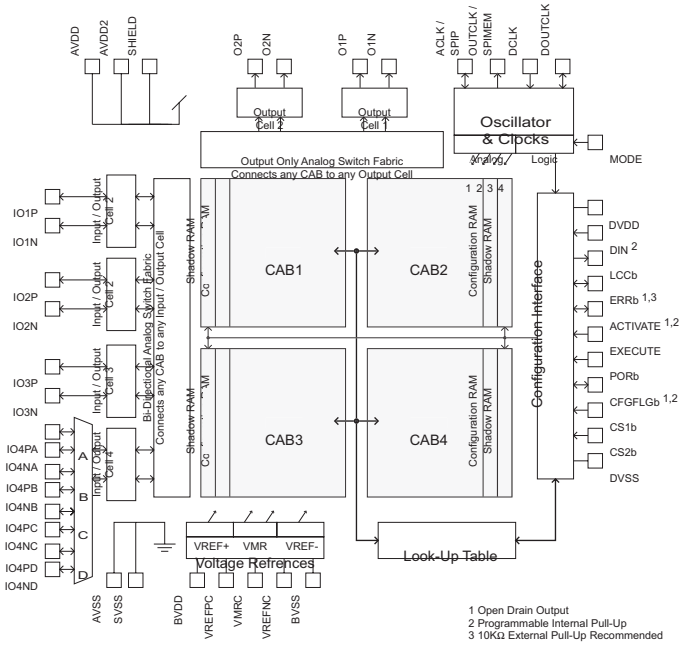
# AN22x series
# AN12x series
## AnadigmVortex FPAA Family User Manual

AnadigmVortex is the second generation field programmable analog array (FPAA) device family from Anadigm. Nine members of the AnadigmVortex family are currently shipping providing a range of solutions to meet your analog signal processing requirements.

| Static Configuration | Dynamically Reconfigurable | Input Cells | In/Out Cells | Output Cells | CABs | |
|---|---|---|---|---|---|---|
| AN120E04 | AN220E04 | 4 | - | 2 | 4 | Specialized input cell features |
| AN121E04 | AN221E04 | - | 4 | 2 | 4 | Bi-directional input/output cells |

AN12x devices are best suited for are geared towards high-volume applications requiring consolidation of discrete analog functionality. The configuration interface of the AN22x devices is enhanced to accommodate dynamic reconfiguration - a breakthrough capability that allows analog functions to be integrated within the system and controlled by companion processor




AnadigmVortex devices consist of a 2 x2 or 1x2 matrix of fully Configurable Analog Blocks (CABs), surrounded by a fabric of programmable interconnect resources. Fully differential signal paths ensure high fidelity operation. The second generation AnadigmVortex architecture provides a significantly improved signal-to-noise ratio as well as higher bandwidth.

These devices accommodate non-linear functions such as sensor response linearization and arbitrary waveform synthesis.

The ANx27 SonicMaster™ series is specially optimized for optimal performance through the audio band, providing never before possible audio system design options.

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
- High precision operation despite system degradation and aging
- Eliminating the need to source and maintain multiple inventories of product
- Ability to implement multiple chip configurations in a single device and to adapt functionality in the field

*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 without 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.

## Architecture Overview



## Analog Architecture Details



## Configuration Interface



## Mechanical

# 1 Architecture Overview

Anadigm offers the AN12x and the AN22x series Field Programmable Analog Array (FPAA) devices. The AN12x devices can be reprogrammed as many times as desired, however the device must first be reset before issuing another configuration data set. Once a Primary Configuration is complete; the configuration interface of the AN12x device ignores all further input. No further data writes are accepted unless a reset sequence is first completed. The AN22x devices are dynamically reconfigurable; the behavior of the FPAA can modified partially or completely while operating.

Dynamic Reconfiguration available on the AN22x devices, 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 signal processing configuration happens in a single clock cycle. Dynamic Reconfiguration in the AN22x device allows the user to develop innovative analog systems that can be updated (fully or partially) in real-time.

The FPAA contains either 2 or 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 CABs have access to a single Look Up Table (LUT) which offers a 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, and construct 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.

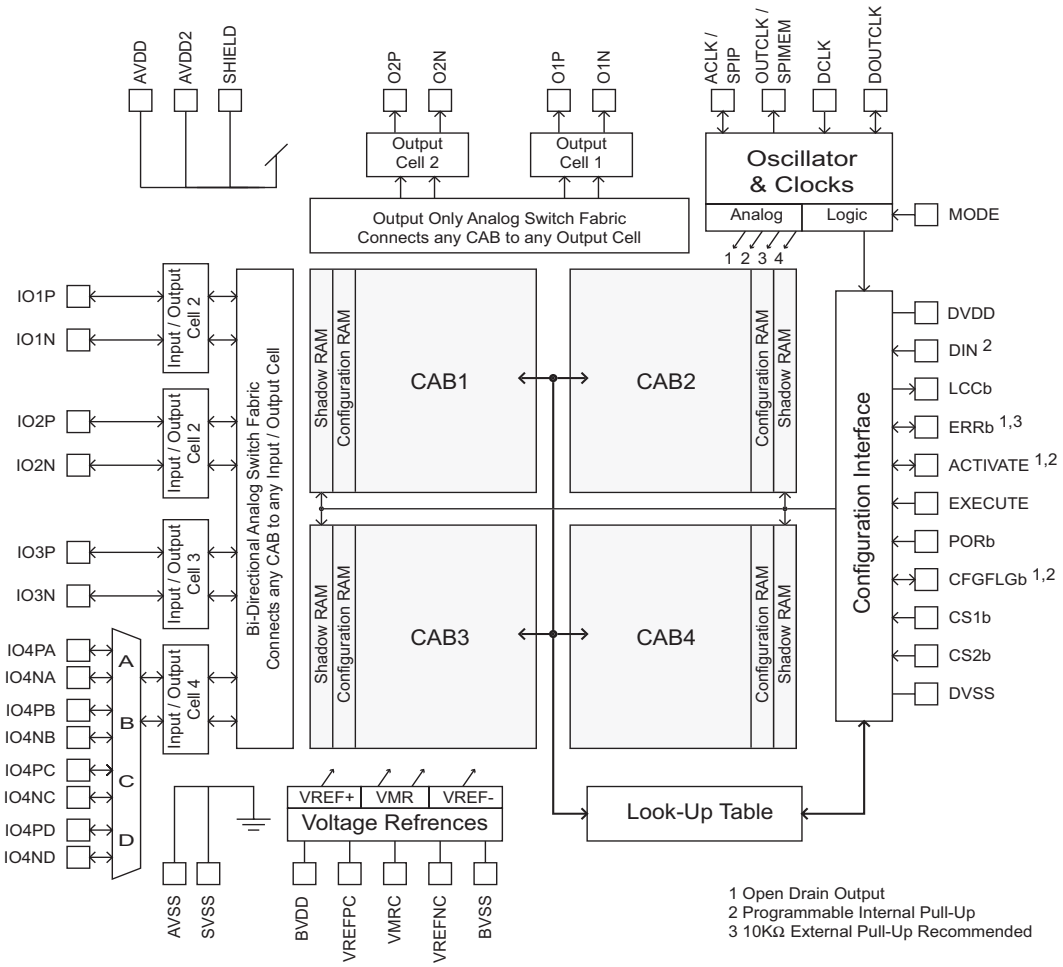



*Figure 1 – Chip Overview for a Typical Four CAB AnadigmVortex FPAA*

Analog inputs signals are routed into the FPAA core via any of the four Input Cells. The Input Cells can accept either differential pair or single-ended signals. The fourth Input Cell has a special front end multiplexer which allows for the connection of up to four differential pairs or eight single ended signals. ANx20 and ANx21 devices provide optional active signal processing elements in the Input Cells which provide programmable gain and anti-aliasing filtering. All the Input Cells also provide a direct input path to the FPAA core.

The ANx20 Input Cell provides only an input path; all other family members also provide a direct output path.

Analog output signals are routed out of the FPAA core via either of the two Output Cells or through the Input Cells direct output option. ANx20 and ANx21 devices provide optional active signal processing elements in the Output Cells which provide programmable gain and reconstruction filtering. The Output Cells also provide a digital output path used for comparator and SAR results data.

The FPAA 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 AN22x devices 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 AN12x devices.

The architecture includes a highly flexible digital configuration interface. The configuration interface is designed to work in stand-alone mode by connecting to either a common SPI type serial EPROM. In this mode, after the device powers up, it will automatically load its configuration data from the EPROM and begin analog signal processing.

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 compatible slave. The configuration interface also allows multiple devices to be easily connected together to build up larger analog processing systems.

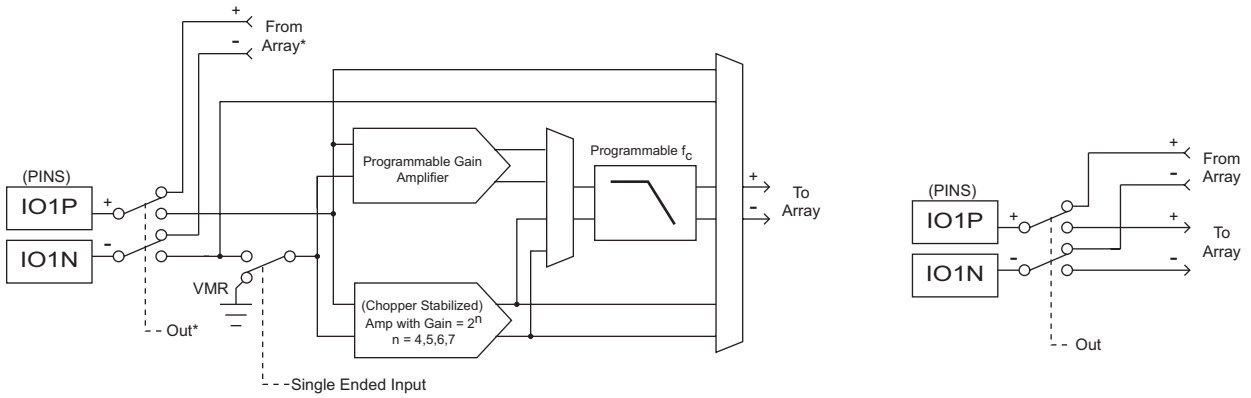
# 2 Analog Architecture Details

## 2.1 Input Cell

Each Input Cell contains a collection of resources which allow for high fidelity connections to and 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.

| Device Number | Unique Input Cell Features |
|---|---|
| ANx20 | Input only. Chopper stabilized low offset input amplifier with programmable gain, Standard input amplifier with programmable gain, Programmable anti-aliasing input filter, Direct input |
| ANx21 | Bidirectional. Chopper stabilized low offset input amplifier with programmable gain, Standard input amplifier with programmable gain, Programmable anti-aliasing input filter, Direct input, Direct output |
| ANx22 | Bidirectional. Direct input, Direct output |
| ANx27 | Bidirectional. Direct input, Direct output |

A single ended signal can be used as an input to the 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).




* The output path is not available on the ANx20 devices.

*Figure 2 – Fully Featured Input Cell for ANx20 & ANx21, and Simplified ANx22 & ANx27*

As with any sampled data system, it may sometimes be necessary to low pass filter the incoming signal to prevent aliasing artifacts. The input path of the ANx20 and ANx21 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. Where input signal frequencies are higher, Anadigm does recommend the use of external anti-aliasing.

A second unique input resource available within each ANx20 and ANx21 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 CABs. Single-ended input signals must use either the amplifier or the anti-alias filter in order to get the required single to differential conversion. The programmable gain amplifier, the chopper stabilized amplifier and the programmable anti-aliasing filter are all resources available only on the input signal path.

When the Input Cell is used as a bypass mode input or as an output, the connection is direct and unbuffered. There are no active circuit elements available in the Input Cell when it is configured for either input or output bypass.

ANx22 and ANx27 devices have streamlined Input Cells. These devices offer only the direct unbuffered signal paths into and out of the CABs.

### 2.1.1 Special Consideration for Bypass Outputs

When using either an Output Cell or an Input Cell in bypass output mode, special care must be taken to not overload the device. In bypass mode, there are no buffers between the CAB's signal source and the device's output pins.

The CAB op-amps are not designed to drive low impedances. Also, too much load capacitance will destabilize the CAB op-amps. The minimum recommended external load resistance should be not less than 100 KΩ and the maximum external load capacitance should be not more than 100 pF. When using bypass mode outputs, characterization of the final system is essential.

## 2.2 Muxed Analog Input / Output

There is a bi directional multiplexer available in front of one of the Input Cells. This allows the physical connection of 8 single ended inputs, 4 differential pair inputs, or 4 differential pair output loads at once, though only one source or load at a time can be processed by the FPAA. As with the regular Input Cells, the optimal input 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.

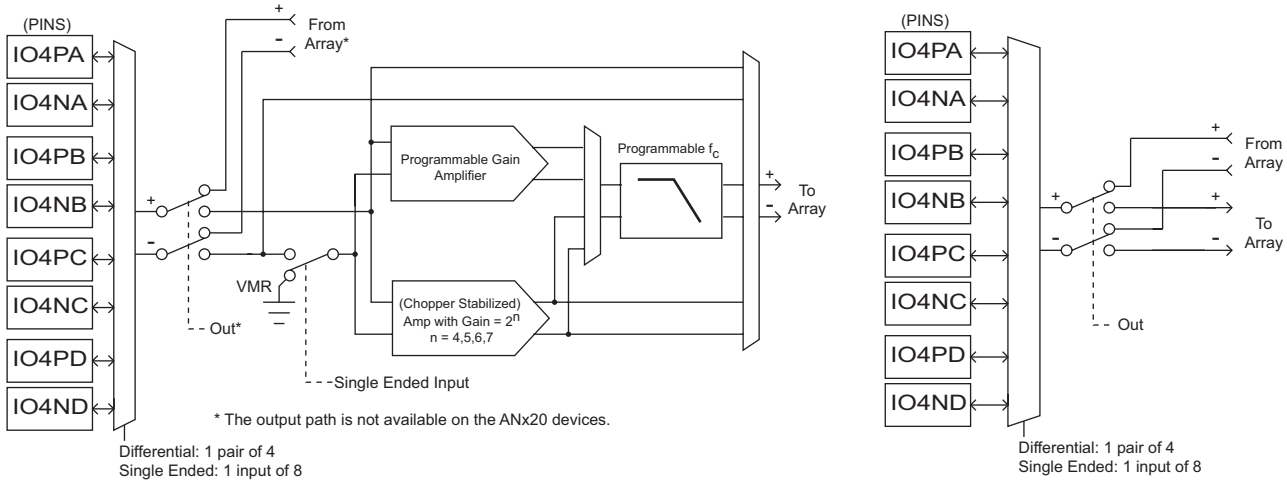



*Figure 3 – Fully Featured Muxed Input Cell for ANx20 & ANx21, and Simplified ANx22 & ANx27*

## 2.3 Output Cell

Like the Input Cells, the Output Cells are designed 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.

| Device Number | Unique Output Cell Features |
|---|---|
| ANx20 | Direct Output, Unity Gain Buffer with Programmable Gain Reconstruction Filter |
| ANx21 | Direct Output, Unity Gain Buffer with Programmable Gain Reconstruction Filter, Digital Output (SAR and Comparator) |
| ANx22 | Direct output |
| ANx27 | Direct output |

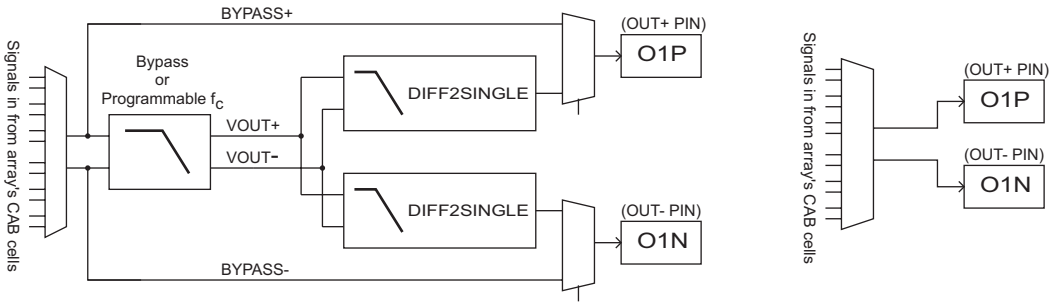



*Figure 4 – Fully Featured Output Cell for ANx20 & ANx21, and Simplified ANx22 & ANx27*

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

It may be desirable to route the core analog signals to the outside world with no additional buffering or filtering. The ANx20 and ANx21 Output Cells have bypass paths which allow the core signals to come out with no further processing or buffering. For special considerations governing the use of bypass mode outputs, see Section 2.1.1.

The ANx20 and ANx21 Output Cells contain 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 2$^{nd}$ order reconstruction filter. In this function, it smooths out the sampling induced stair step 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.

ANx22 and ANx27 devices have streamlined Output Cells. These devices offer only the direct unbuffered signal paths out of the CABs.

## 2.4 Configurable Analog Block

Within the FPAA, there are two (AN221E02) or four (all other family members) 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).

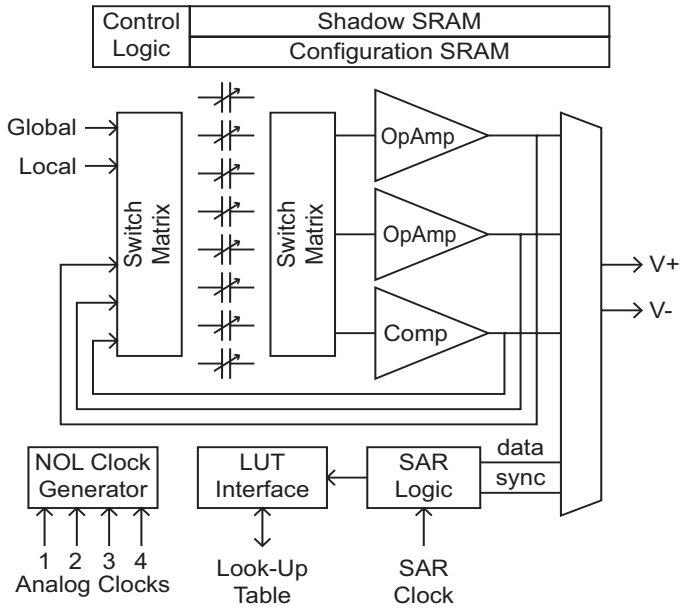



*Figure 5 – Overview of a Configurable Analog Block*

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 AN22x devices allows reconfiguration. While an AN22x 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 AN12x devices must be reset between complete configuration loads and do 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 op-amps 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 large bank of 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%.

There is a second switch matrix used to complete the circuit topology by making the appropriate connections. There are two op-amps 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 converter. Routing the SAR'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 8 bit output or from a special 8 bit LUT counter. The functional description of the SAR 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 Operation

Circuitry is included within the CAB which allows the construction of an 8-bit Successive Approximation Register (SAR) type analog converter. The SAR 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 and should not exceed 250 KHz. 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 result can be routed to either the LUT's address port or back into its host CAB. The most common use of the SAR 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 conversion. The SAR 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 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 conversion byte is routed directly back into the Configuration SRAM of its host CAB, self modifying circuits can be constructed.

The SAR may also be used to generate a serial data stream and an accompanying sync pulse. These two signals can be routed to either of the two dedicated output cells (configured in Digital Output mode). Only on the ANx21E04 devices offer this direct access to the SAR circuitry.

## 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 band gap 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

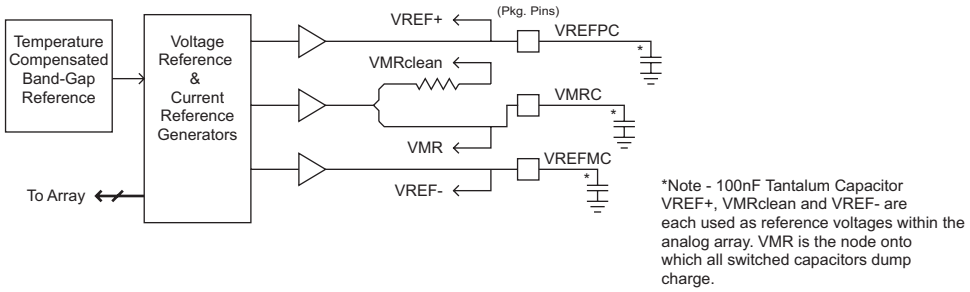



*Figure 6 – Voltage Reference and Bias Current Generation*

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 op-amps within the CABs. This quiet version of VMR is used by the op-amps 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 section 3.5.5 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 3.5.2 for complete details on this feature.

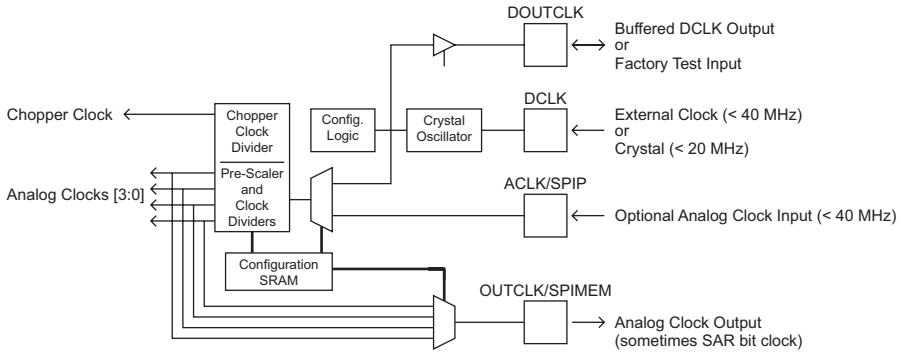



*Figure 7 – Clock Features and Clock Domains*

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 / Output 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.

## 2.9 Unique CAM Library Features

There are some unique CAM Library elements in support of the hardware differences between the members of the AnadigmVortex family.

| Device Number | Unique CAM Library Features |
|---|---|
| ANx20 | Extensive standard library: comparator, differentiator, divider, filter, gain, voltage controlled gain, sample and hold, integrator, multiplier, oscillator, arbitrary waveform generator, rectifier, square root, sum/difference amp, arbitrary transfer function, reference voltage, zero crossing detector |
| ANx21 | Same as ANx20 plus Analog to Digital Converter (SAR) |
| ANx22 | Same as ANx20 plus Analog to Digital Converter (SAR) |
| ANx27 | Specialized audio signal processing functions |

# 3 Configuration Interface

The configuration interface provides a flexible solution for transferring data into the configuration memory of the FPAA. The interface supports both self booting via a serial EPROM and microprocessor hosted operation.

Master mode enables the FPAA to self-configure from a companion SPI EPROM. The FPAA provides all the necessary signaling for the serial transfer of data out of reset. The Master mode is useful for non-hosted, stand alone operation where the FPAA functionality is *set-and-forget*.

Slave mode sets the FPAA to present itself as a SPI bus compatible slave. The Slave mode is useful when the FPAA is to be configured from a host processor. As a slave, the FPAA can be configured anytime after reset. The AN22x devices offer the additional feature of allowing *reconfiguration*. This feature (not supported by the AN12x devices) 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 dynamic requirements of an application.

Systems of multiple FPAAs can be booted using either the Master or Slave modes via a bused connection of configuration interface signals, and logical device addressing (part of the configuration protocol). Slave mode FPAAs can be reconfigured concurrently or one at a time using fixed connections to the host. Configuration speeds of up to 40MHz are supported.

## 3.1 Slave Mode

The FPAA can be set to function as a SPI bus compatible slave. A fixed configuration data file may be sent in from the host processor to the FPAA as a one time / post-reset event. A more powerful use model is to have the host processor make on-the-fly adjustments to the analog circuitry: performing the calculation of new circuit parameters, assembling these new values into a configuration data block and transferring that data block into the FPAA.

### 3.1.1 Single FPAA - Connection Details

Figure 8 shows a simple slave connection driven by a companion host processor's SPI port (Master). In this example, the host processor manually controls the FPAA's reset input. (This is not required.) The FPAA's configuration data file is then simply transferred out of the host's SPI port.

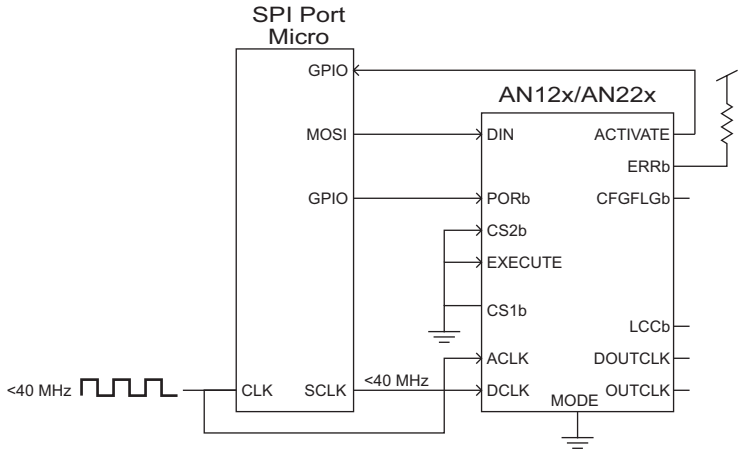



*Figure 8 – A Typical Host SPI Port Connection*

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.

The configuration interface is synchronous but there is no requirement for the configuration clock to be uninterrupted. For host processors lacking a SPI port, a *bit banging* software emulation of the SPI transfer protocol is perfectly suitable.

### 3.1.2 Multiple FPAAs - Connection Details

A system may contain an arbitrary number of slaved FPAA devices. As with any multi-slave SPI compatible bus, the slaves all connect to the Master's MOSI and SCLK signals in parallel. The host micro selects the first device in the chain for configuration. The upstream device holds off configuration for the next device downstream in the chain until its configuration is complete, as signaled by assertion of its LCCb pin. Using this scheme, FPAAs are sequentially loaded in the order shown. After the Primary Configuration is complete for each device in the chain, the devices can subsequently be *logically* addressed by LOGICAL ID (individually or by groups).

#### Logical Addressing of Multiple FPAAs - (AN22x Only)

ADDR1 (the primary logical address) is established for each device during its Primary Configuration. The configuration data itself establishes ADDR2 (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 ADDR1 value, a matching ADDR2 value or 0x FF in the TARGET ID byte field. The hex address of 0xFF serves as a global ID; all devices respond to this ID.

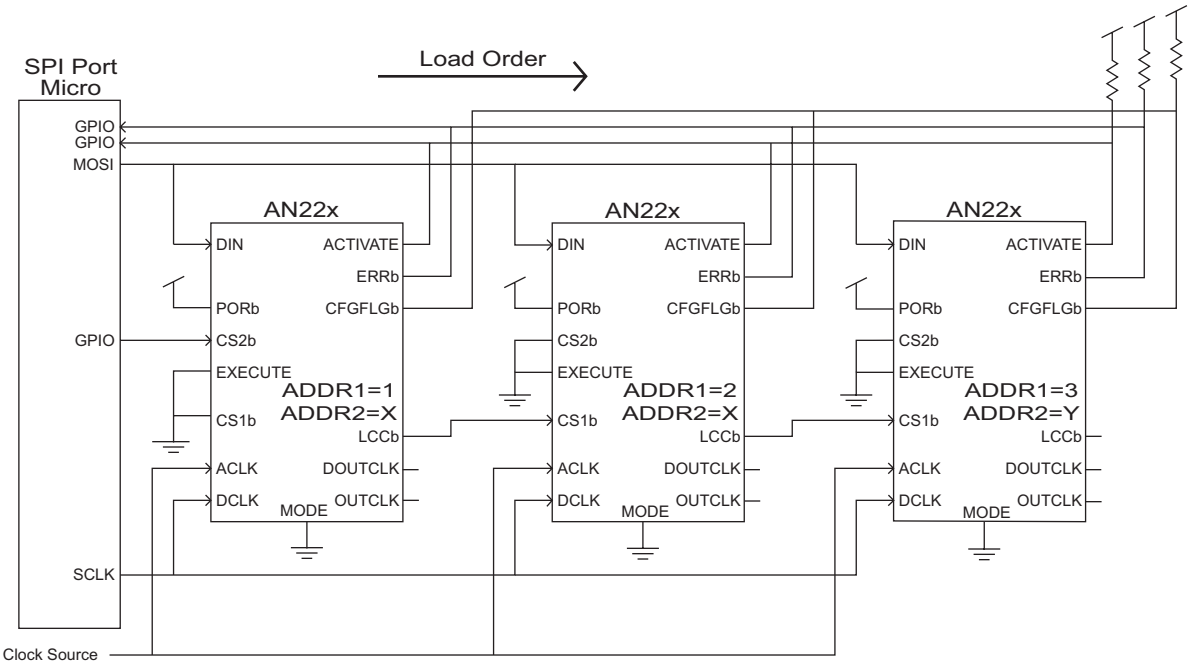



*Figure 9 – Configuring Multiple Devices from a Host Processor*

Figure 9 shows a valid connection and configuration example for multiple FPAA's being hosted from a single master SPI port. During Primary Configuration, each device in the chain received a unique ADDR1 and a non-unique ADDR2. The ADDR1's were assigned via the ADDR1 fields of the Primary Configuration data streams. The ADDR2'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 ADDR1 or ADDR2 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 ADDR2 values of X and Y respectively. During the Primary Configuration, the host processor assigned the first device an ADDR1 of 1 and filled that device with the X configuration. Likewise, the ADDR1=2 device also got the X configu-

ration. The ADDR1=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.

*Versions of AnadigmDesigner2 previous to 2.5 refer to ID1 and ID2 rather than ADDR1 and ADDR2.*

### 3.1.3 Configuration Clocking Considerations

The state machines within the device's configuration logic are driven by DCLK. 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.

### 3.1.4 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 14. 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.

### 3.1.5 EXECUTE

An advanced configuration feature allows the transfer of configuration data to the FPAA's Shadow SRAM to complete but the transfer of this data to its Configuration SRAM will be held off until signaled by the external EXECUTE pin. This feature allows finer temporal control and coordination between the programmable analog circuitry and the digital host processor.

*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, however Shadow SRAM for the LUT is not supported. Thus data written to the LUT becomes effective as it is written.*

## 3.2 Master Mode

The simplest method of configuring the FPAA is to wire it in Master mode and allow it to self boot from a previously programmed SPI EPROM. Anadigm FPAAs are directly compatible with industry standard 25 series SPI EPROMs.

| Manufacturer | Part Number |
|---|---|
| Atmel | AT25080 |
| Xicor | X5043 |
| Microchip | 25AA160 |

*Figure 10 – Known Compatible 25 Series SPI Memory Devices*

As system power comes up, the FPAA completes its internal power-on reset then provides the necessary signalling to read data out of the SPI EPROM. Once the transfer of configuration data is complete, the FPAA will activate its analog circuitry. The entire power-on reset and configuration process is automatic.

### 3.2.1 Single FPAA - Connection Details

A typical connection scheme for a standard SPI EPROM is shown in Figure 11. Once the FPAA'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. If there is any error encountered during this process, ERRb will assert low and the device will ignore all subsequent serial data.

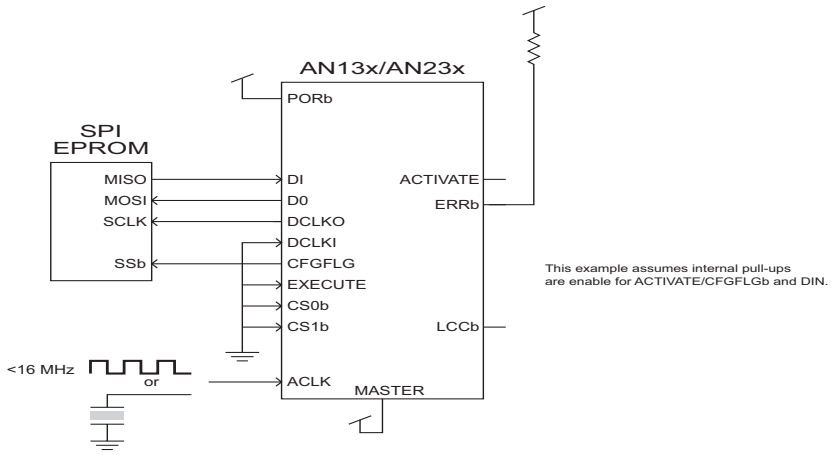



*Figure 11 – A Typical SPI EPROM Connection*

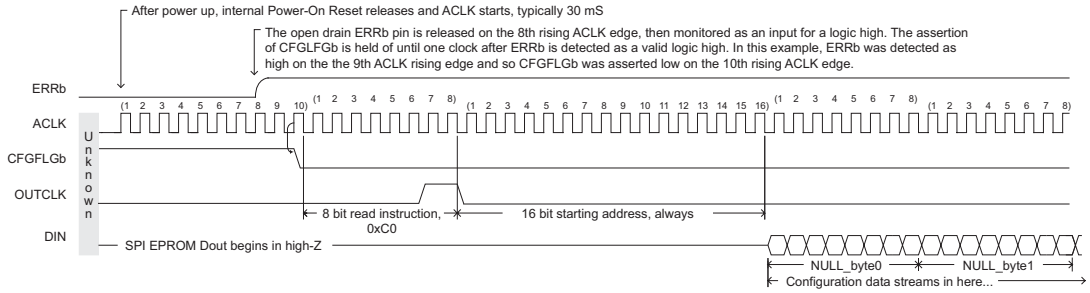



*Figure 12 – SPI EPROM, Initial Timing Sequence*

*ERRb is an open drain bi-directional pin driven active low during reset. On the 8th rising ACLK edge, the FPAA releases ERRb and then monitors the state of the pin as an input. CFGFLGb is asserted one ACLK after detection of a valid logic high on ERRb. In the timing example shown above, ERRb pulls up to a valid logic high in time to be detected as a logic high by the 9th rising ACLK edge; accordingly the FPAA asserts CFGFLGb low on the 10th rising ACLK edge. A high frequency ACLK, or any combination of high load capacitance or weak pull-up on the ERRb node may cause the node to pull high more slowly than shown in this example. In such situations, it is still the case that CFGFLGb will not assert low until one ACLK after a valid logic high is detected on ERRb.*

As the system's power supply first begins to ramp up, the ACLK, CFGFLGb, and OUTCLK signals are unknown. The device's internal Power-On Reset circuitry asserts and gets everything into 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.)

Serial 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.

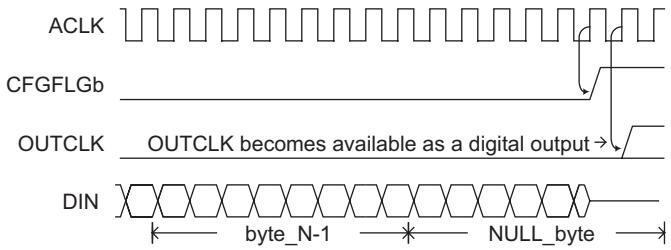



*Figure 13 – SPI EPROM, Completion Sequence*

As the last configuration data byte (a NULL byte) is being clocked into the device CFGFLGb de-asserts high. One ACLK later, OUTCLK becomes available as a digital output, typically reflecting one of four internal clocks.

### 3.2.2 Multiple FPAAs - Connection Details

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

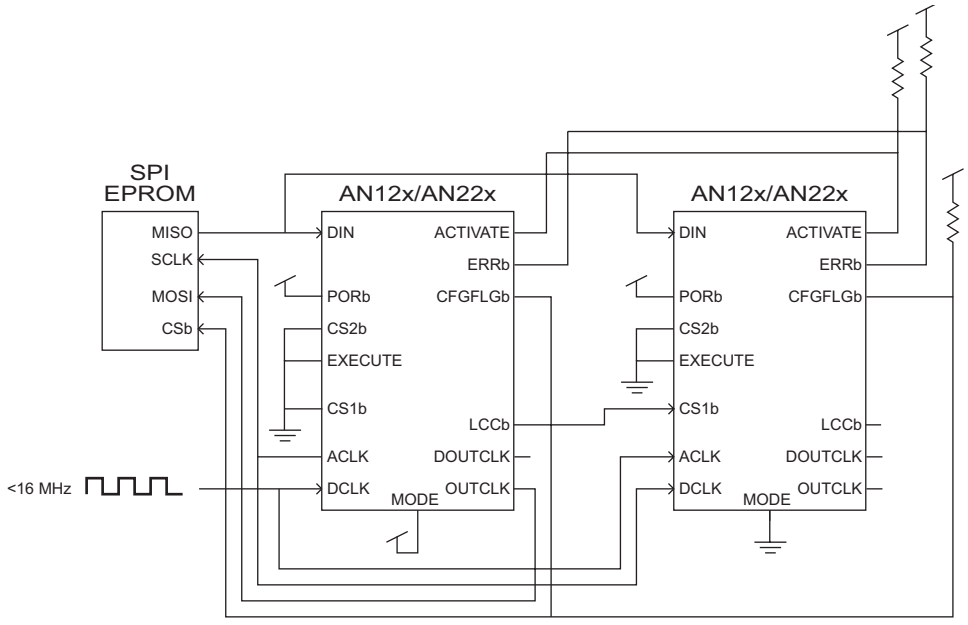



*Figure 14 – SPI Master Multi-Device Boot*

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 also have their open drain ACTIVATE bi-directional pins 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.

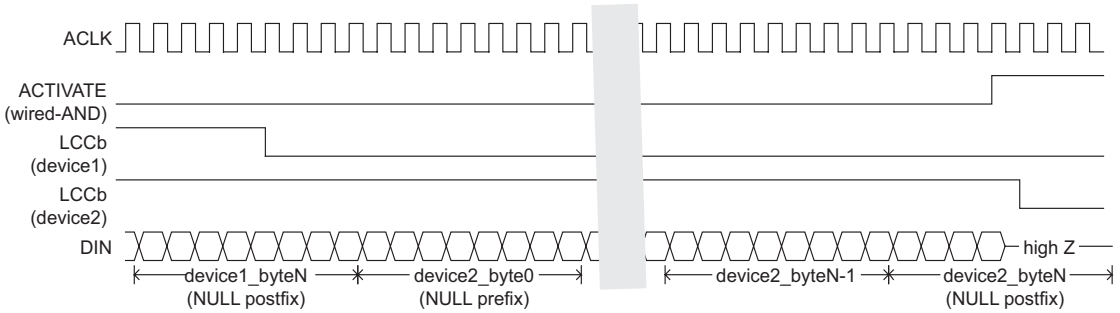



*Figure 15 – Multi-Device SPI EPROM Boot Handoff*

Figure 14 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 12. 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 NULL postfix byte, but it is part of a standard configuration data set. The clocks associated with a NULL post-fix byte are however required at the end of the configuration load for the second device. AnadigmDesigner2 provides these extra "for clocking only" prefix and postfix NULL bytes.

In Figure 14, 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 Figure 22 for further detail.)

### 3.2.3 Special Considerations for Use of Pull-Ups

There are several pins on the FPAA's configuration interface which have programmable internal pull-ups available. The ACTIVATE and CFGFLGb internal pull-ups are programmed as a pair. The DIN pull-up is separately programmable.

In small systems (one to three FPAAs), the internal pull-ups are sufficiently strong for proper operation under normal conditions. In such systems, only one of the FPAAs should have its internal pull-ups enabled. In larger systems, it may be necessary to use more robust external pull-up resistors. In such cases, the pull-up should be supplied with the external resistor alone and not in combination with any internal pull-ups.

A DIN pull-up (either internal or external) should always be used in EPROM booted (Master mode) systems. EPROMs typically hold their data output pin in tri-state and require a pull-up on the node to get it into a safe logic state.

## 3.3 Reset

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.

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 AN121E04, reconfiguration data is ignored.

## 3.4 Configuration Protocol

The serial configuration data, no matter how it comes into the device, must adhere to the protocol defined in this section. AnadigmDesigner2 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.

### 3.4.1 Primary Configuration Format & Byte Definitions

The Primary Configuration format is the format of the data that is generated by AnadigmDesigner2 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.

> *Primary Configuration data sets presume the device has been reset (Configuration SRAM zeroed out). In order to make configuration as efficient as possible, Primary Configuration data sets only contain data where one's need to be programmed.*

The Primary Configuration format is comprised of a Header Block followed by one or more Data Blocks. A header block contains a Sync byte, Device ID, ADDR1 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 a single data block terminator byte or two CRC 16 check bytes.

| | Data | Byte Name | Description |
|---|---|---|---|
| Header Block | 11010101 D5 | SYNC | Synchronization byte, always D5 |
| | 10110111 B7 | Device ID BYTE 0 | Bits [7:0] of Device ID - 0x800022B7 |
| | 00100010 22 | Device ID BYTE 1 | Bits [15:8] of Device ID |
| | 00000000 00 | Device ID BYTE 2 | Bits [23:16] of Device ID |
| | 10000000 80 | Device ID BYTE 3 | Bits [31:24] of Device ID |
| | XXXXXXXX | ADDR1 | ADDR1 Byte, Primary Logical Address for the FPAA |
| | 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 |
| | XXXXXXXX<br>or<br>00101010 2A | CRC_MSB<br>or<br>Data Block End | (depending on Bit 5 of BYTE ADDRESS)<br>Most significant byte of CRC16 error code<br>or<br>Data Block End Constant of 0x2A |
| | XXXXXXXX | CRC_LSB | Least significant byte of CRC16 error code (if used) |
| 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 |
| | XXXXXXXX<br>or<br>00101010 2A | CRC_MSB<br>or<br>Data Block End | (depending on Bit 5 of BYTE ADDRESS)<br>Most significant byte of CRC16 error code<br>or<br>Data Block End Constant of 0x2A |
| | XXXXXXXX | CRC_LSB | Least significant byte of CRC16 error code (if used) |

*Figure 16 – Primary Configuration Data Stream Structure*

### 3.4.2 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).

**Device ID BYTE n**

Every Anadigm device type has a unique 32 bit Device ID. Requiring the Device 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 Device 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.

| Family Member | 32 bit Device ID |
|---|---|
| AN120E04 | 0x300012B7 |
| AN220E04 | 0x300022B7 |
| AN121E04 | 0x800012B7 |
| AN221E04 | 0x800022B7 |
| AN221E02 | 0x000122B7 |
| AN122E04 | 0x000212B7 |
| AN222E04 | 0x000222B7 |
| AN127E04 | 0x000312B7 |
| AN227E04 | 0x000322B7 |

*Figure 17 – Device IDs for the AnadigmVortex Family*

**ADDR1 BYTE**

The ADDR1 field establishes one of the two logical addresses for the device. ADDR1 is considered the primary logical address for the device. The alternate logical address (ADDR2) 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 3.4.5 for further details.

### 3.4.3 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.) |
| | | | | | | | | (0 constant): **0:** Must be 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 3.5.14 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 AN121E04 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 3.5.7 for further explanation.) |
| | | | | | | | | (0 constant): **0:** Must be set to "0". |

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. |
| | | | | | | | | ENABLE CCITT-CRC16 CHECK | **1:** CCITT-CRC16 error checking is enabled<br>**0:** DATA BLOCK END checking is enabled |
| | | | | | | | | DATA_FOLLOWS | **1:** A subsequent Data Block will be expected by the configuration logic.<br>**0:** This block is presumed to be the final block of configuration data. |
| | | | | | | | | CONSTANT 1 | **1:** Must always be set to “1”.<br>**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 |
|---|---|
| 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 | |
| 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 | |

*Figure 18 – AN12x and AN22x 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.

**DATA BLOCK END BYTE**

If bit 5 of BYTE ADDRESS is 0, the only byte expected after the DATA bytes is the DATA BLOCK END constant 0x2A. 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 3.5.7.

**CRC_MSB BYTE**

If bit 5 of BYTE ADDRESS is 1, CRC_MSB followed by a CRC_LSB byte is expected at the conclusion of end of each data block. The 16 bit CRC is calculated using CCITT method with the polynomial:

x^16 + x^12 + x^5 + 1

*Calculation of a CRC is a compute intensive chore for a host processor and is not often used. A pre-calculated look-up table based approach can be considerably faster for a but requires significant storage for all of the possible results.*

### 3.4.4 Update Format (AN22x Only)

Once a Primary Configuration has been completed, there is no longer any requirement for the host to transmit out Device ID information or reprogram the ADDR1'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 3.4.1.

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 | ADDR1, ADDR2, 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 Data Block End | (depending on Bit 5 of BYTE ADDRESS) Most significant byte of CRC16 error code or Data Block End Constant of 0x2A |
| | 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 Data Block End | (depending on Bit 5 of BYTE ADDRESS) Most significant byte of CRC16 error code or Data Block End Constant of 0x2A |
| | XXXXXXXX | CRC_LSB | Least significant byte of CRC16 error code (if used) |

*Figure 19 – Update Data Stream Structure*

### 3.4.5 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" NULL prefix data. After the 40th clock,
00000000    //the configuration logic is ready.
00000000

11010101    //0xD5 is the required sync header.
10110111    //0xB7 is Least Significant Byte of Device ID word. (AN221E04)
00100010    //0x22 is byte 2 of Device ID word.
00000000    //0x00 is byte 3 of Device ID word.
10000000    //0x80 is Most Significant Byte of Device 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 Device ID, ADDR1
            //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" NULL data.
```

**Update Format Example (AN22x Only)**

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

00000001    //TARGET ID - The ADDR1 or ADDR2 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" NULL 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 SRAM to the Configuration SRAM
            //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 Configuration and Clock Pins - Detailed Information

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.<br>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<br>1, select clock support for SPI & FPGA EPROM interface |
| DCLK | Input | drive with < 40 MHz external configuration clock, or<br>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 .<br>After power-up, sources selected internal analog clock or comparator output. |
| PORb | Input | 0, Chip Held in reset state<br>Rising Edge, re-initiates power on reset sequence - 30 mS to complete |
| ERRb | Input | 0, Initiate Reset (hold low for 15 clocks)<br>1, No Action |
| | O.D. Output<br>10 kΩ p/u reqd. | 0, Error Condition<br>Z, No Error Condition |
| ACTIVATE | Input | 0, Hold off completion of configuration<br>Rising Edge, Complete configuration |
| | O.D. Output | 0, Device has not yet completed Primary Configuration<br>Z, Device has completed Primary Configuration |
| LCCb | Output | 0, Local configuration complete<br>1, Local configuration is not complete<br>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...<br>0, Ignore incoming data (unless currently addressed)<br>1, Pay attention to incoming data (watching for address) |
| | O.D. Output | 0, Device is being reconfigured<br>Z, Device is not being reconfigured |
| DIN | Input | Serial Configuration Data Input |
| CS1b | Input | (Prior to completion of a Primary Configuration)<br>0, Allow configuration to proceed<br>1, Hold off configuration |
| | Input | (After completion of a Primary Configuration)<br>Data input pin, serves as a serial data pass through port for a multi-device chain. |
| CS2b | Input | 0, Chip is selected<br>1, Chip is not selected |
| EXECUTE | Input | 0, No Action. This pin should normally be tied low.<br>1, Transfer Shadow SRAM into Configuration SRAM, depending on configuration settings |

*Figure 20 – Pins Associated with Device Configuration*

### 3.5.1 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.

### 3.5.2 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 40 MHz. 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.

| Manufacturer | 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 21 – Known Compatible Crystals*

### 3.5.3 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 SPI master clock for an attached SPI EPROM. It is a divide down (by16) version of DCLK.

### 3.5.4 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.

### 3.5.5 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 below.

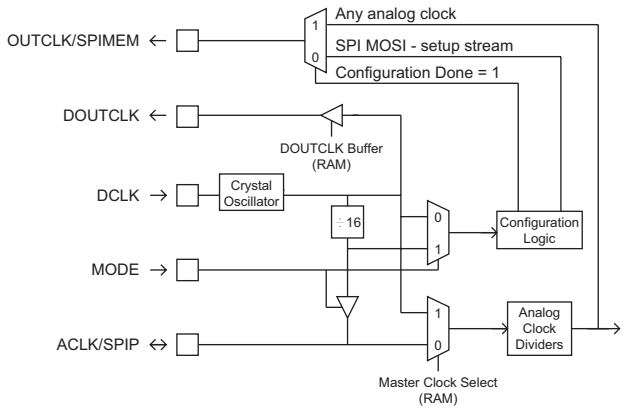



*Figure 22 – System Clocks as Directed by MODE Pin*

DCLK is the digital clock input. A crystal attached to this input will be detected automatically and used as the tuning element of the on-chip oscillator, otherwise it may be driven directly. The DCLK is signal is used to drive the configuration logic and may also be used as the master clock source for the analog clocks 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.

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. Once configuration completes, OUTCLK/SPIMEM reverts to serving as an analog clock or comparator output port.

DOUTCLK is a buffered version of DCLK.

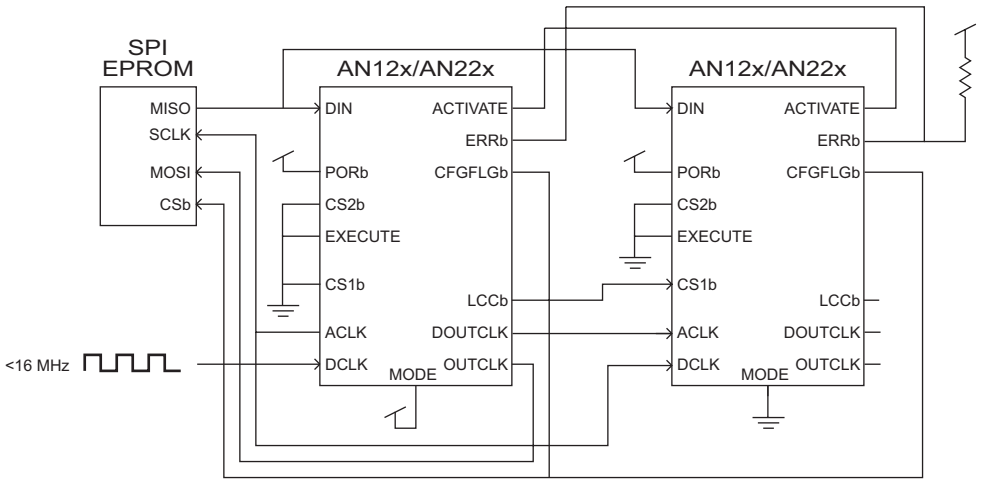



*Figure 23 – Using DOUTCLK in a Multi-Device System*

### 3.5.6 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.

### 3.5.7 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.

### 3.5.8 ACTIVATE

The ACTIVATE pin is an open drain input/output with an internal pull-up resistor selectable via configuration. It de-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 3.4.1 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 3.1.4 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.

### 3.5.9 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 3.2.2 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 9 for a detailed look at this configuration.

### 3.5.10 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.

### 3.5.11 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.

### 3.5.12 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 3.2.2 for further details on the LCCb to CS1b connection.

### 3.5.13 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.

### 3.5.14 EXECUTE

The EXECUTE input pin should normally be tied low.

# 4 Mechanical

## 4.1 Package Pin Assignments

| | | | |
|---|---|---|---|
| 1 | IO4PA | Analog IO+ | |
| 2 | IO4NA | Analog IO- | |
| 3 | O1P | Analog OUT+ | |
| 4 | O1N | Analog OUT- | |
| 5 | AVSS | Analog VSS | Analog Ground: 0 Volts |
| 6 | AVDD | Analog VDD | Analog Power: +5 Volts ±5% |
| 7 | O2P | Analog OUT+ | |
| 8 | O2N | Analog OUT- | |
| 9 | IO1P | Analog IO+ | |
| 10 | IO1N | Analog IO- | |
| 11 | IO2P | Analog IO+ | |
| 12 | IO2N | Analog IO- | |
| 13 | SHIELD | Analog VDD | Low noise VDD bias for capacitor array n-wells: +5 Volt |
| 14 | AVDD2 | Analog VDD | Analog Power: +5 Volts ±5% |
| 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: +5 Volts |
| 19 | BVSS | Analog VSS | Analog Ground for Bandgap Vref Generators: 0 Volts |
| 20 | CFGFLGb | Digital IN/OUT (open drain) (optional pull-up) | Configuration flag. A low output indicates configuration is in progress. |
| 21 | CS2b | Digital IN | Chip Select 2 |
| 22 | CS1b | Digital IN | Chip Select 1 |
| 23 | DCLK | Digital IN | Configuration data strobe and configuration state machine clock. |
| 24 | SVSS | Digital VSS | Digital Ground - Substrate Tie: 0 Volts |
| 25 | MODE | Digital IN | Sets configuration mode |
| 26 | ACLK/SPIP | Digital IN/OUT | Analog sample clock or EPROM clock |
| 27 | OUTCLK/SPIMEM | Digital OUT | Programmable Digital Output or EPROM MOSI data stream |
| 28 | DVDD | Digital VDD | +5 Volts ±5% |
| 29 | DVSS | Digital VSS | 0 Volts |
| 30 | DIN | Digital IN (optional pull-up) | Serial Configuration Data Input |
| 31 | LCCb | Digital OUT | Local configuration complete |
| 32 | ERRb | Digital IN/OUT (open drain) (10 KΩ p/u required) | Configuration error signal |
| 33 | ACTIVATE | Digital IN/OUT (open drain) (optional pull-up) | Enables Shadow SRAM to Configuration SRAM transfer |
| 34 | DOUTCLK | Digital OUT | Buffered version of DCLK |
| 35 | PORb | Digital IN/OUT (open drain) | Power On Reset - The minimum pulse width required is 25 nS. |
| 36 | EXECUTE | Digital IN | External trigger for Shadow SRAM to Configuration SRAM transfer |
| 37 | IO3P | Analog IO+ | |
| 38 | IO3N | Analog IO- | |
| 39 | IO4PD | Analog IO+ | Analog multiplexer input / output signals. The multiplexer can accept 4 differential pairs or 8 single ended connections. |
| 40 | IO4ND | Analog IO- | |
| 41 | IO4PC | Analog IO+ | |
| 42 | IO4NC | Analog IO- | |
| 43 | IO4PB | Analog IO+ | |
| 44 | IO4NB | Analog IO- | |

*Figure 24 – Package Pin Assignments*

## 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.

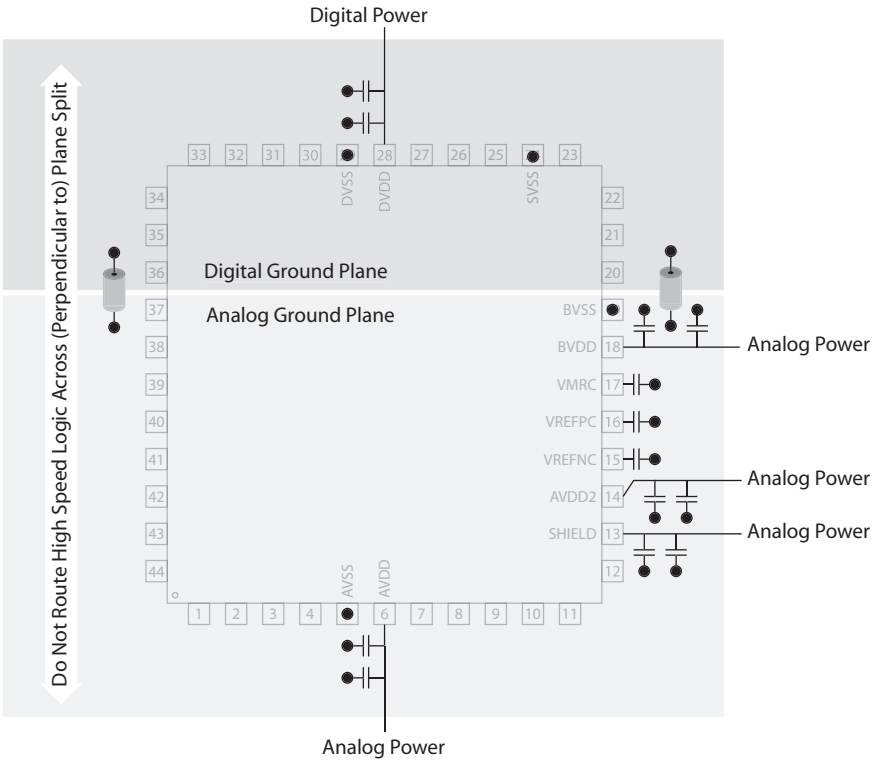



*Figure 25 – Basic Guidelines for Optimal PCB Design (Ground Planes)*

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 25, 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