**MIPSfpga+ ADC Interface Module for Altera MAX10 FPGA (ADC\_MAX10)**

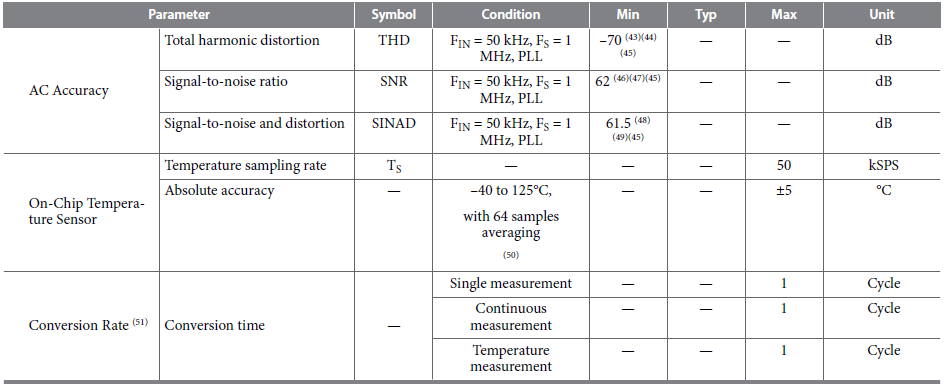
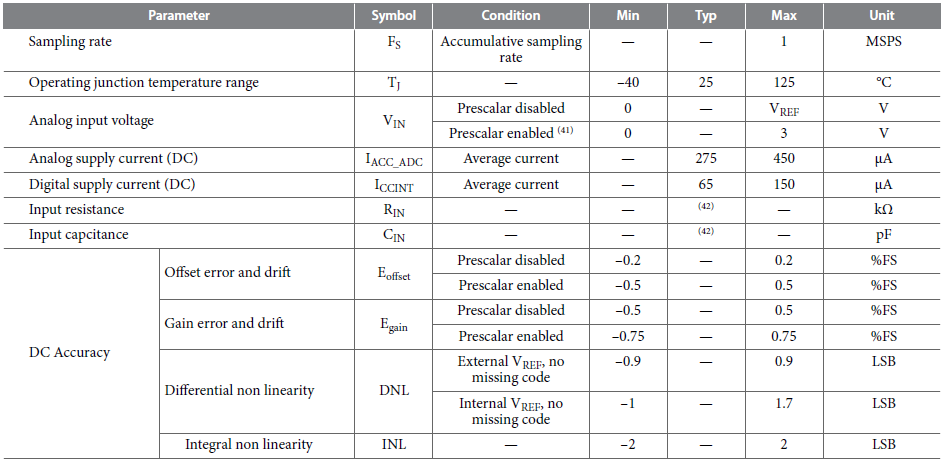
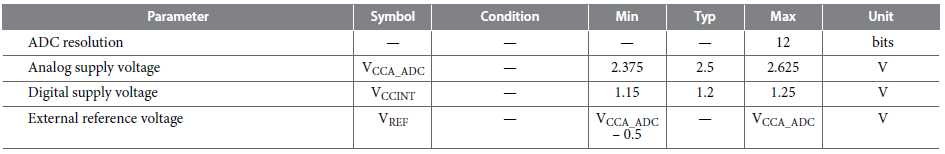
**Main Features**

* Small. The module core have about 250 lines of Verilog code);
* Simple. Module register interface (architecture) is very similar to Atmel ATmega88 devices ADC block;
* Supports up to 10 input pins (1 dedicated, 8 dual purpose analog input pins, 1 device temperature sensor);
* Supports ADC conversion speed up to 1 MSPS (in Free Running mode) and up to 0.33 MSPS when conversion starts by external trigger input;
* ADC Conversion End interrupt signal support;
* Checked on Terasic DE10-Lite board (Altera MAX10 10M50DAF484C7G FPGA device);
* Merged to MIPSfpga+ system: <https://github.com/MIPSfpga/mipsfpga-plus>
* Project for standalone ADC core debug: <https://github.com/zhelnio/ahb_lite_adc_max10>

**Module Interface Signals**

|  |  |  |
| --- | --- | --- |
| **Signal Name** | **Type** | **Description** |
| **AHB-Lite Interface Signals** | | |
| HCLK | input | Clock input. |
| HRESETn | input | The bus reset signal is active LOW |
| HADDR [31:0] | input | The 32-bit system address bus |
| HBURST [2:0] | input | Ignored |
| HMASTLOCK | input | Ignored |
| HPROT [3:0] | input | Ignored |
| HSEL | input | Peripheral device select signal |
| HSIZE [2:0] | input | Ignored. Only x32 operations are supported |
| HTRANS [1:0] | input | Indicates the transfer type of the current transfer |
| HWDATA [31:0] | input | The write data bus transfers data from the master to the slaves during write operations. |
| HWRITE | input | Indicates the transfer direction. When HIGH this signal indicates a write transfer  and when LOW a read transfer. |
| HRDATA [31:0] | output | Read Data |
| HREADY | output | When HIGH, the HREADY signal indicates that a transfer has finished on the  bus. This signal can be driven LOW to extend a transfer. |
| HRESP | output | The transfer response. When LOW, the HRESP signal indicates that the transfer  status is OKAY. |
| SI\_Endian | input | Ignored |
| **Altera MAX10 ADC Control Core Interface Signals (Avalon-ST bus)** | | |
| ADC\_C\_Valid | output | Command Stream. Indication from the source port that current transfer is valid. |
| ADC\_C\_Channel [4:0] | output | Command Stream. Indicates the channel that the ADC hard block samples from for current command:  • 17—temperature sensor  • 16:0—channels 16 to 0; where channel 0 is the dedicated analog input pin and channels 1 to 16 are the dual purpose analog input pins. **In current realization only 8:0 are supported.** |
| ADC\_C\_SOP | output | Command Stream. Indication from the source port that current transfer is the start of packet. |
| ADC\_C\_EOP | output | Command Stream. Indication from the source port that current transfer is the end of packet. |
| ADC\_C\_Ready | input | Command Stream. Indication from the sink port that it is ready for current transfer. |
| ADC\_R\_Valid | input | Response Stream. Indication from the source port that current transfer is valid. |
| ADC\_R\_Channel [4:0 ] | input | Response Stream. Indicates the ADC channel to which the ADC sampling data  corresponds for the current response.  • 17—temperature sensor  • 16:0—channels 16 to 0; where channel 0 is the dedicated analog input pin and channels 1 to 16 are the dual purpose analog input pins. **In current realization only 8:0 are supported.** |
| ADC\_R\_Data [11:0] | input | ADC sampling data |
| ADC\_R\_SOP | input | Response Stream. Indication from the source port that current transfer is the start of packet. |
| ADC\_R\_EOP | input | Response Stream. Indication from the source port that current transfer is the end of packet. |
| **Trigger and Interrupt Signals** | | |
| ADC\_Trigger | input | The signal can be used to start the conversion for all the selected channels when the ADCS.EN = 1 and ADCS.TE = 1. |
| ADC\_Interrupt | output | Indicates the ADC Conversion End interrupt request. This signal is set when ADCS.EN = 1, ADCS.IE = 1, ADC conversion completes for all enabled channels and the data registers are updated. Signal is cleared by writing a logical one to the ADCS.IF flag. |

**Altera MAX10 FPGA ADC Performance Specifications (Dual Supply Devices)**



For more details about interface signals and MAX10 FPGA device, refer to:

* MIPS32 microAptiv UP Processor Core Family Integrator’s Guide;
* MAX 10 FPGA Device Handbook;
* MAX 10 FPGA Device Datasheet;
* Avalon Interface Specifications;
* MIPSfpga+ source code.

**ADC Interface Module Register Description**

|  |  |  |  |
| --- | --- | --- | --- |
| **Register** | | **Description** | **Read/Write** |
| **Name** | **Offset** |
| ADCS | 1 | ADC control and status | R/W |
| ADMSK | 2 | ADC channel mask | R/W |
| ADC0 | 3 | ADC dedicated analog input conversion results (channel 0) | R |
| ADC1 | 4 | ADC channel 1 conversion results | R |
| ADC2 | 5 | ADC channel 2 conversion results | R |
| ADC3 | 6 | ADC channel 3 conversion results | R |
| ADC4 | 7 | ADC channel 4 conversion results | R |
| ADC5 | 8 | ADC channel 5 conversion results | R |
| ADC6 | 9 | ADC channel 6 conversion results | R |
| ADC7 | 10 | ADC channel 7 conversion results | R |
| ADC8 | 11 | ADC channel 8 conversion results | R |
| ADCT | 12 | ADC temperature channel conversion results | R |

**ADCS - ADC Control and Status Register**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Fields** | | **Description** | **Read/Write** | **Reset State** |
| **Name** | **Bit(s)** |
| EN | 0 | ADC enable. Writing this bit to one enables the ADC. By writing it to zero, the ADC is turned off. | R/W | 0 |
| SC | 1 | ADC start conversion. In Single Conversion mode, write this bit to one to start conversion for all the enabled in ADMSK channels. In Free Running mode, write this bit to one to start the first conversion. | R/W | 0 |
| TE | 2 | ADC trigger enable. When this bit is written to one, Triggering of the ADC is enabled. The ADC will start a conversion for all the enabled in ADMSK channels on a HIGH level of the *ADC\_Trigger* signal. | R/W | 0 |
| FR | 3 | ADC free running mode. In this mode the conversion runs cyclically for all the enabled in ADMSK channels until ADCS.EN = 1 and ADCS.FR = 1. If ADC Conversion End is enabled it will be called after every cycle end. | R/W | 0 |
| IE | 4 | ADC interrupt enable. Enables the ADC Conversion End interrupt | R/W | 0 |
| IF | 5 | ADC interrupt flag. Indicates the ADC Conversion End interrupt request. This flag is set when ADCS.EN = 1, ADCS.IE = 1, ADC conversion completes for all enabled in ADMSK channels and the data registers (ADC0-ADC8 and ADCT) are updated. Flag is cleared by writing a logical one to it. | R/W | 0 |
| - | 31:6 | Reserved | R | 0 |

**ADMSK - ADC Channel Mask Register**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Fields** | | **Description** | **Read/Write** | **Reset State** |
| **Name** | **Bit(s)** |
| ADMSK0 | 0 | ADC dedicated analog input enable | R/W | 0 |
| ADMSK1 | 1 | ADC channel 1 enable | R/W | 0 |
| ADMSK2 | 2 | ADC channel 2 enable | R/W | 0 |
| ADMSK3 | 3 | ADC channel 3 enable | R/W | 0 |
| ADMSK4 | 4 | ADC channel 4 enable | R/W | 0 |
| ADMSK5 | 5 | ADC channel 5 enable | R/W | 0 |
| ADMSK6 | 6 | ADC channel 6 enable | R/W | 0 |
| ADMSK7 | 7 | ADC channel 7 enable | R/W | 0 |
| ADMSK8 | 8 | ADC channel 8 enable | R/W | 0 |
| ADMSK9 | 9 | ADC temperature channel enable | R/W | 0 |
| - | 31:10 | Reserved | R | 0 |

**ADC0 - ADC dedicated analog input conversion results (channel 0)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Fields** | | **Description** | **Read/Write** | **Reset State** |
| **Name** | **Bit(s)** |
| DATA | 11:0 | ADC dedicated analog input conversion results (channel 0) | R | 0 |
| - | 31:12 | Reserved | R | 0 |

**ADC1 - ADC channel 1 conversion results**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Fields** | | **Description** | **Read/Write** | **Reset State** |
| **Name** | **Bit(s)** |
| DATA | 11:0 | ADC channel 1 conversion results | R | 0 |
| - | 31:12 | Reserved | R | 0 |

**ADC2 - ADC channel 2 conversion results**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Fields** | | **Description** | **Read/Write** | **Reset State** |
| **Name** | **Bit(s)** |
| DATA | 11:0 | ADC channel 2 conversion results | R | 0 |
| - | 31:12 | Reserved | R | 0 |

**ADC3 - ADC channel 3 conversion results**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Fields** | | **Description** | **Read/Write** | **Reset State** |
| **Name** | **Bit(s)** |
| DATA | 11:0 | ADC channel 3 conversion results | R | 0 |
| - | 31:12 | Reserved | R | 0 |

**ADC4 - ADC channel 4 conversion results**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Fields** | | **Description** | **Read/Write** | **Reset State** |
| **Name** | **Bit(s)** |
| DATA | 11:0 | ADC channel 4 conversion results | R | 0 |
| - | 31:12 | Reserved | R | 0 |

**ADC5 - ADC channel 5 conversion results**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Fields** | | **Description** | **Read/Write** | **Reset State** |
| **Name** | **Bit(s)** |
| DATA | 11:0 | ADC channel 5 conversion results | R | 0 |
| - | 31:12 | Reserved | R | 0 |

**ADC6 - ADC channel 6 conversion results**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Fields** | | **Description** | **Read/Write** | **Reset State** |
| **Name** | **Bit(s)** |
| DATA | 11:0 | ADC channel 6 conversion results | R | 0 |
| - | 31:12 | Reserved | R | 0 |

**ADC7 - ADC channel 7 conversion results**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Fields** | | **Description** | **Read/Write** | **Reset State** |
| **Name** | **Bit(s)** |
| DATA | 11:0 | ADC channel 7 conversion results | R | 0 |
| - | 31:12 | Reserved | R | 0 |

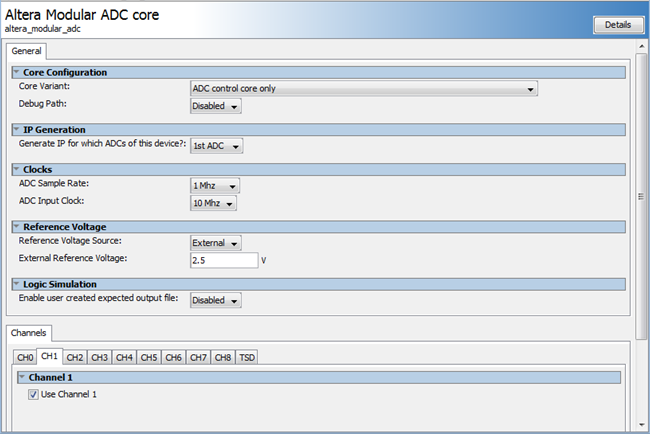
**ADC8 - ADC channel 8 conversion results**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Fields** | | **Description** | **Read/Write** | **Reset State** |
| **Name** | **Bit(s)** |
| DATA | 11:0 | ADC channel 8 conversion results | R | 0 |
| - | 31:12 | Reserved | R | 0 |

**ADCT - ADC temperature channel conversion results**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Fields** | | **Description** | **Read/Write** | **Reset State** |
| **Name** | **Bit(s)** |
| DATA | 11:0 | ADC temperature channel conversion results | R | 0 |
| - | 31:12 | Reserved | R | 0 |

**Altera Modular ADC core configuration to use with Interface Module (example)**



**Terasic DE10-Lite FPGA board specifics**

* The FPGA device (Altera MAX10 10M50DAF484C7G) that was used on DE10-Lite board have dual ADC core. But the second ADC core inputs are connected to ground;
* The operational amplifier that was used for input Low Pass filter (MCP6244-E/SL) Gain Bandwidth Product is 550 kHz

|  |  |
| --- | --- |
| **ADC input pins connection** | **ADC Input Low Pass Filter** |
| **D:\ZSL\electro\projects\FPGA\ahb_lite_adc_max10\doc\img\2017-06-12_15-18-58.png** | **D:\ZSL\electro\projects\FPGA\ahb_lite_adc_max10\doc\img\2017-06-12_15-19-42.png** |

**MIPSfpga+ ADC Example Program run order (on Windows)**

* change directory to MIPSfpga-plus folder;
* check settings in **mfp\_ahb\_lite\_matrix\_config.vh**. The **MFP\_USE\_ADC\_MAX10** option have to be uncommented;
* change directory to **programs\09\_adc**;
* set the correct running mode in **main.c** (SIMULATION or HARDWARE);
* for simulation run those scripts: **02\_compile\_and\_link.bat**, **05\_generate\_verilog\_readmemh\_file.bat**, **06\_simulate\_with\_modelsim.bat**;
* for hardware test you have to build the MIPSfpga system and load the program binary file to its memory;
* the ADC channel is selected by onboard switches position. The conversation result will be shown on HEX indicator.