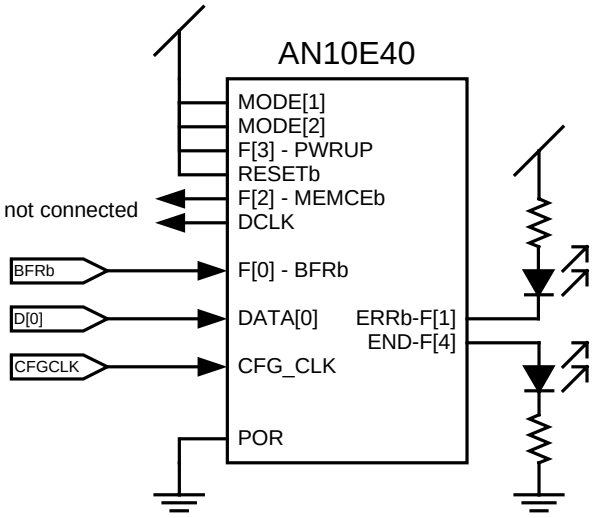
App Note 002

Applying the AN10E40
Configuring with the Minimum Number of Connections

Introduction

This application note gives the details necessary for booting an AN10E40 FPAA from a host processor using the fewest possible number of active connections. It is possible to boot the device by supplying only a start signal, a configuration clock and serial data. The described method is not the intended connection scheme nor is it preferred, but it can be used when only three logic signals are available from the hosting system.



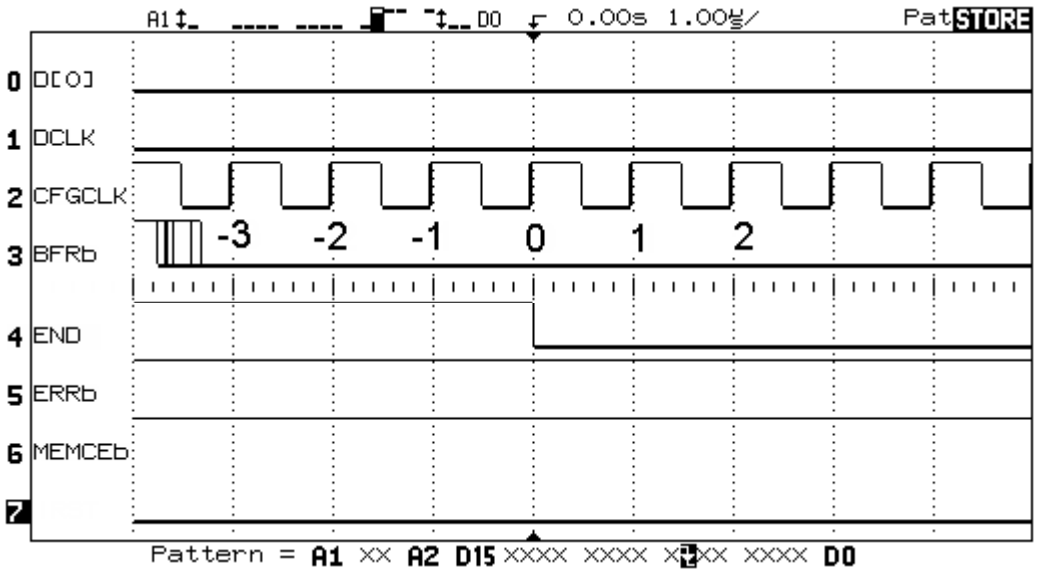
The Connections

The AN10E40 has two configuration modes. Micro mode and Boot from ROM (BFR) mode. In Micro mode, the AN10E40 presents itself as an external peripheral, usually attached to a byte wide data bus. With 8 signals required for data, and another 5 for bus protocol, the Micro mode does not lend itself to situations in which only a few signal lines are available.

Instead, BFR mode is used. Only the AN10E40's BFRb, CFG_CLK and DATA[0] pins are driven by the hosting logic. All return direction handshaking signals are left disconnected. Data transfer in the BFR mode is synchronous and deterministic, but since all handshaking is ignored, the host system must carefully count CFG_CLK edges and present configuration data as described herein. In this example, BFRb is brought low and CFG_CLK (shown as CFGCLK) and DATA[0] (shown as D[0]) are driven by host logic.

The Pre-Boot Sequence

The host system must drive CFG_CLK continuously after reset and for not less than 2^{17} cycles. The AN10E40 has an analog power-on reset circuit followed by a 17 bit counter. Once the internal analog power-on reset signal asserts, the counter begins counting up while the on-chip configuration circuitry continuously clears the internal configuration SRAM. The counter is long enough for nearly any system's power to reach a stable and safe operating point.



The figure above assumes power is stable and that at least 2^{17} CFG_CLK cycles have passed. The figure shows the initiation of a manual Boot From ROM sequence. Presuming the BFRb signal meets the setup up time with respect to CFG_CLK (20 ns), END will go low 4 rising edges of CFG_CLK later.

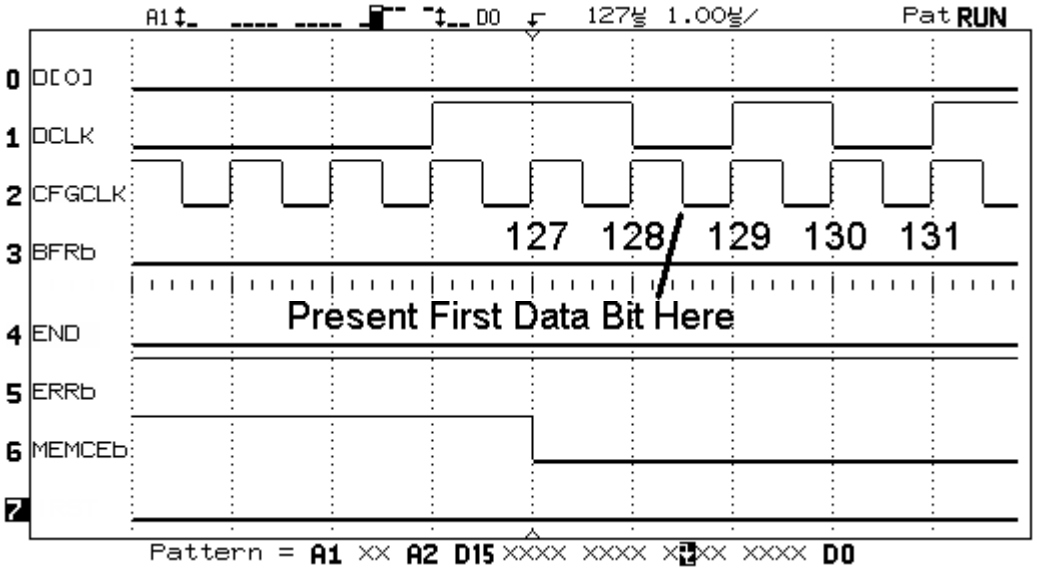
The above screen shot is actually a composite image of multiple trials. The logic analyzer was set to trigger on the falling edge of END, while the BFRb signal was asserted several times asynchronously. Consequently you see several traces of the BFRb signal dropping before CFG_CLK edge number -3. All measurements given in this note are with respect to rising edge of CFG_CLK 0 (CFG_CLK 0↑).

Just what data to present is discussed a bit later in the section, The ASCII Hex File (AHF) Configuration Data File Format. For now let us just focus on the clock to data relationship of that data.

Presenting Data

In this configuration, the AN10E40 derives its DCLK output from the CFG_CLK input. The device clocks in the serial data always on the rising edge of DCLK. Since the timing relationship between a DCLK rising edge and a CFG_CLK rising edge is not characterized, presenting new data concurrent with a rising CFG_CLK represents a timing hazard. For appropriate setup margin you should instead present new data on the falling edge of CFG_CLK.

127 CFG_CLKs after the AN10E40 dropped END, it will drop MEMCEb. MEMCEb is normally tied to a Serial EPROM enable input, and it is at this time the transfer of data should begin. You should present the first configuration data bit at CFG_CLK 128↓. In the example below, the first several data bits are zero, so there are no DATA[0] transitions yet visible.

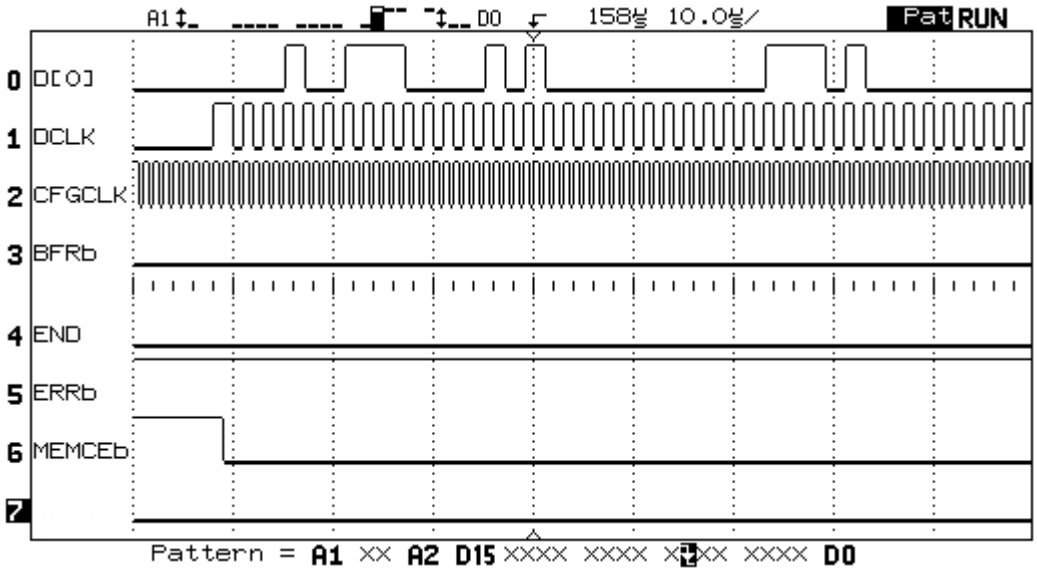


The AN10E40 generates the DCLK output from the CFG_CLK input. DCLK is not used in this particular application, but it is shown for reference. Please note the 2:1 relationship between the CFG_CLK and DCLK. Once MEMCEb asserts low, The AN10E40 always latches in data on the rising edge of DCLK (129↑, 131↑…).

The first data to get clocked into the array makes up a device ID number and a data type byte.

Zooming out a bit (shown just to the right), you see the DATA[0] data start to stream into the array. In this case the device ID of 1385001D is streaming in. The bits are always presented MSB first. Reading the DCLK (rising edge) & DATA[0] waveform, you should see:

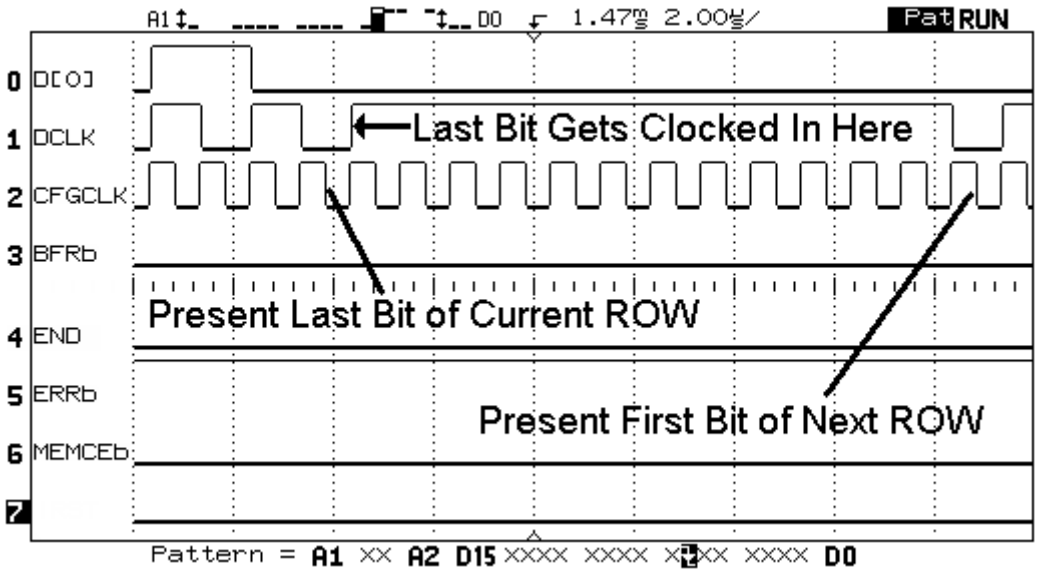
0001 (1) 0011 (3) 1000 (8) etc….



note - 138501D is a device ID for a pre-production design. The AN10E40 will have a device ID of 138502B7.

End of Row Timing

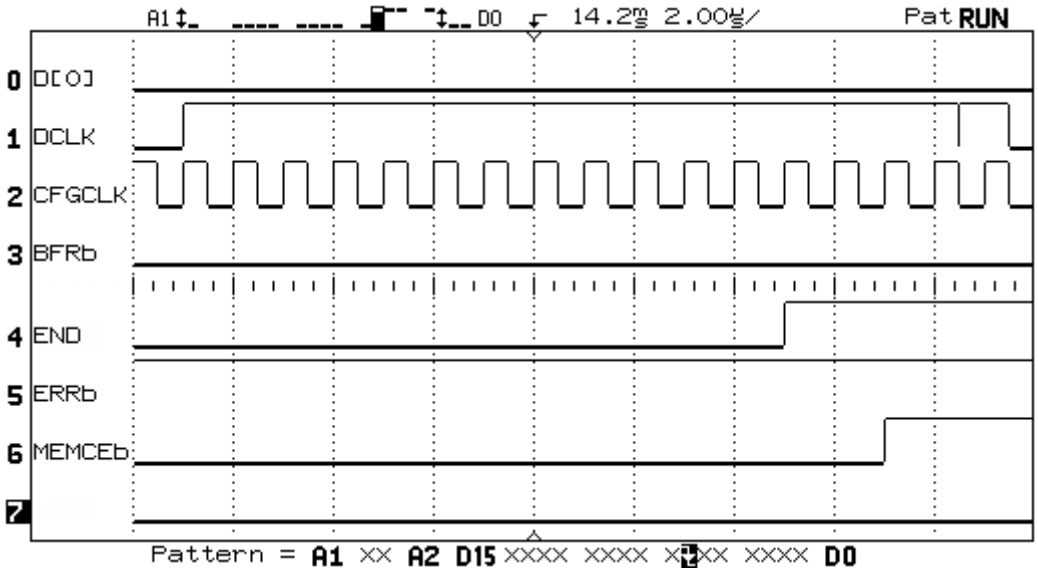
After the Device ID and Data Type bytes have been successfully digested by the FPAA's configuration circuitry, all subsequent configuration data gets routed through a checksum calculator and on into a configuration memory shift register. The AN10E40 has a configuration memory shift register that is 78 bytes long. During download, the shift register fills while a checksum 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 FPAA's configuration SRAM, and the shift register is again ready to accept the next row's data. If there is ever a checksum mismatch, the FPAA asserts its error signal (ERRb), aborts the download, and any subsequent serial data will be ignored. The AN10E40 has 11 rows of configuration SRAM that need to be loaded in this manner.



The figure above shows end of row behavior. You will want to present the last data bit for the current row on the edge shown. The last rising edge of DCLK before the pause, is where the AN10E40 will latch in the last bit for the current row. You may present the first bit of the next row anytime during or just after the DCLK pause . It will be latched on the next rising edge of DCLK.

End of Array Timing

The figure below shows the very end of the array load. During the last dozen or so CFG_CLK's after the last bit is loaded in, the AN10E40 de-asserts END, then de-asserts MEMCEb. The downloaded analog design is alive and running on the de-assertion of MEMCEb.



The ASCII Hex File (AHF) Configuration Data File Format

Now that you know *how* to clock in the data, let us turn our attention to *what* data to clock in. The AnadigmDesigner design software generates configuration files for programming serial boot PROMs, or in this case, for manual serial booting. The file format that is easiest to use is ASCII Hex File format, also known as AHF. 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 recorded as "F7". A prototype 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.

```
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. You should ignore white space characters in an AHF file.

Next come 11 rows worth of configuration data. Each row of data is 78 bytes long plus one check sum byte. If the checksum byte does not match the internally calculated checksum value then again (like a mismatched device ID code) the array will assert the ERRb pin, halt its configuration engine and ignore all subsequent data.

The Cook Book Approach
This final section walks you through the cook book approach to clocking in data.

With adequate set up time with respect to CFG_CLK (20 ns), assert BFRb low. Four rising CFG_CLK edges later (this is called rising CFG_CLK 0↑), END will de-assert low . At falling edge 128↓, present the MSB of the first byte of the AHF file. Hold the bit until replaced by the next bit at falling edge 129↓ etc. For the sake of brevity, the complete rows are not shown. For example: Row 0 loading spans CFG_CLK 128 through 1470, Row 1 spans CFG_CLK 1483 through 2745. New data should be presented every 2nd CFG_CLK edge, except during end of row DCLK hold offs.

Present Bit Num.	CFG_CLK Edge	Note
(none)	-3↑	Assert BFRb, with > 20nS setup time
(none)	0↑	END asserts low
-	127↑	MEMCEb asserts low,
1	128↓	Present first Bit of Row 0
1	129↑	first Bit of Row 0 is clocked in by the AN10E40
2	130↓	Present second Bit of Row 0
672	1470↓	Present last Bit of Row 0
		End of Row 0 DCLK hold off
673	1483↓	Present First Bit of Row 1 Data
1304	2745↓	
		End if Row 1 DCLK hold off
1305	2758↓	Row 2 Data
1936	4020↓	
		End of Row 2 DCLK hold off
1937	4033↓	Row 3 Data
2568	5295↓	
		End of Row 3 DCLK hold off
2569	5308↓	Row 4 Data
3200	6570↓	
		End of Row 4 DCLK hold off
3201	6583↓	Row 5 Data
3832	7845↓	
		End of Row 5 DCLK hold off
3833	7858↓	Row 6 Data
4464	9120↓	
		End of Row 6 DCLK hold off
4465	9133↓	Row 7 Data
5096	10395↓	
		End of Row 7 DCLK hold off
5097	10408↓	Row 8 Data
5728	11670↓	
		End of Row 7 DCLK hold off
5729	11683↓	Row 9 Data
6360	12945↓	
		End of Row 9 DCLK hold off
6361	12958↓	Row 10 Data
6992	14220↓	Present Last Bit
(none)	14235↑	Analog Array Active

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