Emulation Tech Note 8
Using SdConfigEx

Document Revision 0.01
Jan 30, 2006

1. Introduction

To support the advanced features of the XDS510USB PLUS emulator and to simplify configuration of the current XDS510USB emulator, Spectrum Digital has extended SdConfig with SdConfigEx. The architecture of SdConfig was extended such that the look and feel of SdConfigEx could be easily modified via XML configuration files. This allows for future option extensions and possible customization by the customer.

2. Features and Differences

The key differences between SdConfig and SdConfigEx are summarized below.

- SdConfigEx is optimized for the XDS510USB class emulators.
- In SdConfig emulator features are accessed from the GUI based on the emulator port address. In SdConfigEx the emulator options are accessed by the type of emulator which is little more intuitive. See figure 1.
- SdConfigEx has extended the emulator options for the XDS510USB class emulators.
- SdConfigEx has removed the Data Logging options as they do not apply to XDS510USB class emulators.
- SdConfigEx has removed support for the SPI510 (ISA) and eZdsps.
- SdConfigEx supports the following emulators:
    - XDS510PP
    - XDS510PP PLUS
    - SPI515
    - SPI525
    - XDS510USB
    - XDS510USB PLUS
    - XDS510USB GALVANIC
- SdConfig and SdConfigEx still use the sdopts.cfg file to store emulator options and all options may still be configured manually.

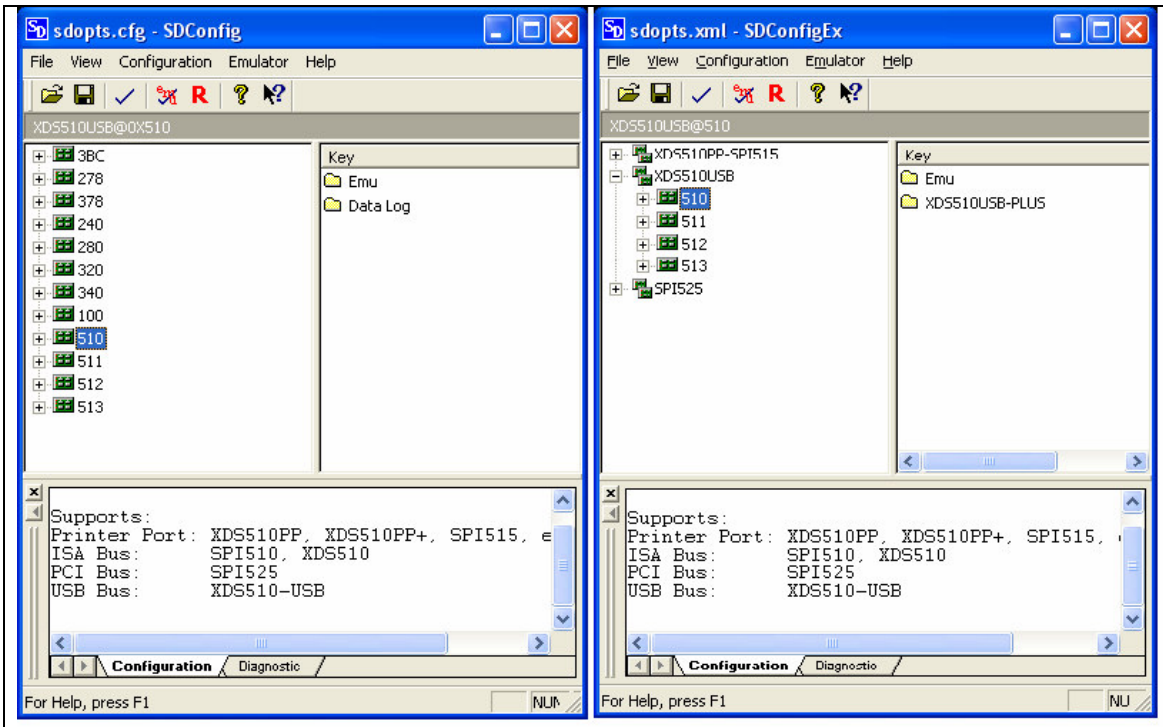

Figure 1 SdConfig vs SdConfigEx Layout

3. Installation

The SdConfigEx utility is installed as part of your CCS driver install for driver releases 3.xx.15 and higher. The SdConfigEx utility will be located in your <ccs_install_dir>\specdig\sdconfigex directory.

4. Setting XDS510USB PLUS EMULATOR OPTIONS

Under the XDS510USB option each port has two setup panels, Emu and XDS510USB-PLUS. The Emu options apply to all XDS510USB class emulators. Under the XDS510USB-PLUS panel the following settings apply to both the XDS510USB and XDS510USB-PLUS:

- TCK Divisor/Adaptive TCK Delay
- JTAG I/O OFF on exit
- Emulator Serial Number

All other options apply only to the XDS510USB PLUS.

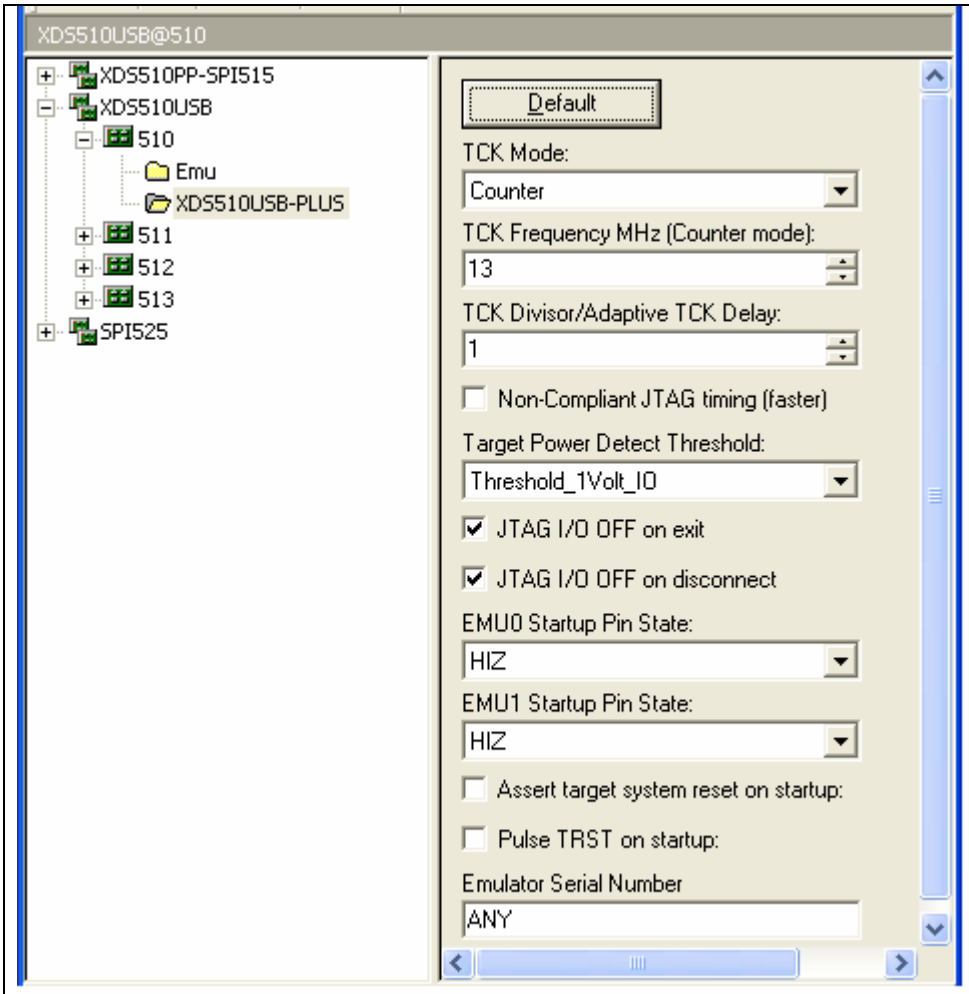

Figure 2 XDS510USB PLUS Settings

4.1 Selecting TCK MODE

The TCK mode defines how the emulator will generate the TCK to the target. There are three options:

- **Counter**: Traditional method of generating TCK from counter source to generate a specified frequency.
- **Adaptive**: ARM Ltd style adaptive clocking with programmable RTCK to TCK delay.
- **Fast Adaptive**: ARM Ltd style adaptive clocking with no programmable RTCK to TCK delay.

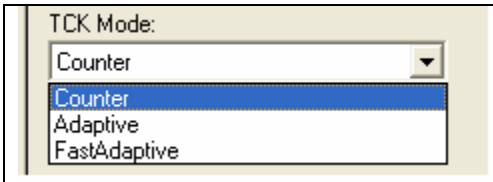

Figure 3 TCK Mode

4.1.1 TCK - Counter Mode

In Counter Mode the TCK is driven by a clock counter/timer which can be programmed to generate frequencies from 1 to 32 MHz. The actual frequency driven on TCK will be less than or equal to the selected frequency and may not be exactly the frequency selected. Counter Mode takes the TCK Frequency MHz option as its frequency parameter.

4.1.2 TCK – Adaptive Mode

In Adaptive Mode the TCK is generated adaptively from the RTCK (TCLK_RET) signal. In this mode the emulator generates a TCK edge and waits for the target to send RTCK. When RTCK equals TCK the emulator will wait N-clocks and generate an inverted TCK. This method is most commonly referred to as ARM Ltd style adaptive clocking and allows the TCK to track RTCK which may be slaved to the operating frequency of the target processor. Adaptive Mode takes the TCK Divisor/Adaptive TCK Delay option as its RTCK to TCK N-clock delay. Each N-clock delay will add approximately 13ns of separation between RTCK and TCK. This delay can be used to compensate for device specific setup and hold times requirements.

4.1.3 TCK – Fast Adaptive Mode

The Fast Adaptive Mode is the same as Adaptive Mode except there is no programmable delay between RTCK and TCK. This yields a faster TCK/RTCK frequency but may also violate setup and hold times on the target device JTAG timings.

4.1.4 TCK Frequency MHz (Counter Mode)

The TCK Frequency parameter sets the TCK frequency when in TCK-Counter mode. The selection is from 1 MHz to 32 MHz. The actual frequency driven on TCK will be less than or equal to the selected frequency and may not be exactly the frequency selected.

4.1.5 TCK Divisor/Adaptive TCK Delay

The TCK Divisor/Adaptive TCK Delay sets the N-clock delay between RTCK and TCK when in TCK-Adaptive Mode. Each N-clock delay will add approximately 13ns of separation between RTCK and TCK.

When used with an XDS510USB, XDS510USB Galvanic or SPI515 emulator the option sets the EmuTckDiv parameter of the sdopts.cfg file. The resulting TCK frequency will be defined as 12 MHz / TCK Divisor.

4.1.6 Non-Compliant JTAG timing

The default JTAG signal timing is to drive TMS and TDI from the falling edge of TCK for a standard master-slave timing model. When this timing model is used simply slowing down TCK will compensate for long cable and buffer delays in the target system. This is the most robust timing model but is also slower then non-compliant timing. By selecting non-compliant timing the TMS and TDI are driven from the rising edge of TCK. This generally results in a higher TCK frequency but can cause TMS and TDI hold time violations. In general non-compliant timing can be used in single chip systems where the delays on TCK and RTCK (TCK_RET) are balanced.

Testing has shown that emulator performance increases with increase in TCK is fairly linear from 1 MHz to 15 MHz. The emulator performance rolls off in the 15 MHz to 18 MHz range and from 20 MHz to 32 MHz emulator performance may only increase by 10 percent. This phenomenon appears to be fairly consistent across XDS510 and XDS560 class emulators with some variation in the roll off point for a given processor. Given this phenomenon it may not make sense to configure the emulator for fastest TCK and non-compliant timing when a 15 MHz TCK with compliant timing will get you 90% of the emulator performance.

4.1.7 Target Power Detect Threshold

The XDS510USB PLUS has dual power detectors optimized for 1.8 volt and 3.3 volt I/O systems. When the Threshold_1Volt_IO option is selected the emulator will sense a target power failure if the voltage at the PD pin falls below approximately 1.1 volts. When the Threshold_3Volt_IO option is selected the emulator will sense a power failure if the voltage at the PD pin falls below approximately 2.1 volts.

The additional higher level threshold option is useful on older target systems that may have I/O leakage paths back to the VCC rails. History has shown that in these older systems when power on the target is removed, the JTAG signals, primarily TCK can supply enough current through the I/O pins to the VCC rail to keep the system powered at approximately 1.5 volts. In this case the 1.1 volt power detect will not sense that target power has been lost. By selecting the higher threshold the emulator will sense the power loss and drive the JTAG signals to low levels. Generally speaking newer chip designs do not have I/O leakage paths to the VCC rails so I/O leakages issues are not that common.

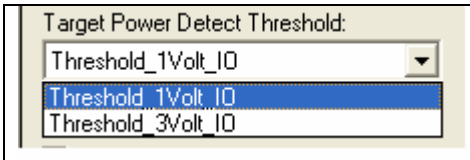

Figure 4 Target Power Detect Threshold

4.1.8 JTAG I/O OFF on exit

This parameter applies to all XDS510USB class emulators and corresponds to the EmuIoOffAtExit parameter in sdopts.cfg. When checked the XDS510USB class emulators will turn off the JTAG backend when an emulation session is not active. This is equivalent to electrically disconnecting the emulator from the target system. This method also allows the emulator to achieve its lowest USB power consumption when an emulation session is not active. When unchecked the XDS510USB class emulator will maintain power and drive to the JTAG signals. This method is only required when it is necessary to maintain the state of the JTAG signals across emulation sessions. In general this is only necessary for some OMAP targets to work around adaptive clocking startup problems or support OMAP devices with IcePick-A.

4.1.9 JTAG I/O OFF on disconnect

Reserved for future use to selectively distinguish between a CCS exit and a CCS disconnect.

4.1.10 EMU0 and EMU1 Startup Pin State

This option allows you to select the state of the EMU pins when the JTAG backend becomes active. The EMUx pin option is primarily provided for board level boundary scan testing. There are three startup options:

- HIZ: Default, emulator does not drive the EMUx pins.
- HIGH: Emulator will drive the EMUx pins high on startup.
- LOW: Emulator will drive the EMUx pins low on startup.

When the emulation session begins the EMUx pins along with other JTAG pins are set to a known state and then sequenced as required for emulation. When the EMUx pins are not set to the HIZ configuration they will be set as specified by the user while the TRSTn pin is driven low. During the emulation startup the TRSTn pin will be driven high and then the EMUx pins will be released to the HIZ state.

The function of the EMUx pins on TI devices is not entirely consistent or predictable for all devices. In many cases the EMUx pin definitions between processor families conflict and makes than unusable in mixed processor systems. The only consistent configuration is that EMU0/1 both high enables emulation mode. The EMUx pins should always be configured for HIZ for default operation with CCS.

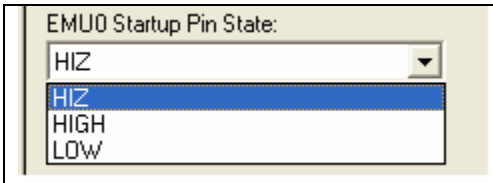

Figure 5 EMUx Startup Pin State

4.1.11 Assert target system reset on startup

This option is only available when a CTI-20 pin cable is used, i.e. primarily for DaVinici target systems. When selected the SRST pin of the CTI-20 pin cable will be pulled low through an open-drain gate and 100 ohm resistor to ground. If the SRST signal is connected into the target board/processor reset circuitry then the emulator can perform a true hardware reset of the target system. When selected the startup state of the SRST, TRSTn and EMUx pins follow the following sequence:

- Assert the TRSTn and SRST pins
- Set the EMUx pins
- Delay minimum of 2ms
- Release SRST pin to HIZ
- Delay minimum of 20ms
- Drive the TRSTn pin high for emulation mode
- Delay minimum of 2ms
- Release the EMUx pins to HIZ

This specific sequence is performed to take advantage of TI wait-in-reset mode which is available on some processors. The TI wait-in-reset mode generally requires that the EMUx and potentially the TRSTn pins be sampled on the rising of the processor reset. If implemented then the target processor is held in a reset condition until specifically released by the emulation driver when CCS is started. Please note that wait-in-reset is only available on selected TI processors and also suffers the same in-consistency of the EMUx pins.

4.1.12 Pulse TRST on startup

When selected the TRSTn pin will be pulsed high-low-high during emulation startup. Some TI processors need to see a rising-falling-rising edge sequence on the TRSTn pin to clear and set specific emulation modes which are related to the EMUx pins.

4.1.13 Emulator Serial Number

The XDS510USB class emulators allow you to connect to an emulator via its serial number. This enables the use of up to 4 XDS510USB class emulators connected to one PC. This parameter corresponds to the EmuSerialNumber option in the sdopts.cfg file and allows mapping of an emulator port address to a specific XDS510USB unit via it's serial number. Multiple XDS510USB emulator setup is described in Spectrum Digital TechNote_4.pdf. When this option is set to ANY then any XDS510USB class emulator connected to the PC will be addressed via port address 0x510.

To find the XDS510USB units serial number, from SdConfigEx select Configuration->Ports Available->Printer,USB. If an XDS510SUB class emulator is found it will return a EmuSerialNumber string to the output window. You can simply cut the serial number (right of the equal sign) and paste into the option field.

5. XDS510USB PLUS Manual Configuration

All options for configuring SD emulators can be manually set by editing the sdopts.cfg file which is located in the windows\system 32 directory. The XDS510USB PLUS configurations discussed in chapter 4 can also be set manually as described in this chapter

5.1 TCK Mode

```
Option:           EmuJclkMode
Valid Settings:   Counter
                  Adaptive
                  FastAdaptive
```

5.2 TCK Frequency MHz

```
Option:           EmuJclkFreqMHz
Valid Settings:   1 to 32
```

5.3 TCK Divisor/Adaptive TCK Delay

Option: EmuTckDiv
Valid Settings: 1 to 12

5.4 Non-Compliant JTAG timing

Option: EmuAltTckTiming
Valid Settings: YES or No

5.5 Target Power Detect Threshold

Option: EmuPowerMode
Valid Settings: Threshold_1Volt_IO or Threshold_3Volt_IO

5.6 JTAG I/O OFF on exit

Option: EmuIoOffAtExit
Valid Settings: YES or No

5.7 JTAG I/O OFF on disconnect

Option: EmuIoOffAtDisconnect
Valid Settings: YES or NO

5.8 EMU1 Startup Pin State

Option: Emu0PinLevel
Valid Settings: HIZ, HIGH, LOW

5.9 EMU0 Startup Pin State

Option: Emu1PinLevel
Valid Settings: HIZ, HIGH, LOW

5.10 Assert target system reset on startup

Option: EmuAssertSysReset
Valid Settings: YES or NO

5.11 Pulse TRST on startup

Option: EmuPulseTrst
Valid Settings: YES or NO

5.12 Emulator Serial Number

Option: EmuSerialNumber
Valid Settings: ANY or serial number string as reported by SdConfigEx