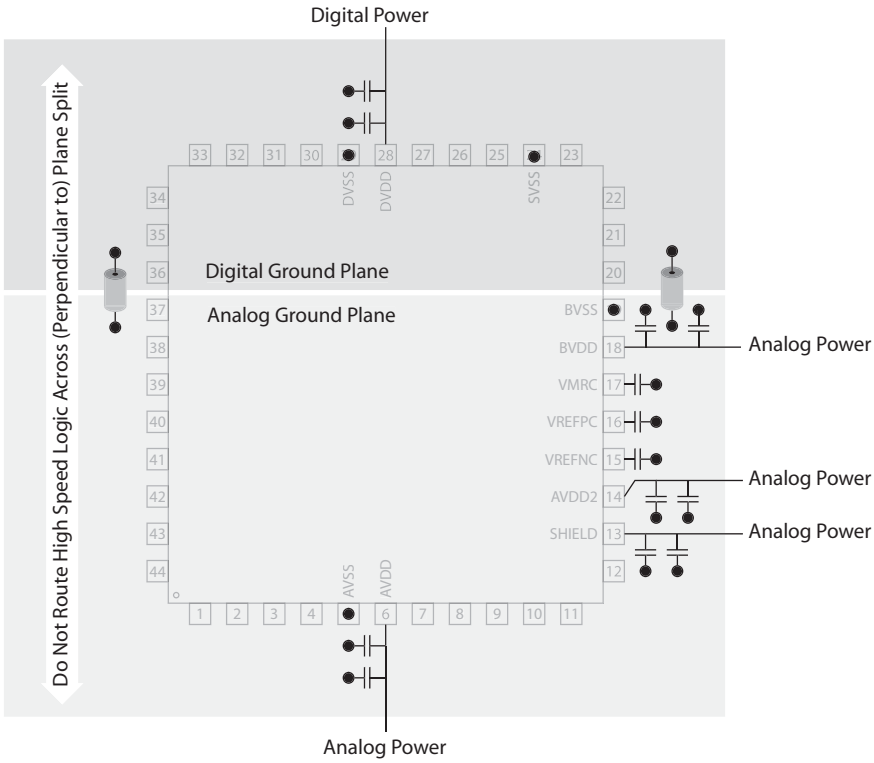


Figure 33 – Basic Guidelines for Optimal PCB Design (Ground Planes)

Your PCB design should include some of the following features to ensure good separation between the digital and analog signal environments in your system. Good PCB design practices dictate that the digital and analog power and ground planes be separated. It is important to maintain these planes at the same basic potential but care should be exercised to prevent the usual noise of a digital plane from coupling onto the analog plane. In Figure 33, the electrical connection between the two planes is made at only two points, through Ferrite bead choked wire. The Ferrite beads act as low pass filters.

As with any mixed signal board design, it is good practice to keep digital signals (especially digital signals with high edge rates) routed only over areas where digital power and ground planes underlay. Care should be exercised to never route a high edge rate digital signal perpendicular to a plane split. Doing so will cause a noise wavefront to launch (left and right) onto both planes along the split.

It is recommended that the digital supply DVDD be bypassed to DVSS using ceramic capacitors. A .1 µF capacitor in parallel with a .01 µF capacitor is usually sufficient. The capacitor connections to the device should be made as close as practical to the package to reduce detrimental inductance. This same bypassing scheme will work sufficiently for BVDD - BVSS, AVDD - AVSS, AVDD2 - AVSS, SHIELD - AVSS pairs as well.

For more information logon to:
www.anadigm.com