



eSi-7510 Digital Down Converter

1 Contents

| | | |
|-----|---|---|
| 1 | Contents | 2 |
| 2 | Overview | 3 |
| 3 | Hardware Interface | 4 |
| 3.1 | Configuration | 5 |
| 4 | Software Interface | 6 |
| 4.1 | Register Map | 6 |
| 4.2 | Interrupts | 7 |
| 5 | Direct Digital Down Converter Operation | 8 |
| 5.1 | Introduction | 8 |
| 5.2 | Interfacing | 8 |
| 5.3 | Operation | 8 |
| 5.4 | References | 8 |

2 Overview

The eSi-7510 core provides the control and data plane interfaces to a Digital Down Converter. The signal processing in the core consists of a digital frequency synthesizer and a digital mixer. It supports the following features:

- Uses a 16x compressed table for the digital frequency synthesiser
- Mixes a real input signal down to a complex baseband signal
- Fine control over the frequency and phase
- Suitable for all digital receivers using a free running clock
- Suitable for multi-standard wireless receivers
- ASIC or FPGA target
- Fully synchronous design
- AXI4-Streaming inputs and outputs
- APB configuration
- Verilog 2001

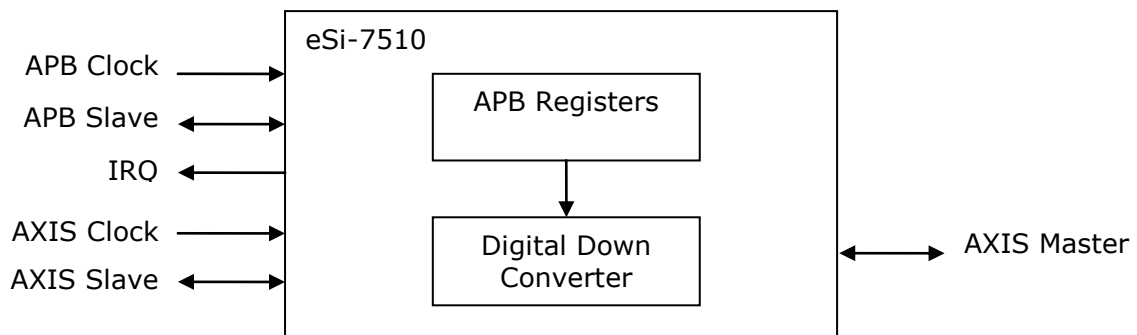


Figure 1: eSi-7510

The control logic handles the sequencing and control for handling multiple packets. This reduces the number of interrupts to the processor and significantly reduces the software overhead that would otherwise be spent on controlling the dataflow. The APB interface provides simple programmability to set frequency, phase and control the IP.

3 Hardware Interface

| | |
|---------------------|---|
| Module Name | esi_ddc |
| HDL | Verilog 2001 |
| Technology | Generic |
| Source Files | esi_ddc.v, ddc.v, ddc_apb.v, cexp.v, axis_adapter.v, eig_ocosdt_include.v, eig_osindt_include.v |

| Parameter | Range | Default | Description |
|-------------------|-------|---------|-----------------------------------|
| apb_address_width | 3-16 | 16 | APB address bus width |
| apb_data_width | 16/32 | 32 | APB data bus width (same as BITS) |
| data_width | 8-32 | 16 | AXI4-Stream data width |
| phase_width | 10-32 | 24 | phase/freq register width |
| table_width | 10-14 | 12 | Full cycle sine table width |
| id_width | 1-8 | 1 | AXI4-Stream ID width |

Table 1: Parameters

| APB Port | Direction | Width | Description |
|--------------|-----------|-------|-------------------------------|
| pclk | Input | 1 | Clock |
| presetn | Input | 1 | Reset, active-low |
| paddr | Input | 8 | Address |
| psel | Input | 1 | Slave select |
| penable | Input | 1 | Enable |
| pwrite | Input | 1 | Write |
| pwdata | Input | BITS | Write data |
| pclk_cactive | Output | 1 | Clock active |
| pready | Output | 1 | Ready |
| prdata | Output | BITS | Read data |
| pslverr | Output | 1 | Slave error |
| interrupt_n | Output | 1 | Interrupt request, active-low |

Table 2: APB I/O Ports

| AXI4S Port | Direction | Width | Description |
|--------------|-----------|--------------|-------------------|
| aclk | Input | 1 | Clock |
| aresetn | Input | 1 | Reset, active-low |
| s_tdata | Input | data_width | Slave data in |
| s_tid | Input | id_width | Slave id |
| s_tlast | Input | 1 | Slave last |
| s_tvalid | Input | 1 | Slave valid |
| s_tready | Output | 1 | Slave ready |
| m_tdata | Output | 2*data_width | Master data |
| m_tid | Output | id_width | Master id |
| m_tlast | Output | 1 | Master last |
| m_tvalid | Output | 1 | Master valid |
| m_tready | Input | 1 | Master ready |
| aclk_cactive | Output | 1 | Clock active |

Table 3: AXI4-Stream I/O Ports

For complete details of the APB and AXI4-Stream signals, please refer to the AMBA Protocol specifications available at <http://www.arm.com/products/solutions/AMBAHomePage.html>

The APB clock `pclk` and AXI clock `aclk` must be synchronous, but do not have to be the same frequency. This enables the APB interface to be run from a slower clock than the AXI interfaces.

The `aresetn` and `presetn` are used internally as active low synchronous resets. These reset signals may be asserted asynchronous to the clocks but deassertion must be synchronous after the rising edge of their respective clock.

The core may also be used without an APB interface by instantiating the file `ddc.v` as the top level and providing signals `phase` and `freq`.

The deliverable contains a Quartus project to build a Stratix IV version of the chosen core. Simply open up the Quartus project `esi_ddc.qpf` and compile to get the top level. The following table gives some typical results using balanced optimization and 200MHz clock specified in the Synopsys Design Constraints File.

| | ALUTs | Regs | M9K blocks | Mem (bits) | DSP 18-bit | F_{max} (MHz) |
|----------------|--------------|-------------|-------------------|-------------------|-------------------|------------------------------|
| table_width=12 | 213 | 243 | 1 | 5120 | 4 | 250 |

All F_{max} are worst case (i.e. for 85°C 1.2V slow corner). For higher speed optimization please contact EnSilica for custom effort, the current design is a trade off of speed and area.

3.1 Configuration

The default configuration uses an equivalent 4096 full cycle table of 10-bit amplitude resolution for the direct digital frequency synthesis. Various optimization techniques have been applied so the actual stored table is only 5-bit values covering 1/8 of a sine wave and 1/8 of a cosine wave. Using well known reconstruction techniques it delivers resolution bit-exact to a floating point sine wave rounded 10-bits and symmetrically saturated. Compared to storing a full cycle sine and cosine this method achieves a 16x data compression. A C++ program is provided to generate the tables for any phase and amplitude resolution. In general the optimum amplitude resolution is 1 or 2 bits less than the phase resolution. The table below shows the maximum spur level due to phase quantization and the noise floor due to amplitude quantization. The default 12-bit phase, 10-bit amplitude results in >60dB SNR.

The mixer result is rounded before being output.

| PARAMETER | Maximum spur (dB) | SNR (dB) |
|------------------|--------------------------|-----------------|
| phase_width=10 | -56.28 | |
| phase_width=11 | -62.30 | |
| phase_width=12 | -68.32 | |
| phase_width=13 | -74.34 | |
| phase_width=14 | -80.36 | |
| phase_width=15 | -86.38 | |
| phase_width=16 | -92.4 | |
| ampl_width=8 | | -49.92 |
| ampl_width=9 | | -55.94 |
| ampl_width=10 | | -61.96 |
| ampl_width=11 | | -67.98 |
| ampl_width=12 | | -74.00 |
| ampl_width=13 | | -80.02 |
| ampl_width=14 | | -86.04 |
| ampl_width=15 | | -92.06 |

4 Software Interface

4.1 Register Map

The software register map is given below

| Register | Address offset | Access | Description |
|----------|----------------|--------|--------------------|
| control | 0x00 | R/W | Control register |
| status | 0x04 | R/W | Status register |
| freq | 0x08 | R/W | Frequency register |
| phase | 0x10 | R/W | Phase register |

Table 4: Register Map

4.1.1 Control Register

The control register contains a selection of flags that control the operation of the module.

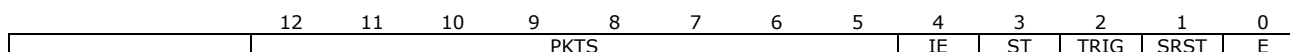


Figure 2: Format of the control register

| Register | Values | Description |
|----------|---|--|
| E | 0 - Disabled 1 - Enabled | Enables the peripheral. When disabled data will not be processed. |
| SRST | 0 - Normal operation 1 - Synchronous reset | Synchronous reset of the peripheral and AXI4 streaming interface |
| TRIG | 1 - Trigger | Write 1 to trigger processing of PKTS packets. This field is self clearing. Execution stops after PKTS packets are processed. |
| ST | 0 - Packet based 1 - Continuous stream | Writing 0 to this register allow packet based control over the peripheral using the TRIG and PKTS field. Writing 1 causes the peripheral to continuously process packets as soon as they become available. |
| IE | 0 - Disable interrupts 1 - Enable interrupts | Writing 1 will generate an interrupt on interrupt_n once PKTS packets have been processed. |
| PKTS | 1-255 | Number of packets to process before stopping |

Table 5: Fields of the control register

4.1.2 Status Register

The status register contains the interrupt bit. To clear a bit in the status register, write a 1 to it. Writing a 0 will leave it unchanged.

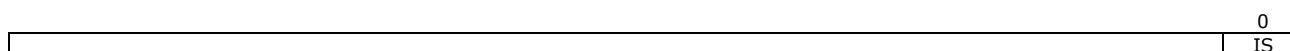


Figure 3: Format of the status register

| Register | Values | Description |
|----------|--|---|
| IS | 0 - Interrupt not set 1 - Interrupt set | Interrupt status. This bit is set to 1 once PKTS packets have been processed. It is set independent of the IE field to allow software polling instead of hardware interrupt generation. |

Table 6: Fields of the status register

4.1.3 Freq Register

The `freq` register holds the signed frequency. The mixing frequency is given by

$$f = \frac{FREQ}{2^{phase_width}} f_s$$

where f_s is the sampling frequency of the input data.



Figure 4: Format of the `freq` register when `phase_width=24`

4.1.4 Phase Register

The `phase` register holds the signed phase offset. The phase offset is given by

$$\phi = \frac{PHASE}{2^{phase_width}} 2\pi$$



Figure 5: Format of the `phase` register when `phase_width=24`

4.2 Interrupts

The `interrupt_n` signal will be raised when `PKTS` packets have been processed and the `IE` flag in the control register is set to 1. The interrupt status bit `IS` is set when `PKTS` packets have been processed. The `IS` bit is set independent of the `IE` flag setting. The `IS` bit and `interrupt_n` signal are both cleared by writing a 1 to the status register. The receive interrupt operation and `IS` flag are also active in streaming mode `ST=1`, but are not intended to be used in streaming mode.

5 Digital Down Converter Operation

5.1 Introduction

This mixes a real input signal to a complex baseband signal with fine control over the frequency and phase.

5.2 Interfacing

The slave AXI4-Stream input interface and master AXI4-Stream output interface obey a handshaking protocol using the `x_tvalid` and `x_tready` signals. The following description is taken from the AXI4-Stream specification v1.0. The `x_tvalid` and `x_tready` handshake determines when information is passed across the interface. A two-way flow control mechanism enables both the master and slave to control the rate at which the data and control information is transmitted across the interface. For a transfer to occur both the `x_tvalid` and `x_tready` signals must be asserted. Either `x_tvalid` or `x_tready` can be asserted first or both can be asserted in the same `aclk` cycle. A master is not permitted to wait until `m_tready` is asserted before asserting `m_tvalid`. Once `x_tvalid` is asserted it must remain asserted until the handshake occurs. A slave is permitted to wait for `s_tvalid` to be asserted before asserting the corresponding `s_tready`. If a slave asserts `s_tready`, it is permitted to deassert `s_tready` before `s_tvalid` is asserted.

The final sample of a packet must have `s_tlast` asserted on the slave handshake. A packet can have an ID associated with it though the AXI ID signal. This allows the system to pass on sidechannel information with each sample.

5.3 Operation

After a reset the core generates the following output

$$y(n) = x(n) \exp(j(\theta(n) + 2\pi f(n-1)/f_s) + j\phi(n))$$

$$f(-1) = 0, \quad \phi(0) = 0, \quad \theta(0) = 0$$

So that the current phase is added to the previous integrated frequency. This allow the core to generate the natural output

$$y(n) = x(n) \exp(j(2\pi n f / f_s + \phi)), \quad n = 0, 1, \dots$$

5.4 References

- [1] Direct digital synthesis: a tool for periodic wave generation (part 1), IEEE signal processing magazine, Lionel Cordesses, July 2004, p50-54.
- [2] Direct digital synthesis: a tool for periodic wave generation (part 2), IEEE signal processing magazine, Lionel Cordesses, September 2004, p110-117.
- [3] A low power, low voltage direct digital frequency synthesizer, S. Liao, L.G.Chen.