AN10DS40

Preliminary Information

Development System for AN10E40 FPAA
Firmware Version 1.0

Anadigm was pleased to present the design community with the world's first true programmable analog array, the AN10E40. We coupled this array offering with a unique design system called AnadigmDesigner; supported with an extensive collection of analog circuit building blocks called IPmodules™. This triad of technology represents a powerful analog design environment for the designer with minimal analog knowledge. For the system designer, Anadigm is pleased to present the Anadigm programmable analog solution in a context that will allow you to leverage your strengths in processor/controller based system design.

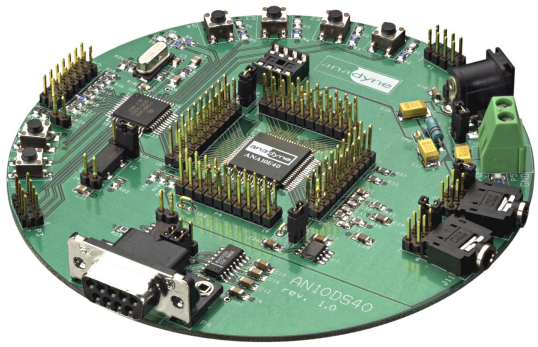
The AN10DS40 development system provides a complete and easy to use platform that allows you to easily investigate the benefits of field programmable, reconfigurable analog technology coupled with host processors/controllers. The system centers on the AN10E40 Field Programmable Analog Array (FPAA) and demonstrates how simple its connection to a companion microprocessor is. Stable, drift free analog circuits that can be reconfigured on the fly by companion logic in a killer combination of technologies that will "*extend the boundaries of digital systems*".

Out of the box, the AN10DS40 is ready to serve you in a number of ways. The companion microprocessor holds four FPAA configurations in its on-chip Flash memory. Simply power the board up, and hit one of the four configuration buttons to instantly configure the AN10E40 FPAA. Using an oscilloscope, you'll be able to quickly see the AN10E40 in action.

Connect the AN10DS40 FPAA development board via your PC's RS-232 port to AnadigmDesigner and you are ready to download your own designs directly to the FPAA. You may also download to the microprocessor's Flash memory for later stand alone operation, just described above. As your system design progresses to the prototyping phase, the AN10DS40 can also connect to your target system and serve as the download conduit.

Advanced users may elect to drive the AN10DS40 with an RS-232 host of their own design or drive it with a standard SPI master. The "next level" engineer will quickly realize the power of a tightly coupled host processor and start writing his own code. We encourage this exploration of the technologies, and have equipped you to do so. The schematic and source listing are both made available so you can use the AN10DS40 as the reference starting point that will quickly jumpstart your own designs.

The Major Features



- Investigate Microcontroller and FPAA Coprocessing
- Use as a Stand Alone FPAA Evaluation Platform
- Four FPAA Configurations Held in Flash
- Can be used with Standard Serial PROMs
- Connects to AnadigmDesigner via RS-232
- Connect to any SPI Master or RS-232 Host
- Can be Used to Configure Your Target System
- Speedy Downloads
- Full Source Code Included
- Schematics Included

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 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 convey any license under its patent rights nor the rights of others. Anadigm products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which the failure of the Anadigm product could create a situation where personal injury or death may occur. Should buyer purchase or use Anadigm product for any such unintended or unauthorized application, buyer shall indemnify and hold Anadigm and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that Anadigm was negligent regarding the design or manufacture of the part.

Table of Contents

Development System Overview

What does the development system do?

The development system demonstrates the easy pairing of an AN10E40 field programmable analog array (FPAA) with a microcontroller. The system is designed with resources that facilitate: stand alone operation, connection to AnadigmDesigner software, and connection to a target FPAA system.

Stand Alone Operation

Out of the box, the AN10DS40 is ready to demonstrate the AN10E40 FPAA. The on board microcontroller has 4 FPAA configurations stored in its on chip Flash memory. Simply power the board up (+8 to +12 VDC on the center post of P6), then hit one of the S1 through S4 pushbutton switches. Nearly instantly, the micro configures the FPAA and releases it for immediate operation.

The second stand alone function takes the microcontroller out of the picture, and lets the FPAA load up a configuration pattern under its own control out of reset. Placing a shorting block on J8 instructs the microcontroller to allow the FPAA to configure itself using the configuration contained in the serial EPROM socket (U4).

The stand alone operations are handy for quick demonstrations to your colleagues, and serves as an effective demonstration of the ease and speed of connection between Anadigm FPAA's and standard logic. However, when it comes to investigating the behavior of the AN10E40 device itself, you will soon want to connect the development system up to AnadigmDesigner.

Hosted by AnadigmDesigner

The AN10DS40 has a standard 9 pin RS-232 connector that allows quick connection to your PC. The FPAA design software AnadigmDesigner is design system aware. Downloading your experimental circuit configurations to the development system is simply a matter of making a single menu selection. On command, AnadigmDesigner will download the current configuration to the on board AN10E40, or to one of the four Flash memory configuration regions within the microcontroller (for latter stand alone demonstration).

As your design progresses, you may want to boot FPAA's in your own target system rather than on the AN10DS40 board; no problem. Shorting the J1 jumper, instructs the microcontroller to redirect configuration signals to the External Target Interface (P2).

Ready for your custom application

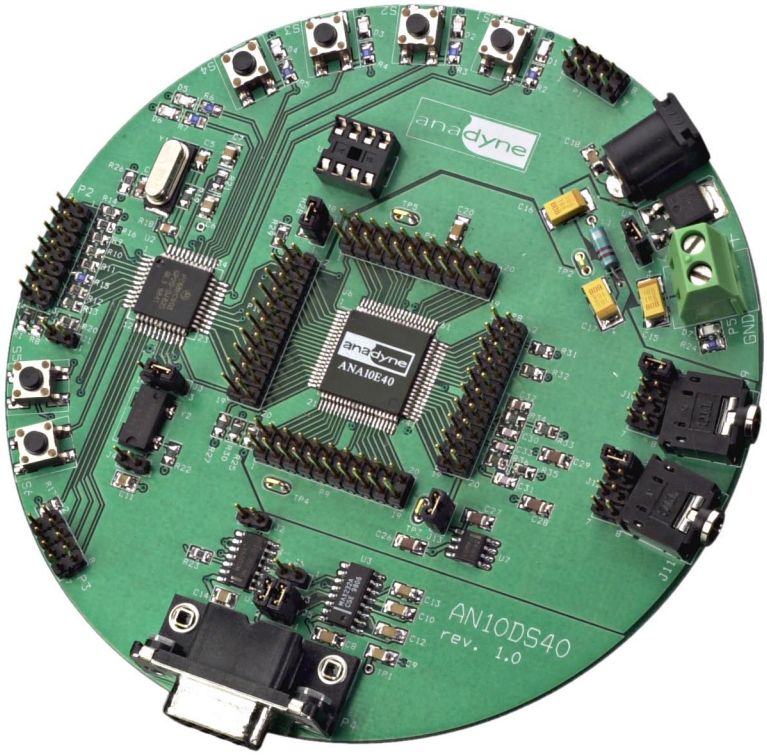
The AN10DS40 is ready for your own custom tasks. In addition to the functionality described above, the AN10DS40 is also ready to serve as a SPI Slave. A simple connection to any SPI Master system, and that system is ready to take control of downloads to the FPAA. The schematics, file formats and everything else you need to know to successfully communicate with the AN10DS40 through either its SPI or RS-232 ports is included in this document.

Now for the really good part…

Once you've gotten your feet wet, and begun to understand the power of programmable analog, we'd like to encourage you to transcend this first level and move on to the real power domain. This is where your logic works in concert with the FPAA. In this system environment, analog configurations are continuously swapped or adjusted under control of conventional logic. Leaving the static configuration paradigm behind, will allow you to build incredibly capable systems with unheard of efficiency. Anadigm is here to support you. Keep tabs on the website (http://www.anadyne-micro.com) for releases of the complete Anadigm Boot Kernel (ABK) library. We will provide you the connection points to software that allows you to concentrate on the power of the analog array, without having mess with the details of configuration communications.

A quick explanation of what is on board

Starting at 6 o'clock and working our way around clockwise, we first see a standard 9 pin RS-232 connector. Just behind the connector is jumper block J5, that allows the port to present itself either as a DCE or DTE interface. For the most common PC and cable combinations, the factory setting of DCE works fine.



P3 is a SPI port connection. Out of reset the design system watches for RS-232, SPI or push button activity. The design system is configures as a SPI slave. Any standard SPI master can talk directly to the AN10DS40 with no need for jumper setting or code changes.

IRQ and RESET buttons are provided. The reset button does the obvious thing, it resets the AN10DS40 back to its power up reset state. The IRQ button will cause an ABORT interrupt service routine to run. No matter what the AN10DS40 happens to be doing at the moment, pressing the IRQ button will take the ABK back to the IDLE state.

the 14 pin P2 connector brings a set of FPAA configuration signals off the board to facilitate connection to your own target system. Shorting jumper J1 causes the microcontroller to re-route configuration signals from the on board FPAA to this External Target Array interface.

The upper left quadrant of the board is occupied by 4 push button switches. Hitting any one of these 4 buttons tells the microcontroller to download the associated pre-stored configuration to the FPAA (either on board of to the target system). Adjacent to the switches is the P1 connector, which brings the pushbutton signals to a hand connector. A logic signal can be used to emulate a button press.

The P6 connector allows connection of any regulated 8-12 VDC power supply. Behind P6 is a 5 volt linear regulator. A pair of screw terminals is also provided if you should desire to bypass the on board voltage regulator (open jumper J7) and connect your own power.

J9 and J11 provide hand mini-stereo jacks for output and input respectively. Jumper blocks J10 and J12 allow for connection options.

Moving to the center of the board is the AN10E40 array itself with every pin connected up to a header for easy access. Please consult the AN10E40 data manual for a detailed description of the FPAA device.

Moving left again, is a Motorola MC68HC908 microcontroller. The logic portion of the board handles SPI, pushbutton and RS-232 communications tasks. It sources two sets of FPAA configuration signals. The first set is directed to the on chip AN10E40 array, the second set is directed to the External Target Array interface P2.

There are multiple options for driving a clock to the FPAA, the details of which are described below. The default jumper settings connect an on board 1MHz oscillator to the AN10E40's clock input.

There is one Green LED associated with each of the four configuration pushbutton switches. If the switch line goes low by either the button or its associated logic line on the P1 header, the LED will light.

A steady Yellow LED adjacent to the power connectors indicates that the board is powered up.

There are two other LED's that reflect that state of the FPAA's END and ERRb signals. If the array encounters an error during configuration, the download will be aborted, it will drop its ERRb signal, and the Red Error LED will light. This condition will remain until either a reset signal is asserted or until a new download is started.

The FPAA drops its END signal low during configuration, this will cause the Yellow Loading LED to light. This LED only stays light as long as the FPAA is configuring. Since this usually happens so quickly, it may appear as just a brief flicker.

Connecting Power

There are two options for establishing a power connection to the design system:
1. Connect +8 to +12 VDC to the center post of the 2.1mm jack P6 (GND to its sleeve) and leave J7 in place. The on board linear regulator will drop the voltage down to +5 VDC.
2. Remove the J7 jumper, and supply your own regulated +5VDC to adjacent screw terminals. A 3/4 amp in-line fuse is recommended.

Stand Alone Operation

Out of the box, the AN10DS40 is ready to begin work. With board power connected, just hit one of the four pushbutton switches, S1 through S4. The microcontroller will proceed to transfer a configuration image from the appropriate region of its on chip Flash to the on board FPAA. (Shorting jumper J1 will instead direct the configuration data stream to go to the P2 External Target Array interface.)

Configuration 1 - Emergency Broadcast System Alert Tone

The US Emergency Broadcast System was established in 1963 in order to provide a country wide civil communications network. Cooperating radio and television stations air a long alert tone before broadcasting the important information. The alert tone is specified as the sum of 853Hz with 960Hz sinusoidal waveforms. This tone resonates most small speaker systems well, is well within the bandwidth of the broadcasting channels and meets the important requirement that it be pretty darn annoying to listen to.

Recommended Equipment	Amplified Stereo Speakers Oscilloscope	
Input Channels	**None**	This demonstration circuit uses oscillator IPmodules to generate the two fundamental tones and a summing amplifier to combine them.
Output Channels	**P7-17** **P7-4** **P7-8&18** **J9**	853Hz Sinusoid 960Hz Sinusoid Sum of the above two signals. (Tip and Ring of stereo jack J9.) A convenient output jack to plug in a pair of amplified speakers.

Configuration 2 - Gain Stage of 5

The final configuration is a simple gain stage with the gain value set to 5. The Input and Output connections are the same as Configurations 2 and 3. Taken together, these 3 configurations then provide a set of circuits that can be very quickly switched in and out of operation, demonstrating on the fly reconfiguration potential of the AN10E40.

Recommended Equipment	Network Analyzer, or Oscilloscope and Audio Frequency Sweep Generator	
Input Channels	**J11 or P9-7&18**	Apply the (AC coupled) Network Analyzer or Sweep Generator's outputs up to the Tip and Ring of J11
Output Channels	**P7-8&18** **J9**	Sum of the two signals (Tip and Ring of stereo jack J9) A convenient output jack to plug in a pair of amplified speakers.

Configuration 3 - In Phase Music Cancellation

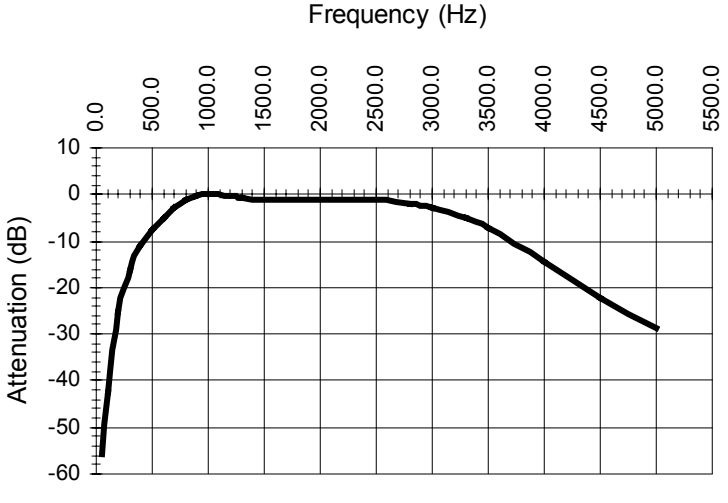
When listening to he world around you, most low frequency noise (or music, or someone else's music that you consider to be noise) reaches your ears roughly in phase. The low frequencies of sound have a longer wavelengths then higher frequencies. Consequently, for low frequency signals (signals in which the sound's wavelength is considerably larger than the distance between your two ears) will arrive at each of your ears roughly in phase. Higher frequency sounds on the other hand are much more likely to arrive at your two ears out of phase. It is this phase difference that allows us to "locate" a sound. This bit of physiology is taken advantage of in the design of surround sound music systems.

In this interesting demonstration, you supply the AN10E40 with a music signal (a portable CD player usually works very well for this). The circuit adds the left channel to the inverted version of the right channel, provides some gain and outputs it for your observation. The signal is a difference of the left and right channels, and is an demonstrates the content of "music" that allow you directionally locate a sound source.

Recommended Equipment	Amplified Stereo Speakers CD Player Mini-Stereo Plug to Mini-Stereo Plug cable	
Input Channels	**J11**	Using the mini-plug to mini-plug cable, connect the output of the CD player to the AN10DS40 input jack.
Output Channels	**P7-8&18** **J9**	Tip and Ring of J9 A convenient output jack to plug in a pair of amplified speakers. The difference signal is presented on both channels.

Configuration 4 - C Weighted Filtering

Without diving too deeply into the history and detailed use of C Weighted filtering, this filter specification (shown below) is commonly used for testing audio systems and environments in which human hearing is concerned.



C Weighted Filter

Passing an audio signal through the AN10E40 with this configuration image loaded will result in an output signal in which the tonal quality is something akin to AM radio or a telephone. While the signal may no longer be very pleasant to listen to, the point of the demonstration is that filters can be easily constructed in the AN10E40 to meet almost any arbitrary set of specifications.

Recommended Equipment	Network Analyzer, or Oscilloscope and Audio Frequency Sweep Generator	
Input Channels	**J11 or P9-7&18**	Apply the (AC coupled) Network Analyzer or Sweep Generator's outputs up to the Tip and Ring of J11
Output Channels	**P7-8&18** **J9**	Sum of the two signals (Tip and Ring of stereo jack J9) A convenient output jack to plug in a pair of amplified speakers.

The network analyzer will demonstrate the response of the circuit with a plot that looks like the above figure. If you are not fortunate enough to have a network analyzer, driving the input with an audio frequency sweep generator and watching the output on an oscilloscope (with the horizontal sweep speed the same as, and triggered by, the sweep generator) will give you a good feeling for the behavior of the circuit.

Using a Serial EPROM

Placing a shorting block on jumper J8 instructs the microcontroller to allow the on board AN10E40 to source its own configuration clock, and pull data directly out of the serial EPROM installed in socket U4. With all the focus of letting a micro handle the configuration tasks, you may need to be reminded that the AN10E40 is perfectly well suited to performing as a stand alone component. With a standard serial EPROM adjacent, the FPAA has its own configuration engine that allows it to automatically download a configuration image out of reset. The serial EPROM socket provided on the AN10DS40 board allows this to be demonstrated. Please note, that this particular boot method is only applicable to the on board FPAA. The J1 jumper setting is ignored.

Connecting Up To AnadigmDesigner

After you've messed around with the design system board a bit using its stand alone features, you'll soon want to connect up to AnadigmDesigner and start downloading your own configurations. The AN10DS40's RS-232 port is designed to make this connection as quick and trouble free as possible.

The physical connection

With the J5 block set to DCE configuration (1 shorted to 2, and 3 shorted to 4) most PC and serial cable combinations hook up just fine. AnadigmDesigner has options for setting up which of your PC's serial ports will be used. The firmware sets up the microcontroller to receive data at 9600 Baud, no start bits, 1 stop bit, 8 bits of data, no parity. If you have a non-standard serial cable, all that may be necessary to establish the connection is to configure the AN10DS40 to present itself as DTE (using J5, short 1 to 3, and 2 to 4).

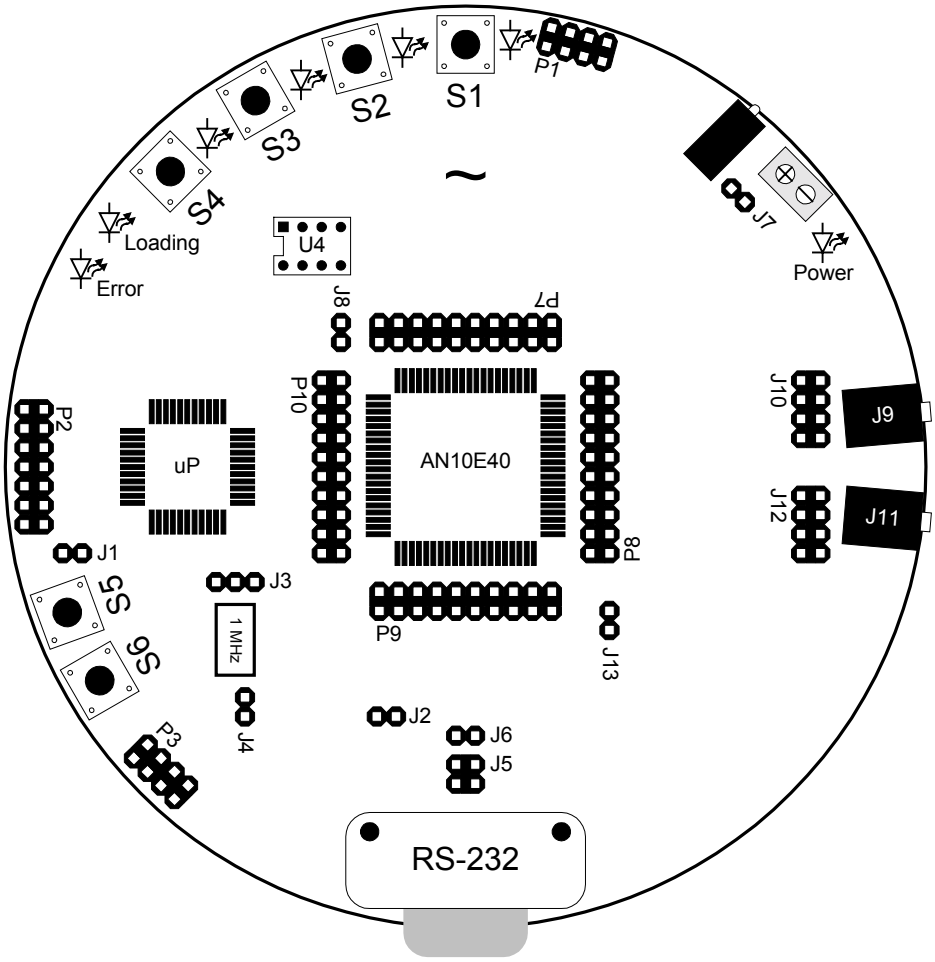
The download procedure

From the AnadigmDesigner pull down menu "Configure", you will see choices for downloading the currently loaded circuit either directly to an FPAA or to one of the 4 Flash memory regions within the microcontroller. Downloading to the microcontroller's Flash allows for later stand alone (disconnected from the PC) operation.
During the download, the Yellow LED will light, indicating that the download is in progress. In the unlikely even that any error is encountered by the FPAA which prevents the successful download, the Red LED will light, signifying an error condition. The ERROR LED will remain on until either a reset is asserted or a new download attempt made.

Note - As soon as you direct AnadigmDesigner to download to one of the configuration segments of the microcontroller's Flash memory, the old contents of that segment of memory will be lost. Please see the section ABK_Boot to understand how to restore your AN10DS40 to its as-first-shipped condition.

Development System Board Details

The AN10DS40 has everything you need to investigate the use of an AN10E40 FPAA as a peripheral to a microprocessor or as a stand alone device. The board has switches, connectors and sockets suitable for demonstration many of the devices possible uses.



Switches

Push Button Switches S1 through S4

As shipped from the factory, the AN10DS40 is ready to demonstrate the versatility of the AN10E40 FPAA. Simply power the board up and hit any one of the S1 through S4 switches. Nearly instantly, the microcontroller passes a segment of its Flash memory down to the AN10E40. After this very brief configuration sequence, the AN10E40 will be operating using the configuration associated with that section of Flash associated with the button pressed.

The P1 connector is wired in parallel to these 4 switches, allowing your own off board logic to emulate the closing of these 4 switches.

Push Button Switch S5 Reset

Pressing S5 resets the on board microcontroller, bringing the entire system back to its power on reset state. Access to this same RST signal is also provided on the P3 SPI connector.

Push Button Switch S6 IRQ

Pressing S6, causes the IRQb pin on the on board microcontroller to go low. As shipped from the factory, this runs and Abort routine. No matter what the processor happens to be doing at the moment, control is passed back to the ABK's IDLE mode. It is also available as a user feature for those advanced users wishing to use an interrupt for some custom functionality. Access to this same IRQ signal is also provided on the P3 SPI connector.

Connectors

For all two row "P" and "J" connectors, pin numbering increases up from the indicated Pin 1 in a zig-zag fashion, as shown in the tables presented below.

P1 - External Push Button Interface

Signal Name			Signal Name	
INPUT1	1	2	DGND	Applying a logic low to any of the INPUT[1:4] pins will have the same effect as manually pressing the normally open switches S1-S4. Logic driving these pins should be able to sink 8mA.
INPUT2	3	4	DGND	
INPUT3	5	6	DGND	
INPUT4	7	8	DGND	

P2 - External Target Array Interface

Signal Name			Signal Name	
ERRb	1	2	END	Connector P2 allows a direct connection to a target system other than the EVM (be sure to short jumper J1). Please consult the AN10E40 data manual for the complete functional description of these signals. Generally speaking, 1 for 1 connection of these signals to the target system's FPAA is all that is required to utilize P2 as a target configuration port.
CFG_CLK	3	4	DCLK	
D0	5	6	DGND	
MEMCEb	7	8	BFRb	
PWRUP	9	10	DGND	
RESETb	11	12	DVDD	
MODE1	13	14	MODE2	

P3 - External SPI Master Interface

Signal Name			Signal Name	
IRQb	1	2	SSb	Connector P3 accommodates connection of an external SPI master; the AN10DS40 board serving as a SPI slave. Details of the SPI signaling are given in the section below, Connecting to a SPI Master.
DGND	3	4	MISO	
DVDD	5	6	MOSI	
RSTb	7	8	SCK	

P7 - Top Side AN10E40 Signals

Signal Name			Signal Name	
ARRAYCLKOUT	1	2	RDX	Connectors P7, P8 and P9 provide easy access points to the AN10E40's analog IO X, Y & Z signals. AVDD supply is a filtered version of DVDD. AGND is the same electrical node as DGND, but there are only a few discrete connections between the two power planes to ensure optimal digital noise isolation to the analog circuitry. ARRAYCLKOUT reflects one of the AN10E40's internal analog clocks, as programmed by the user.
RDY	3	4	RDZ	
DGND	5	6	RDZ2	
RDY2	7	8	RCZ	
RCY	9	10	RCX	
AVDD	11	12	AGND	
DVDD	13	14	AGND	
RBX	15	16	RBY	
RBZ	17	18	RAZ	
RAY	19	20	RAX	

P8 - Right Side AN10E40 Signals

Signal Name			Signal Name
D1X	1	2	D1Y
D1Z	3	4	D2X
D2Y	5	6	D2Z
D3X	7	8	D3Y
D3Z	9	10	AGND
AVDD	11	12	D4X
D4Y	13	14	D4Z
D5X	15	16	D5Y
D5Z	17	18	CEXT
OPAMPVMR	19	20	VMR

Connectors P7, P8 and P9 provide easy access points to the AN10E40's analog IO X, Y & Z signals.

OPAMPVMR should neither be loaded or driven, it simply serves as a bypass point for the Voltage Mid-Rail signal for the array's amplifiers. Similarly CEXT is only a bypass point for the array's internal voltage reference generator. VMR signal is provided for monitoring. If J13 is removed, an externally supplied VMR signal can be applied on this pin or on J13-1.

P9 - Bottom Side AN10E40 Signals

Signal Name			Signal Name
LDY	1	2	LDX
LDZ2	3	4	LDZ
LCZ	5	6	LDY2
LCX	7	8	LCY
AVDD	9	10	AGND
LBX	11	12	AGND
LBZ	13	14	LBY
LAY	15	16	LAZ
VREFOUT	17	18	LAX
AVDD	19	20	AGND

Connectors P7, P8 and P9 provide easy access points to the AN10E40's analog IO X, Y & Z signals.

VREFOUT provides a convenient test point for monitoring the FPAA's internal reference voltage.

P10 - Left Side AN10E40 Signals (Configuration)

Signal Name			Signal Name
MODE1	1	2	CLOCK
CFG_CLK	3	4	MODE2
D0	5	6	D_CLK
D2	7	8	D1
D4	9	10	D3
D6	11	12	D5
ERRb	13	14	D7
BFRb	15	16	MEMCEb
END	17	18	PWRUP
RESETb	19	20	OPAMPDIS

The P10 connector gives direct access to the AN10E40's configuration interface.

For a complete description of the function and timing of these signals, please reference the AN10E40 data manual.

Jumpers

For all two row "P" and "J" connectors, pin numbering increases up from the indicated Pin 1 in a zig-zag fashion, as shown in the tables presented below.

J1 - Off Board Target Enable

Signal Name	
PTC6	1
DGND	2

Normally Open - Leaving J1 open tells the ABK that the target FPAA is on board.

Shorting J1 informs the ABK that the target FPAA is off board. If shorted, all configuration signals are routed to the P2 target system interface connector

J2 - Factory Reserved, Flash Restoration

Signal Name	
Factory Reserved	1
Factory Reserved	2

Normally Open - Leave J2 open during normal operation.

Shorting J2 should only be done during Flash restoration, please see the section, Rebuilding the AN10DS40 back to Factory Contents.

J3 - On Board Clock Resource

Signal Name	
(uP)CLKOUT	1
(FPAA) CLOCK	2
DGND	3

Normally 1& 2 are Shorted - This allows either the on board 1 MHz oscillator (J4 Open) or the microcontroller (J4 Shorted) to drive the AN10E40's CLOCK signal.

Removing the shorting block allows you to drive 2 using an external signal generator (using 3 as a return ground).

J4 - Microcontroller CLKOUT Enable

Signal Name	
CLKSEL	1
DVDD	2

Normally Open - Leaving J4 open instructs the microcontroller to disable its CLKOUT signal, allowing the 1 MHz on board oscillator to provide the clock to the AN10E40.

Shorting J4 disables the 1 MHz on board oscillator output and instructs the microcontroller to drive the FPAA CLOCK signal using its CLKOUT signal.

J5 - DCE vs. DTE Configuration Block

Signal Name			Signal Name
TXD	1	2	TOUT
RIN	3	4	RXD

Normal (DCE) Configuration:
1 shorted to 2
3 shorted to 4
In the normal configuration, this presents the AN10DS40's RS-232 port as DCE. This is the configuration compatible with most PC's and commons serial port cables.

DTE Configuration:
1 shorted to 3
2 shorted to 4
In the DTE configuration, the RS-232 Tx and Rx lines are crossed before going to the AN10DS40's transceiver chip.

J6 - Factory Reserved, Flash Restoration

Signal Name	
Factory Reserved	1
Factory Reserved	2

Normally Open - Leave J6 open during normal operation.

Shorting J6 should only be done during Flash restoration, please see the section, Rebuilding the AN10DS40 back to Factory Contents

J7 - On Board Voltage Regulator Enable

Signal Name		
Volt Regulator Out	1	**Normally Shorted** - Shorting J7 connects the output of the on board voltage regulator to the board's DVDD power plane. In order for things to work correctly, you'll of course need to supply between 8 and 12 Volts DC to P6 (center post is +). If J7 is open, the user is expected to supply a clean will regulated 5V DC to the screw terminals on P5.
DVDD	2	

J8 - Serial EPROM Enable

Signal Name		
MODE2	1	**Normally Open** - Leave J8 open during normal operation. Shorting J8 allows the on board AN10E40 to boot from the serial EPROM socket U4. The ABK detects this signal being low, and gets out of the way.
DGND	2	

J10 - Stereo Jack Output & Configuration Block

Signal Name			Signal Name	
RAZ	1	2	LEFT OUT (ring)	**Normal Configuration:** **1 shorted to 2** **3 shorted to 4** **6 shorted to 8** In the normal configuration, AGND is used as the ground reference to the signals being presented to the mini stereo plug J9. 5 shorted to 6 In this alternate configuration, VMR is instead passed as the ground reference to J9. This is not a typically recommended configuration. The VMR output is not designed to source or sink any load current.
RCZ	3	4	RIGHT OUT (tip)	
VMR	5	6	GND (sleeve)	
	7	8	AGND	

J12 - Stereo Jack Input & Configuration Block

Signal Name			Signal Name	
LAX	1	2	LEFT IN (ring)	**Normal Configuration:** **1 shorted to 2** **3 shorted to 4** **5 shorted to 6** In the normal configuration, VMR (typically 2.5V with respect to AGND) is used as the ground reference to the signals being presented to the mini stereo plug J11. 6 shorted to 8 In this alternate configuration, AGND is instead passed as the ground reference to J11.
LCX	3	4	RIGHT IN (tip)	
VMR	5	6	GND (sleeve)	
	7	8	AGND	

J13 - Enable External VMR Source

Signal Name		
(FPAA) VMR	1	**Normally Open** - Leave J13 open during normal operation. Shorting J13 allows you to use the AN10DS40's external 2.5V reference voltage to drive the AN10E40's VMR rail. Be sure to disable the on-chip VMR in the particular FPAA configuration being used when this jumper is in place.
External 2.5V Ref.	2	

AN10E40 Package to Board Signal Cross Reference

The following table provides a quick cross reference between the AN10E40 package pins and the most commonly used board connections for access to those pins. For a more complete description of the connections, please see the board schematic in Appendix C.

The signal naming convention holds that active low signals are named with a “b” suffix.

Board	Pkg.	Signal Name	Type	Description
P7-1	1	ARRAYCLKOUT	Dig. Out	Programming allows one of the 4 internal clocks to be presented here.
P10-1	2	MODE[1]	Dig. In	Configuration mode control pin 0 = Boot From Serial ROM (BFR) 1 = Micro Peripheral Interface Mode
P10-4 J8-2 to DGND	3	MODE[2]	Dig. In	Configuration mode control pin 0 = Use Internal Clock (CFG_CLK is output, 1/8 internal ring oscillator freq.) 1 = Use External Clock (CFG_CLK is clock input to the configuration logic.)
P10-3	4	CFG_CLK	Dig. I/O	Configuration logic clock Direction controlled by MODE[2]
P10-6	5	DCLK	Dig. Out	SPROM Configuration clock output 1/2 frequency of CFG_CLK.
P10-5	6	DATA[0]	Dig. I/O	Data pins used for loading configuration data. DATA[0] is used for serial BFR mode, and the entire byte width is used in Micro mode.
P10-8	7	DATA[1]	Dig. I/O	
P10-7	8	DATA[2]	Dig. I/O	
P10-10	9	DATA[3]	Dig. I/O	
P10-9	10	DATA[4]	Dig. I/O	
P10-12	11	DATA[5]	Dig. I/O	
P10-11	12	DATA[6]	Dig. I/O	
P10-14	13	DATA[7]	Dig. I/O	
P10-13	14	F1 (ERRb, RDb)	Dig. I/O	Configuration Function pins (BFR Mode , Micro Mode)
P10-16	15	F2 (MEMCEb, WRb)	Dig. I/O	
P10-15	16	F0 (BFRb, CSb)	Dig. In	
P10-18	17	F3 (PWRUP, RS)	Dig. In	
P10-17	18	F4 (END, BUSYb)	Dig. Out	
P10-20	19	OPAMP_DISABLE	Dig. In	Op-Amp disable input (normally tied to Vss, not usually utilized in systems) Takes precedence over BFR’s PWRUP input and Micro’s Function Register Bit Position 4 (Analog Enable) 0 = Analog circuitry enabled 1 = Analog circuitry disabled
P10-19	20	RESETb	Dig. In	Chip RESET Falling edge detected to start Reset
P9-2	21	IOLDX	Ana. In	Unbuffered Analog input
P9-1	22	IOLDY	Ana. In	Buffered Analog input
P9-4	23	IOLDZ	Ana. Out	Buffered Analog output
P9-3	24	IOLDZ2	Ana. Out	Uncommitted op-amp output
P9-6	25	IOLDY2	Ana. In	Uncommitted op-amp input
P9-5	26	IOLCZ	Ana. Out	Buffered op-amp output
P9-8	27	IOLCY	Ana. In	Buffered Analog input
P9-7	28	IOLCX	Ana. In	Unbuffered Analog input
AVDD	29	AVDD	Pwr/Gnd	Analog VDD, 5 Volts
AGND	30	AVSS	Pwr/Gnd	Analog VSS, 0 Volts
AGND	31	SVSS	Pwr/Gnd	Substrate VSS, 0 Volts
P9-11	32	IOLBX	Ana. In	Unbuffered Analog input
P9-14	33	IOLBY	Ana. In	Buffered Analog input

P9-13	34	IOLBZ	Ana. Out	Buffered analog output
P9-16	35	IOLAZ	Ana. Out	Buffered op-amp output
P9-15	36	IOLAY	Ana. In	Buffered Analog input
P9-18	37	IOLAX	Ana. In	Unbuffered Analog input
P9-17	38	VREFOUT	Ana. Out	Reference voltage
AVDD	39	BVDD	Pwr/Gnd	Bandgap VDD, 5 Volts
AGND	40	BVSS	Pwr/Gnd	Bandgap VSS, 0 Volts
P8-20	41	VMR	Ana. Out	Signal ground, 2.5 Volts (normally left floating)
P8-19	42	OPAMP_VMR	Ana. In	Signal ground, 2.5 Volts (usually loaded with 10nF to AVSS)
P8-18	43	CEXT		External VMR capacitor (usually loaded with 10nF to AVSS)
P8-17	44	IOD5Z	Ana. Out	Buffered op-amp output
P8-16	45	IOD5Y	Ana. In	Buffered Analog input
P8-15	46	IOD5X	Ana. In	Unbuffered Analog input
P8-14	47	IOD4Z	Ana. Out	Buffered op-amp output
P8-13	48	IOD4Y	Ana. In	Buffered Analog input
P8-12	49	IOD4X	Ana. In	Unbuffered Analog input
DVDD	50	ESD_VDD	Pwr/Gnd	ESD Structures VDD, 5 Volts
DGND	51	ESD_VSS	Pwr/Gnd	ESD Structures VSS, 0 Volts
P8-9	52	IOD3Z	Ana. Out	Buffered op-amp output
P8-8	53	IOD3Y	Ana. In	Buffered Analog input
P8-7	54	IOD3X	Ana. In	Unbuffered Analog input
P8-6	55	IOD2Z	Ana. Out	Buffered Analog output
P8-5	56	IOD2Y	Ana. In	Buffered Analog input
P8-4	57	IOD2X	Ana. In	Unbuffered Analog input
P8-3	58	IOD1Z	Ana. Out	Buffered Analog output
P8-2	59	IOD1Y	Ana. In	Buffered Analog input
P8-1	60	IOD1X	Ana. In	Unbuffered Analog input
P7-20	61	IORAX	Ana. In	Unbuffered Analog input
P7-19	62	IORAY	Ana. In	Buffered Analog input
P7-18	63	IORAZ	Ana. Out	Buffered Analog output
P7-17	64	IORBZ	Ana. Out	Buffered Analog output
P7-16	65	IORBY	Ana. In	Buffered Analog input
P7-15	66	IORBX	Ana. In	Unbuffered Analog input
DVDD	67	CFG_VDD	Pwr/Gnd	Configuration (Digital) VDD ,5 Volts
AGND	68	SVSS	Pwr/Gnd	Substrate VSS, 0 Volts
AVDD	69	SVDD	Pwr/Gnd	Substrate VDD, 5 Volts
P10-2	70	CLOCK	Dig. In	System master clock. Used by clock generator which feeds all switch capacitor analog circuitry.
P7-10	71	IORCX	Ana. In	Unbuffered Analog input
P7-9	72	IORCY	Ana. In	Buffered Analog Input
P7-8	73	IORCZ	Ana. Out	Buffered Analog output
P7-7	74	IORDY2	Ana. In	Uncommitted op-amp input
P7-6	75	IORDZ2	Ana. Out	Uncommitted op-amp output
DGND	76	CFG_VSS	Pwr/Gnd	Configuration (Digital) VSS, 0 Volts
P7-4	77	IORDZ	Ana. Out	Buffered Analog output
P7-3	78	IORDY	Ana. In	Buffered Analog input
P7-2	79	IORDX	Ana. In	Unbuffered Analog input
DGND	80	POR	Dig. In	Power on Reset (normally tied to VSS)

Downloading to your Target System

Once your design progresses to the point where you are ready to start booting your own target system containing an AN10E40, the AN10DS40 is still ready to serve you. When driving the development board via its RS-232 port, shorting the J1 jumper instructs the microcontroller to route all configuration traffic to the P2 External Target Array interface, rather than to the on board FPAA. The design systems otherwise behaves as if it were talking to the on board FPAA. In this fashion, the AN10DS40 now serves as an intelligent download cable to your target system, translating ASCII Hex configuration data from RS-232 to binary compatible with the AN10E40's Boot From ROM port.

Please refer to the design system schematic in Appendix C for connection details.

Advanced User Topics

The Microcontroller's Memory Map

Before covering any unfamiliar landscape its good to have a map handy. Please refer back to these maps as the discussion of the firmware progresses. All addresses and address contents are shown in hex. The first table shows a high level view of the entire memory map, showing only major sections. Subsequent tables show more detail for each of these major sections.

High Level View - The Complete Memory Map

Mem. Type	Section Name	Addrs.	Description
RAM	IO and RAM	0000 023F	Configuration and Data registers for IO Ports and RAM area for program execution
	Unimplemented	0240 7FFF	Unimplemented portion of the memory map.
Flash	FPAA Configuration Files	8000 8DFF	Four blocks of Flash. Each contains a unique non-volatile configuration image for the AN10E40. Great for stand alone demonstrations of the AN10DS40 system.
	User Playground	8E00 CFFF	The user is free to experiment in this area of Flash.
	ABK	D000 FAFF	This area contains the Anadigm Boot Kernal (ABK). This is the program that allows AnadigmDesigner to talk to the AN10DS40 board.
Protected Flash	Protected ABK	FB00 FDFF	This portion of Flash is protected from erasure. It contains the bootloader portion of the ABK.
RAM, ROM and FLASH	Status, Control, Monitor ROM and ISR Vectors	FE00 FFFF	This final section contains status and control registers for many of the micro's features. Also in this range is a small monitor program in ROM as well as vectors for each of the interrupt sources routines (ISRs).

IO and RAM

Area Name	Addrs.	Contents	Description
IO Registers	0000 003F		IO Data and Control registers, including: MPIO ports, SPI, SCI etc.
Bootloader Variables	0040 0041 0042 0043 0044 0045 0046	reserved reserved reserved First Addr. Hi First Addr. Lo Last Addr. Hi Last Addr. Lo	 First address of Flash to be erased or programmed Last address +1 of Flash to be erased or programmed
User Application and ABK	0100 01DF		User Application and ABK RAM space.
Bootloader Stack and RAM Execution Space	01E0 023F		The stack is initialized at 0200 and grows up to 01E0. The remainder of this space is used for RAM execution routines.

FPAA Configuration Files

Area Name	Addrs.	Contents	Description
Configuration File 1	8000 837F		This is the FPAA configuration image that gets loaded into the AN10E40 when pushbutton S1 is pressed.
Configuration File 2	8380 86FF		…when pushbutton S2 is pressed.
Configuration File 3	8700 8A7F		…when pushbutton S3 is pressed.
Configuration File 4	8A80 8DFF		…when pushbutton S4 is pressed.

User Playground

Area Name	Addrs.	Contents	Description
User Playground	8E00 CFFF		This area is left available to the advanced user for the installation of custom software. A typical use might be a program which uses the microcontroller's A-D converter in conjunction with the FPAA to yield even more powerful applications.

ABK

Area Name	Addrs.	Contents	Description
General ABK Code	D000 FABF		The unprotected portion of the ABK resides in this area. This portion of the ABK handles things like the SPI interface, configuration Flash programming and AN10E40 configuration.
Configuration Settings	FAC0 FAC1 FAC2 FAC3 FAC4 FAC5 FAC6 FAC7 FAC8 FAC9	reserved reserved reserved Init_Baud Init_Config1 Init_Config2 First Addr. Hi First Addr. Lo Last Addr. Hi Last Addr. Lo	MC68HC908GP32 specific parameters Initial BAUD register value (default = 0x30) Initial CONFIG1 register value (default = 0x01) Initial CONFIG2 register value (default = 0x03) Initial First Address to be erased or programmed (default = 0x8000) Initial Last Address +1 to be erased or programmed (default = 0xFB00)
Jump Table	FACA FACD FAD0 FAD3 FAD6 FAD9 FADC FADF FAE2 FAE5 FAE8 FAEB FAEE FAF1 FAF4 FAF7 FAFA FAFD	JMP Hi/Lo JMP Hi/Lo JMP Hi/Lo JMP Hi/Lo JMP Hi/Lo JMP Hi/Lo JMP Hi/Lo JMP Hi/Lo JMP Hi/Lo JMP Hi/Lo JMP Hi/Lo JMP Hi/Lo JMP Hi/Lo JMP Hi/Lo JMP Hi/Lo JMP Hi/Lo JMP Hi/Lo JMP Hi/Lo	"JMP" and Hi&Lo address bytes for the Timebase ISR …for ADC …for Keyboard …for SCI Transmit …for SCI Receive …for SCI Error …for SPI Transmit …for SPI Receive …for Timer2 Overflow …for Timer2 Channel 1 …for Timer2 Channel 0 …for Timer1 Overflow …for Timer1 Channel 1 …for Timer1 Channel 0 …for PLL …for IRQ …for SWI …for RESET of the ABK

Protected ABK

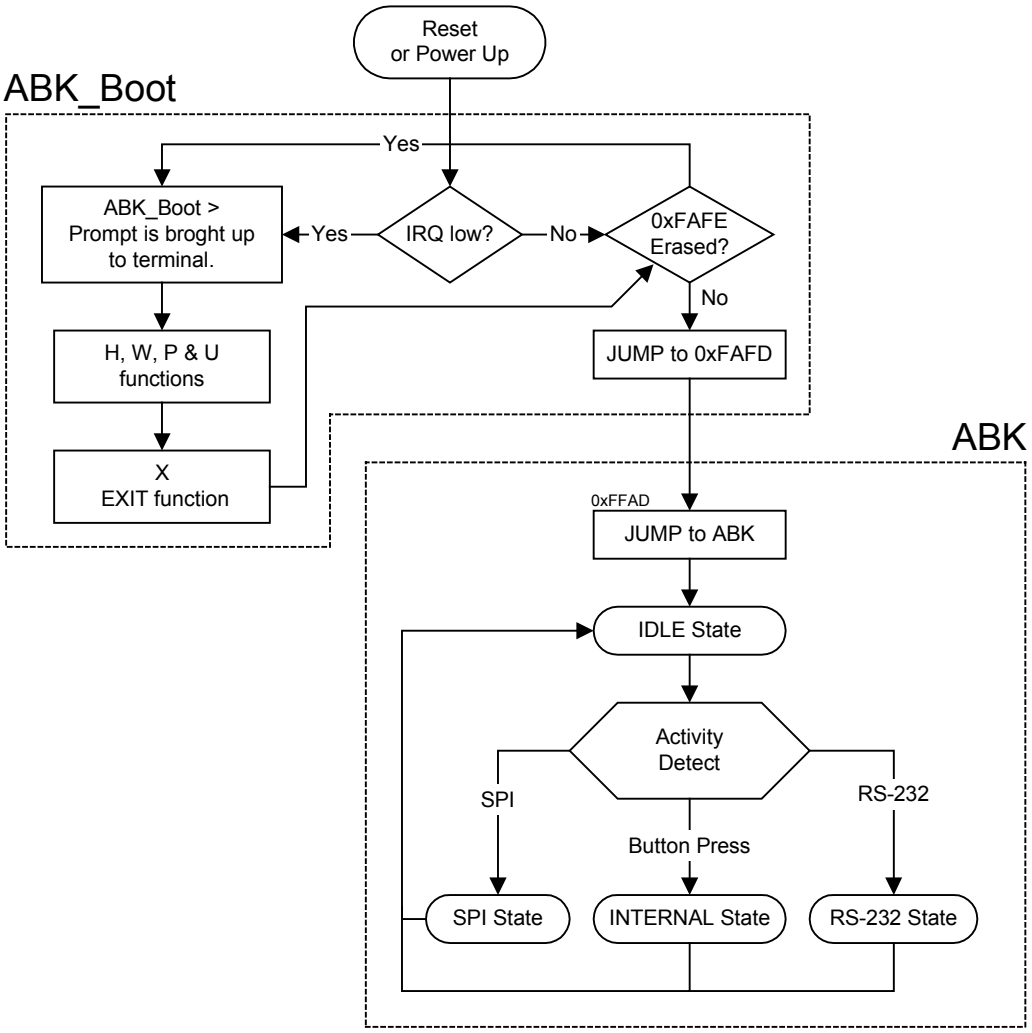
Area Name	Addrs.	Contents	Description
Protected ABK	FB00 FDFF		This area of Flash is protected from erasure. It contains the bootloader portion of the ABK. This portion of the ABK is the first to run out of a reset condition. It checks the state of the IRQ line, and if low drops into a terminal hosting mode which allows the erasure and reprogramming of the remainder of Flash.

Status, Control, Monitor ROM and ISR Vectors

Mem. Type and Area Name	Addrs.	Contents	Description
RAM - Status and Control	FE00 FE0C		Status and Control registers for things like the SIM, Flash, Interrupt Controller and Breakpoint unit.
ROM - Monitor ROM	FE20 FE52		Monitor ROM
RAM - Flash Block Protect	FE7F		Flash Block Protect Register - This is the register that keeps the FB00-FDFF range protected.
FLASH - Interrupt Vectors	FFDC FFFF	FACA FACD FAD0 FAD3 FAD6 FAD9 FADC FADF FAE2 FAE5 FAE8 FAEB FAEE FAF1 FAF4 FAF7 FAFA FAFD	Hi&Lo address bytes for Timebase ISR JMP to point …for ADC …for Keyboard …for SCI Transmit …for SCI Receive …for SCI Error …for SPI Transmit …for SPI Receive …for Timer2 Overflow …for Timer2 Channel 1 …for Timer2 Channel 0 …for Timer1 Overflow …for Timer1 Channel 1 …for Timer1 Channel 0 …for PLL …for IRQ …for SWI …for RESET of the ABK

Firmware Overview

As provided from Anadigm, the firmware program on board the AN10DS40's microcontroller is actually a pair of programs: ABK_Boot, and Anadigm Boot Kernal (ABK). Usually, the design environment of AnadigmDesigner handles all the details of talking to the AN10DS40 board, but we recognize that many users may eventually move to an environment where the AN10DS40 is hosted by a system of their own design. In such instances, the firmware is designed to make your connection to the FPAA via the AN10DS40 as trouble free as possible.



Anadigm Boot Kernal (ABK)

The ABK running in the microcontroller as shipped from our factory has 4 basic states: IDLE, SPI, RS-232 and INTERNAL. Out of reset control is usually quickly from ABK_Boot to ABK and it moves into the IDLE state. From the IDLE state, the ABK watches for activity from its general purpose IO (connected to 4 pushbuttons), SPI and RS-232 ports. Once activity has been detected, the ABK moves to the next appropriate state. Once in any state besides IDLE, all other state functionality is excluded until completion of a successful download, or until an error is detected and displayed.

There is a RESET pushbutton which when pressed will bring the board back to the IDLE state after a reset. There is also an IRQ button that is used to get the ABK_Boot program into the state which allows firmware upgrade (more on that later).

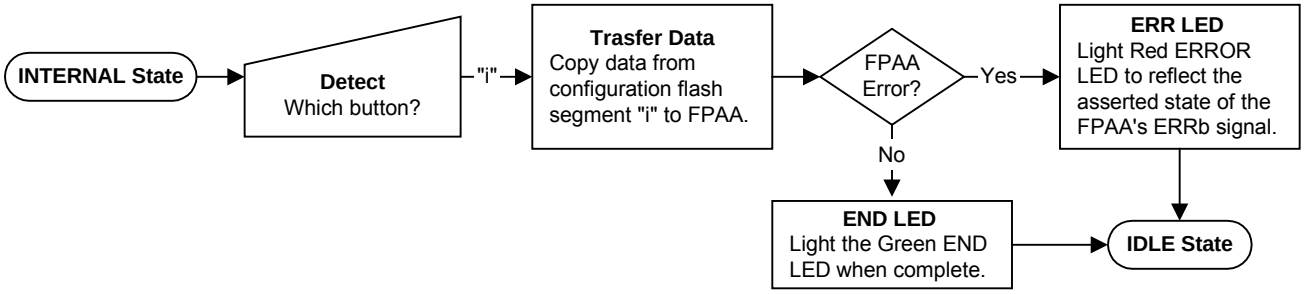
IDLE State

Once out of reset, the ABK will sit forever in the IDLE state. During Idle, the ABK watches for activity on PortA [0A-3A], the SPI port (configured as a slave) and the RS-232 port. Once activity from any of these sources is detected, control is passed to the appropriate state processing section of the ABK. Once that section's processing is complete, control is passed back to the IDLE state.

If any problems are encountered in the IDLE state, the ERROR LED will light solid forever. A cold boot, reset or watch dog time out are the only ways out of such an unexpected condition. If functioning correctly, the watch dog time out period is just 250mS, so "forever" will not last longer than a quarter of a second.

INTERNAL State

Out of reset, the microcontroller configures the first 4 lines of the PortA general purpose IO to detect logic low levels (usually sourced from the four configuration buttons on the AN10DS40 board). When the IDLE state of the ABK detects a button press, control is passed to INTERNAL state processing. In this state, the appropriate segment of the microcontroller's internal flash memory is downloaded to the target FPAA.



If there are any problems encountered during INTERNAL state processing, the download will be aborted and the Red ERROR LED will light, reflecting the state of the FPAA's ERRb signal. If everything went according to plan, then instead the Green END LED will light signaling the completion of a successful configuration sequence.

At the end of configuration (or on the detection of ERRb), the INTERNAL state terminates and control is passed back to the IDLE state of the ABK.

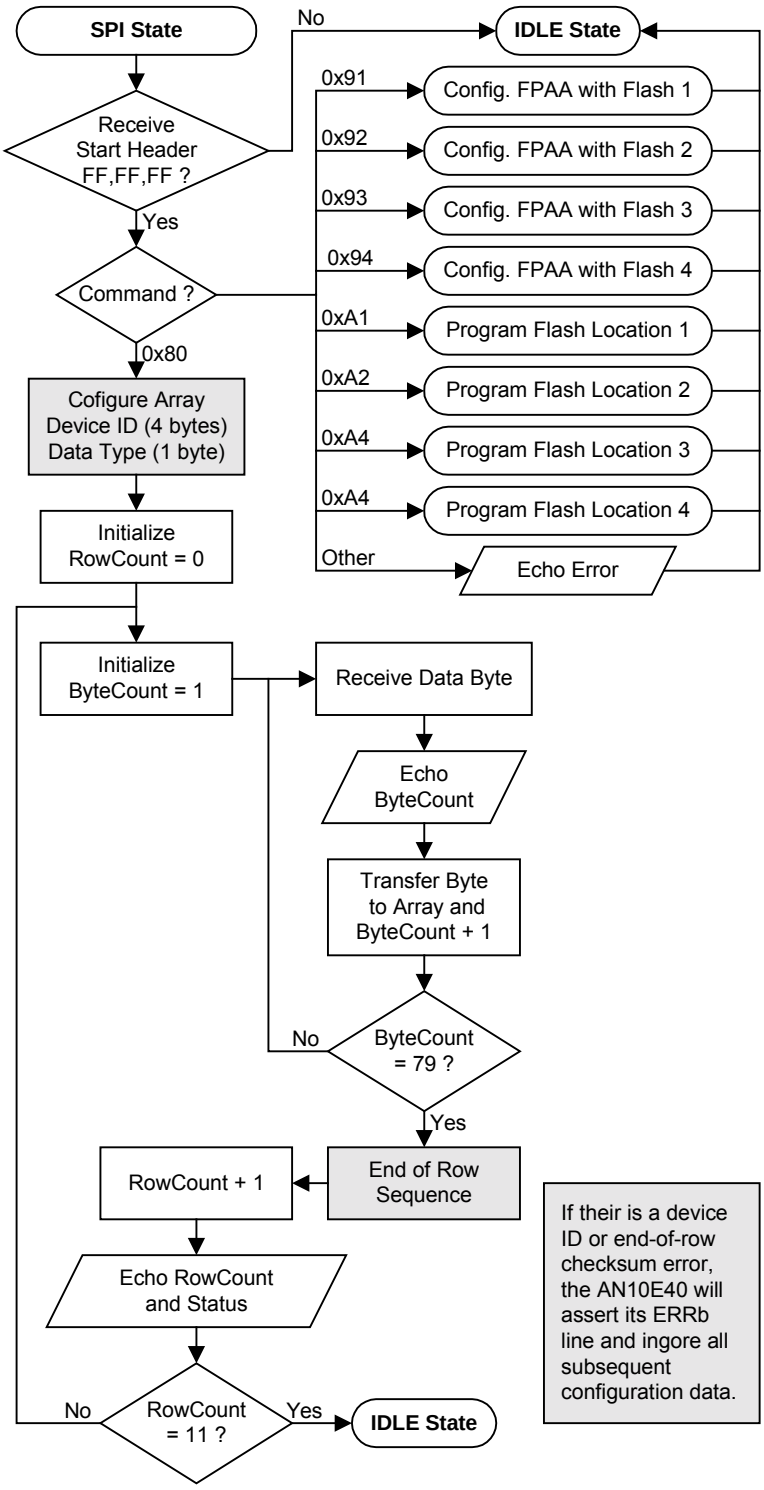
SPI State

Out of reset, the microcontroller configures its SPI port as a SPI slave. When the IDLE state of the ABK detects incoming SPI traffic, control is passed to SPI state processing. In this state, an incoming SPI data stream is accepted. (Simultaneous with the receipt of incoming data bytes the slave SPI port returns status bytes.) The data stream may be directed to the target FPAA or to the microcontroller's internal Flash memory depending on the data stream's command word.

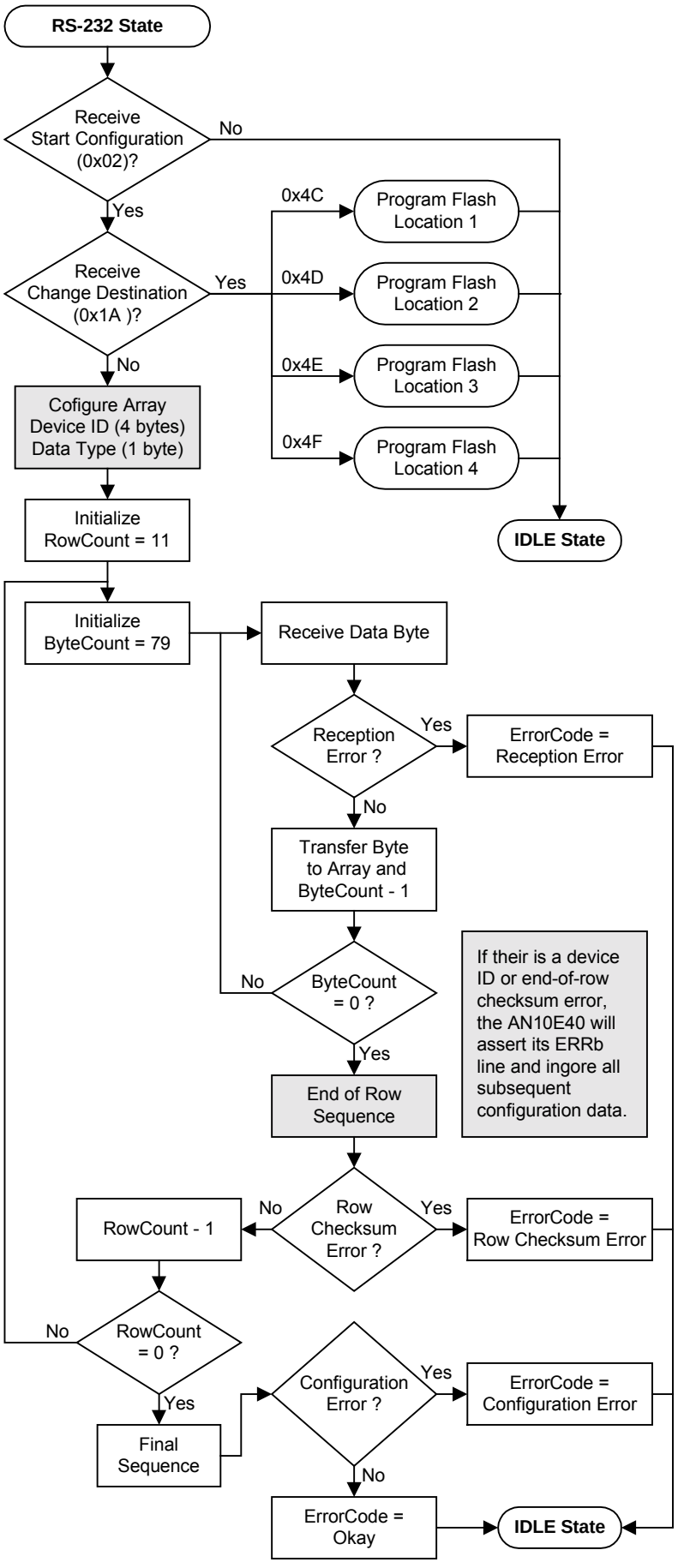
The SPI data stream is expected to be the binary representation of the configuration data (unlike an RS-232 stream in which the data comes across in ASCII Hex format). The binary representation of the data is more efficient and effectively halves the transfer time over moving ASCII data.

If there are any problems encountered during SPI state processing, ERRb will assert and the Red ERROR LED will light. All subsequent configuration data will be ignored by the FPAA.

The SPI state concludes and control is passed back to the IDLE state after the complete configuration bit stream has been transmitted.



RS-232 State



Out of reset, the microcontroller configures its SCI port as a full duplex RS-232 DCE port. When the IDLE state of the ABK detects incoming RS-232 traffic, control is passed to the RS-232 state for processing. In this state, an incoming RS-232 data stream is accepted. As with the SPI port, the data stream may be directed to the target FPAA or to the microcontroller's internal flash memory depending on the data stream header (if any).

The RS-232 data stream is expected basically a straight copy of the ASCII Hex version of the configuration file (.AHF), prefixed by an STX character and optionally a comand word header. The SPI data stream on the other hand is the binary representation of the configuration file.

If there are any problems encountered during RS-232 state processing, the download will be aborted and the Red ERROR LED will light, reflecting the state of the FPAA's ERRb signal.

At the end of configuration (or on the detection of ERRb), the RS-232 state terminates and control is passed back to the IDLE state of the ABK.

ABK_Boot

The ABK_Boot program resides in a factory protected region of Flash memory. It is this program that allows for user installed upgrades of ABK. It also allows the user to restore Flash contents back to their factory settings.

Post power-on/reset processing will always traverse at least a small portion of ABK_Boot. During this brief time, ABK_Boot will examine the state of the processor's IRQ line. During normal use, the line will be high and ABK_Boot will next check to see that at least "something" is programmed into Flash location 0xFAFE. Normally, the high address byte of the first instruction of the ABK is there, and ABK_Boot Jumps to 0xFAFD and from there the ABK takes over.

If however, either the IRQ line was low (the IRQ button (S6) is held down) during this post power-on/reset processing, or, ABK_Boot detects that location 0xFAFE is erased (to 0xFF) then ABK_Boot retains control.

From this point forward, you must have a terminal program connected up to the AN10DS40 RS-232 using the following settings: 9600 Baud, No Start Bit,8 Data Bits, 1 Stop Bit, No Parity (9600-N,8,1,N).

Note- Teraterm, Hyperterm, Procomm etc are all workable choices.

You should see a prompt the "ABK_Boot>" prompt. From here you have but a few choice to make. The only valid commands are H or ?,W,P,U, and X. Lower case is accepted.

H - Help

The H (or ?) command brings up a brief reminder of all the valid ABK_Boot interactive commands.

W - Wipe

The W command erases all unprotected Flash (0x8000 through 0xFAFF).

Note Well - This region of Flash normally contains the ABK. **The W command will wipe out the ABK.** Do not execute this command unless you are prepared to load in your own custom software, or re-load an ABK image.

P - Program

The P command will prompt you to download a file using your terminal program's file transfer utility. The only file format accepted is a 16 bit Motorola S-Record. The only valid record types are S0, S1 and S9. No more than 16 data bytes per S-Record are accepted. The P command does not check for valid address ranges within the S-Record, nor does it check to ensure that Flash is actually erased before programming begins.

U - Update

The U command is simply a wipe followed by a program. This update command is usual method for updating ABK firmware, or for loading in your own full custom programs to Flash.

X - Exit

The X command will cause ABK_Boot to check the contents of 0xFAFE. If the data at this location is not 0xFF, then ABK_Boot assumes that Flash is filled with a valid program, and a JUMP to 0xFAFD will be performed.

If the data at location 0xFAFE is equal to 0xFF, ABK_Boot will assume that the Flash is erased and will not jump away.

Configuration Bit Streams - AHF Format

After a design is complete, AnadigmDesigner is used to create a configuration data file. These files contain the data that describes the CAB configuration and routing information for your design. One of the most easily understood formats of configuration files is the ASCII Hex Format (.AHF). The following sections of this manual refer to configuration files, so now would be a good time to briefly review what is in them.

In an .AHF file, a byte is represented by a pair of ASCII characters which represent the Hex value of that byte. For example 11110111 is presented as F7. A prototype of an .AHF configuration file for an AN10E40 is given below. In this example, most of the configuration data has been truncated (and replaced with a comment) to facilitate concise explanation. Device programmers and the RS-232 state firmware of the AN10DS40 ignore white space characters in .AHF files (CR and LF being typical examples).

```
138502B7 (4 Byte Device ID)
00 (Data Type Byte)
00AE . . . (78 configuration data bytes, plus row  0 checksum byte) . . . 0000 AF
0000 . . . (78 configuration data bytes, plus row  1 checksum byte) . . . 0000 B6
0300 . . . (78 configuration data bytes, plus row  2 checksum byte) . . . 0000 43
0021 . . . (78 configuration data bytes, plus row  3 checksum byte) . . . 0000 C3
0000 . . . (78 configuration data bytes, plus row  4 checksum byte) . . . 0000 6C
0000 . . . (78 configuration data bytes, plus row  5 checksum byte) . . . 0000 44
0E40 . . . (78 configuration data bytes, plus row  6 checksum byte) . . . 0000 D2
0022 . . . (78 configuration data bytes, plus row  7 checksum byte) . . . 0000 34
3700 . . . (78 configuration data bytes, plus row  8 checksum byte) . . . 0000 DE
00CD . . . (78 configuration data bytes, plus row  9 checksum byte) . . . 0000 52
3031 . . . (78 configuration data bytes, plus row 10 checksum byte) . . . 0000 14
```

All .AHF files begin with a 4 byte device ID. All AN10E40 devices contain the `138502B7` ID code. If there is a mismatch between the device ID downloaded and the actual device ID, then the FPAA asserts an error signal and aborts the download.

Following the ID, there is one additional data type byte which tells the FPAA configuration circuit something about all the subsequent data. Currently, the only supported data type is `00`, which designates unencrypted, uncompressed, sequential data.

After the Device ID and Data Type bytes have been successfully digested by the FPAA's configuration circuitry, all subsequent data gets routed through a checksum calculator and on into a 78 byte long configuration data shift register. During download, the shift register fills while a check sum is internally calculated. This calculated checksum is compared to the 79th byte downloaded for the current row. If there is a match, the shift register contents are shifted down into the next row of the FPAA's configuration SRAM, and the shift register is ready to accept another row's worth of data. If there is ever a checksum mismatch, the FPAA asserts its error signal and aborts the download. The AN10E40 has 11 rows of configuration SRAM that need to be loaded in this manner.

Connecting to an RS-232 Host

The P4 DB-9 connector allows you to connect the AN10DS40 board up to your PC's serial port. This is the typical connection and AnadigmDesigner supports it directly. Of course, you may instead connect an RS-232 host of your own design. Once powered up, the on board microcontroller continuously monitors this port for activity. The following sections describe the physical and logical connections.

The physical connection

A short Glossary of RS-232 keywords is appropriate to make the following reading a bit easier:

DTR	Data Terminal Ready signal line
DSR	Data Set Ready signal line
DCD	(a.k.a. CD) Carrier Detect signal line
RTS	Request To Send signal line
CTS	Clear To Send signal line
TxD	Transmit Data signal line
RxD	Receive Data signal line
DTE	Data Terminal Equipment (a terminal)
DCE	Data Communications Equipment (a modem)

Full schematic details were described in the section above, Development System Board Details. But to recap the essential elements: the AND10DS40's RS-232 connector hardwires DTR to DSR and DCD. It also hardwires RTS to CTS. TxD and RxD pins go to a pair of shorting blocks that allows the AN10DS40 to present itself as either a DTE or DCE device. For DCE configuration (compatible with connection to most PC's using the most common serial cables) place the jumpers of J5 such that pin 1 is shorted to 2, and pin 3 is shorted to 4. If you happen to be using a null-modem or crossover cable, then instead place the jumpers on J5 in the 1-3 and 2-4 positions.

Leave the J2 and J6 jumpers off, these are for factory testing.

At 9600 Baud, there is no possibility of overrun, so no hand shaking is required. Thus the AN10DS40 boards simply shorts RTS and CTS lines together.

Special Command Headers for RS-232 Routed Configuration Bit Streams

Usually, configuration data will be passed directly on to the AN10E40 with no special command headers required. Simply send a start byte (STX character, 0x02) and then the .AHF configuration file to the AN10DS40's RS-232 port. If instead, you want to program that portion of the board's microcontroller Flash memory which stores FPAA configurations (for later stand alone operation), then simply prefix the .AHF file with one of the four command headers described below. (If you are using AnadigmDesigner to handle the download to the AN10DS40 you don't need to concern yourself with modifying any .AHF data stream, the insertion of these special header bytes is handled automatically during download time.)

Every command header begins with a hex 0x1A ("^Z" a.k.a. "SUB") byte. The following are the recognized RS-232 command headers for configuring on the microcontroller's flash:

RS-232 Command	Description	
`0x02 0x1A 0x4C`	`"STX SUB L"`	Program Flash file 1 using the following configuration data stream.
`0x02 0x1A 0x4D`	`"STX SUB M"`	Program Flash file 2 using the following configuration data stream.
`0x02 0x1A 0x4E`	`"STX SUB N"`	Program Flash file 3 using the following configuration data stream.
`0x02 0x1A 0x4F`	`"STX SUB O"`	Program Flash file 4 using the following configuration data stream.

The ABK actually ignores the upper nibble of these command bytes, so actually; 0xC,D,E,F are also valid. All other command headers are factory reserved.

The logical connection

Configure the host RS-232 port with the following settings:

1 start bit
8 data bits
0 stop bits
No Parity
RTS/CTS handshaking
9600 Baud

Once these setting are established do the following:
1. Transmit out an 0x02 byte ("^B" a.k.a. "STX"); this launches the RS-232 state of the ABK.
2. (Optional) Transmit one of the special flash two byte command headers described above.
3. Transmit the .AHF file without translation.
4. (Optional) Transmit an 0x0A byte ("^J" a.k.a. "LF"); this prompts the ABK to return a status byte
5. (Optional) Receive the status byte. 0xC0 means that everything is OK, anything else is a fail.

The ABK expects the entire configuration file to be transferred. If there is a break in the data stream longer than 10 seconds, the firmware assumes there is some problem and takes the following actions:
1. sends out a fail byte code (Anything but 0xC0 is a failure code.)
2. abandons the download
3. clears the Flash segment to a safe state (if Flash was the target of the download)
4. asserts the Red Error LED
5. reverts back to IDLE State

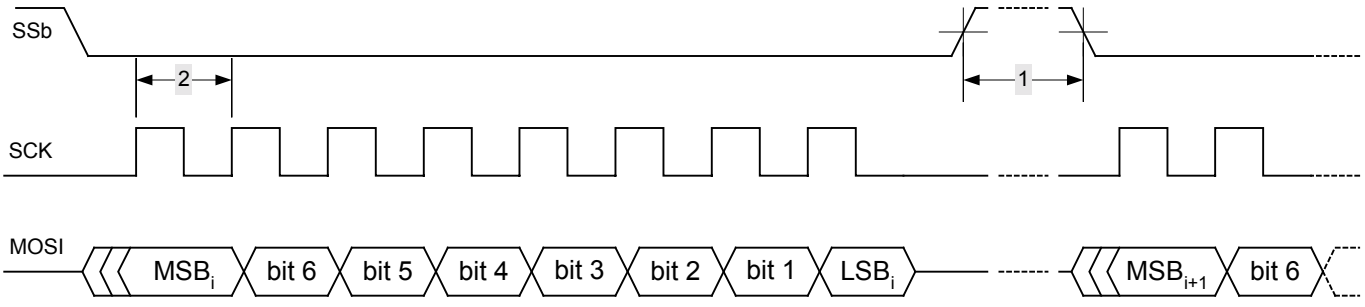
Connecting to a SPI Master

The P3 connector allows you to connect the AN10DS40 board to a SPI master of your own design. The on board microcontroller powers up with its SPI port in slave mode and continuously monitors this port for activity. The following sections describe the physical and logical connections.

The physical connection

The SPI header connector (P4) expects 5V CMOS level signals. The microcontroller's SPI port is configured as a slave. The only required connections are P3[2] - SSb, P3[6] - MOSI, P3[8] - SCK and P3[3] - DGND. Connection to P3[4] - MISO is not required (but that is where the download status byte is returned). There is a 10K pull-up on SSb.

The expected SPI signaling format is shown below.



For most processors, the above waveform represents the most common SPI configuration (CPOL=0, CPHA=1). With these SPI settings, the data is shifted out by both Master and Slave on the rising edge of SCK, and clocked in on the falling edge. The SPI Master is expected to source SSb and SCK.

Timing Notes:

1 Time between byte transfers is 100 μs minimum.

2 The minimum SPI clock period to ensure no possibility of data overrun is 5 μs (200 KHz).

Special Command Headers for SPI Routed Configuration Bit Streams

Every data stream directed to the AN10DS40 SPI port must have a 3 byte header followed by a one byte SPI Command word. The three byte header must always be `0xFF, 0xFF, 0xFF`. The binary representations of the valid one byte SPI Command words are given as follows:

SPI Command	Description
`00000000` through	(available for user software)
`011111111`	(available for user software)
`10000xxx`	Configure the on board FPAA using the following configuration data stream.
`10010xnn`	Configure the on board FPAA using Flash file nn.
`10100xnn`	Program Flash file nn using the following configuration data stream.

Unless otherwise noted in the table above, all other SPI Command words are reserved for factory use.

The SPI Command word is constructed as follows:

Bit7	6	5	4	3	2	1	Bit 0
ABK	R/W	CH2	CH1	CH0	ID2	ID1	ID0

ABK - ABK Service Request
- 0 following data packet directed to user software
- 1 following data packet directed to ABK

R/W - Read Write
- 0 data packet is a write
- 1 data packet is a read

CH[2:0] - ABK Resource Channel
- 0 On Board FPAA
- 1 (factory reserved)
- 2 Configure On Board FPAA with Flash File
- 3 (factory reserved)
- 4 Flash File
- 5 (factory reserved)
- 6 (factory reserved)
- 7 (factory reserved)

ID[2:0] - ABK Resource ID

The logical connection

The following represents a typical download sequence via the SPI Slave port. All data transferred across SPI is assumed to be in its binary representation. During the download, the slaved SPI port will continuously broadcast a download status byte on the MISO line.

1. Transmit one of the command headers described above.
2. Transmit the .AHF file (every pair of ASCII Hex characters must be translated to one binary byte).

The ABK expects the entire configuration file to be transferred. If there is a break in the data stream longer than 10 seconds, the firmware assumes there is some problem and takes the following actions:
1. loads a fail byte code into the SPI's MOSI transmit register (anything but 0xC0 a failure code)
2. abandons the download
3. clears the Flash segment to a safe state (if Flash was the target of the download)
4. asserts the Red Error LED
5. reverts back to IDLE State

Appendix A - Source File Descriptions

Only the most advanced users should have any interest in this information. All of the source files are provided for the microcontroller's firmware to facilitate user customization.

makefile
A standard makefile for compiling the ABK.

mC68hc908gp32.equ
This file contains equates. These equates translate addresses and data structures to human readable tokens.

mC68hc908gp32.h
Port and peripheral definitions. Again, the tokens and data structures defined herein make the C source code much easier to read.

startup.s
Assembler for start up functions, the sort of things that run before Main. The file also contains project dependent low level assembly routines for programming Flash etc.

abk.c and **abk.h**
These files provide the ABK main control loop and global definitions.

array.c and **array.h**
These files provide the AN10E40 array hardware control definitions and function prototypes.

hardware.c and **hardware.h**
These files provide the microcontroller project dependent hardware definitions and function prototypes.

host.c and **host.h**
These files provide the AN10E40 array host interface control definitions and function prototypes.

spi.c and **spi.h**
These files provide the AN10E40 array SPI control definitions and function prototypes.

Appendix B - Motorola S-Record Format

Throughout most of this document, references are made to the .AHF or ASCII Hex File configuration file format. The .AHF file format is understood almost universally by various device programming appliances. A secondn industry standard format for transferring programming information between tools is the Motorola S-Record format. The file is easy to understand once you have been introduced to it. The first and last couple of lines of a S-Record format file for an AN10E40 are shown below.

```
S224000000138502B70000000000000000000000000000000000000000000000000000000000000026
S22400002000000000000000000000000000000000000000000000000000000000000000000000BB
. . .
S20E000360080000080000080000 1462
S9030000FC
```

A field by field dissection of the first and last S-Records within an AN10E40 configuration file is given below:

```
S224000000138502B70000000000000000000000000000000000000000000000000000000000000026
```

S2 → tells downstream tools that this record's address field is 24 bits in length (S1 indicates 16 bits of address)

24 → tells downstream tools that 36 (hex 24) bytes follow.

000000 → tells downstream tools that the starting address for this data is 0x000000.

138502B7 → Not part of the S-Record spec, but this is first of the device specific data.

138502B7 → For Anadigm FPAAs, these 8 bytes represent a unique

138502B7 → device type identifier.

26

This is the end of S-Record check byte ← 26

```
S9030000FC
```

S9 → tells downstream tools that this is the termination record for the file.

03 → byte count for remainder of record

0000 → dummy data

FC→ checkbyte

Appendix C - Board Schematic

NOTES:

All resistors are surface mount 0805 package unless otherwise specified.

All capacitors are ceramic NPO, surface mount 1206 package unless otherwise specified.

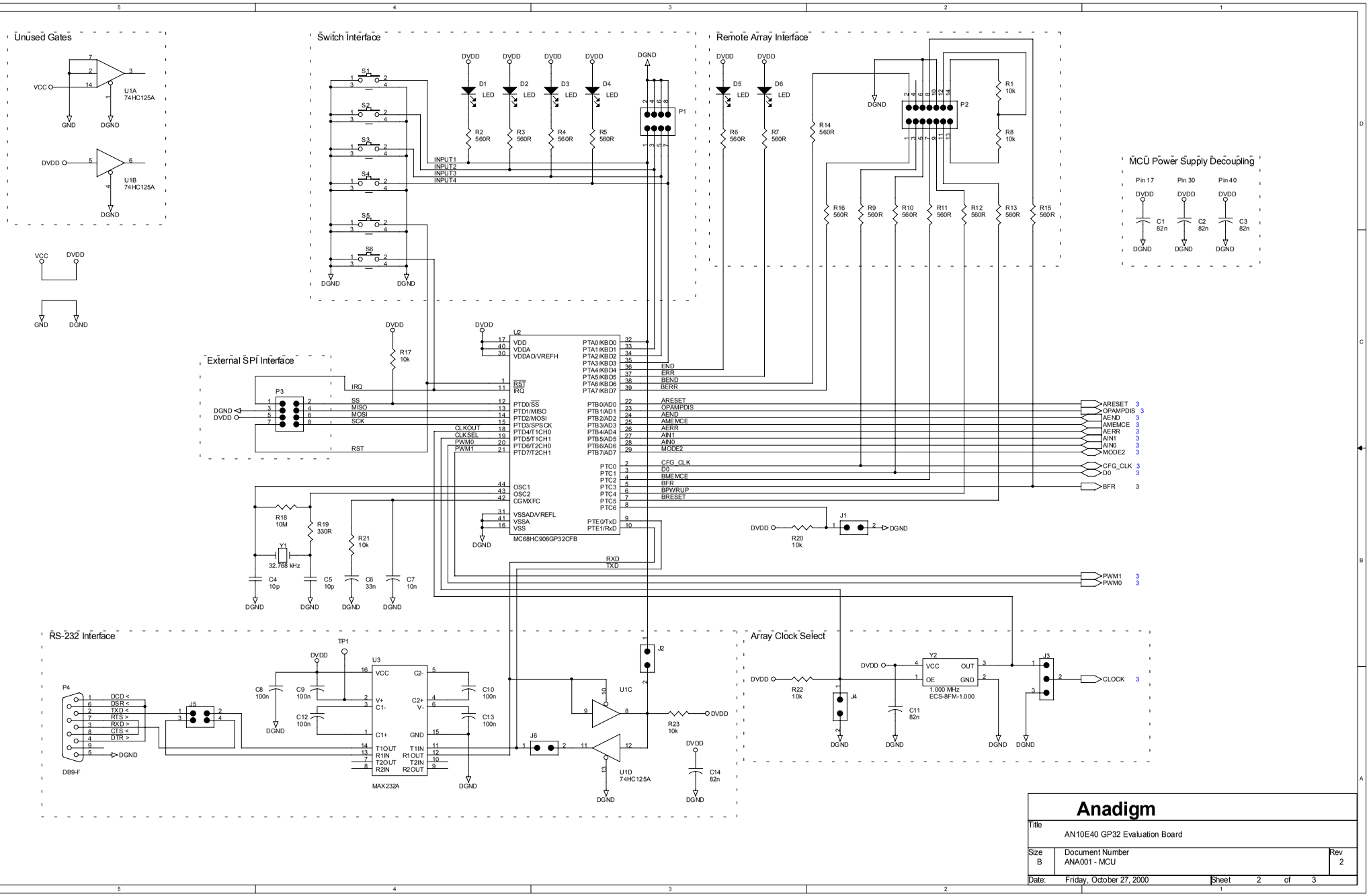
Anadigm Analog Array
AN10E40 GP32 Evaluation Board

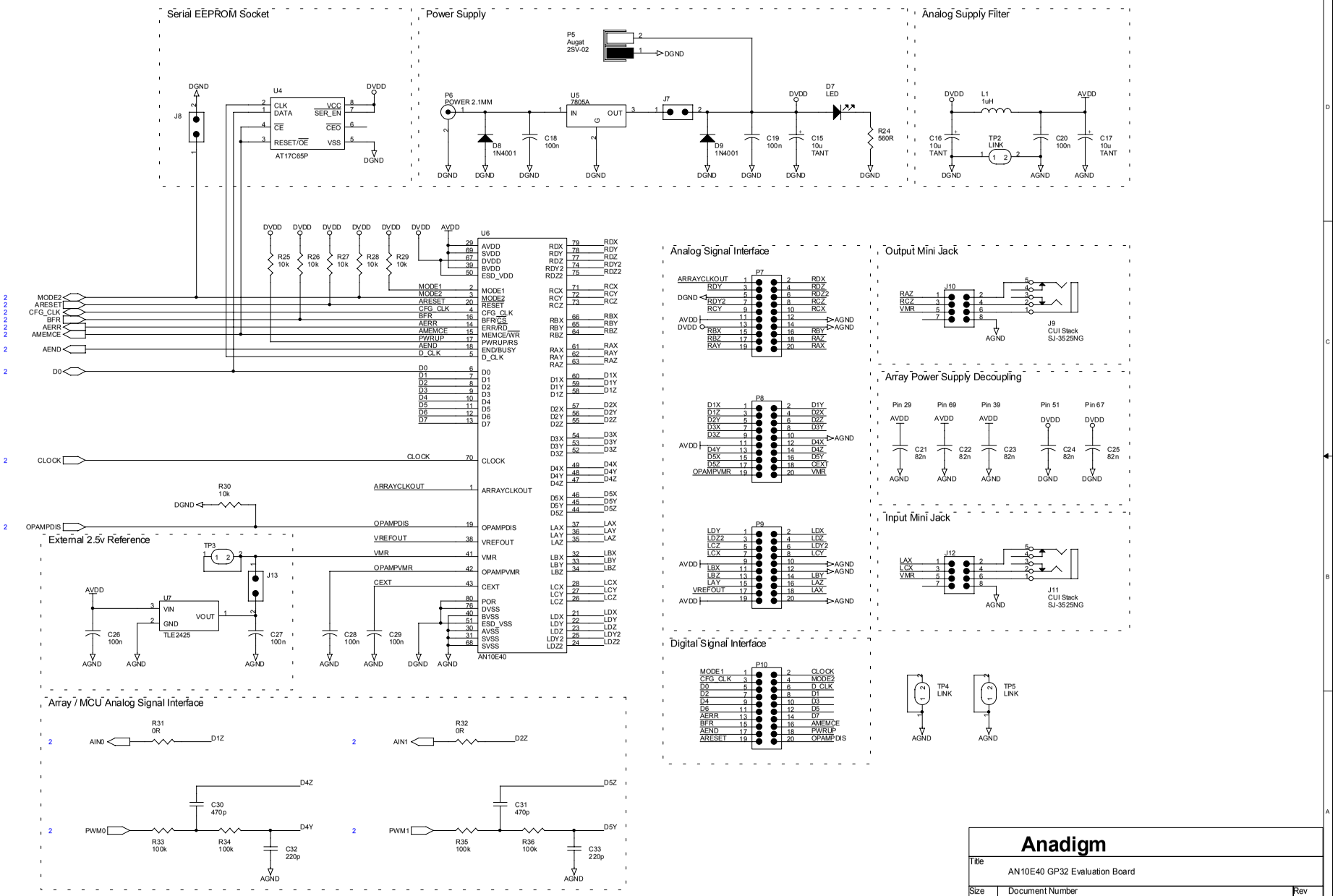
Revisions

Rev	By	Description	Date
0	DHJK	Initial Proposal	08-May-00
1	DHJK	Subtle Circuit Improvements and Corrections	18-May-00
2	DHJK	Added Digital Signal Interface and Power Supply terminal. Changed 2.5v reference and provided more component details.	29-May-00
3	DHJK	Added protection diodes D8 & D9.	27-Oct-00

Anadigm

Title	AN10E40 GP32 Evaluation Board	
Size B	Document Number ANA001 - Cover	Rev 3
Date: Friday, October 27, 2000	Sheet 1 of 3	


Unused Gates
U1A 74HC125A
U1B 74HC125A
Switch Interface
Remote Array Interface
MCU Power Supply Decoupling
External SPI Interface
RS-232 Interface
Array Clock Select
MC68HC908GP32CFB
MAX232A
DB9-F
1.000 MHz ECS-8FM-1.000
Anadigm
AN10E40 GP32 Evaluation Board
ANA001 - MCU
Friday, October 27, 2000


Serial EEPROM Socket
Power Supply
Analog Supply Filter
Analog Signal Interface
Output Mini Jack
Array Power Supply Decoupling
Input Mini Jack
External 2.5v Reference
Digital Signal Interface
Array / MCU Analog Signal Interface
AT17C65P
TLE2425
AN10E40
Anadigm
AN10E40 GP32 Evaluation Board
Friday, October 27, 2000

Anadigm is pleased to offer our
customers direct access to the following offices:

WEB
http://www.anadigm.com/

USA
Anadigm Inc.
21615 Stevens Creek Blvd
Cupertino
CA 95014

Tel: +1 408 996 2091
Fax: +1 408 996 2093

Anadigm Inc.
155 East Chilton Drive
Suite 201
Chandler
AZ 85225-1115

Tel: +1 480 545 6730
Fax: +1 480 545 2915

UK
Anadigm Ltd.
Scott House
Westmere Drive
Crewe
CW1 6ZG

Tel: +44 (0) 1270 531990
Fax: +44 (0) 1270 531999

GERMANY
Anadigm Ltd.
Gottlieb-Daimler Str. 6
82140 Olching

Tel: +49 (0) 8142 4485830
Fax: +49 (0) 8142 4485840