Key Design Features

- Synthesizable, technology independent VHDL Core

- Compensates and corrects the non-linear characteristic of an RF Power Amplifier operating at high power levels

- Corrects both the PA Gain and Phase responses - sometimes referred to as the (AM-AM) and (AM-PM) characteristics

- Features a 256 x 32-bit LUT RAM to store the complex coefficients

- LUT may be initialized at compile time or modified during normal circuit operation

- Optional UART, I2C or SPI interface to allow easy programming via a microcontroller

- Suitable for open-loop or closed-loop operation

- LUT coefficients may be programmed 'on-the-fly' for adaptive precorrection systems

- Typically recovers up to 20dB in the output spectral 'shoulders' after the RF Power Amplifier stages[1]

- Suitable for operation at baseband or IF frequencies

- Pipeline latency of only 8 clock cycles

- Sample rates in excess of 200 MHz[2]

Applications

- Power amplifier linearization for mobile Base-stations, Broadcasting etc.

- Precorrection of wide bandwidth signals such as UMTS, WCDMA and OFDM

- Precorrection of any type of digitally modulated signal where the signal envelope varies and therefore the instantaneous input power
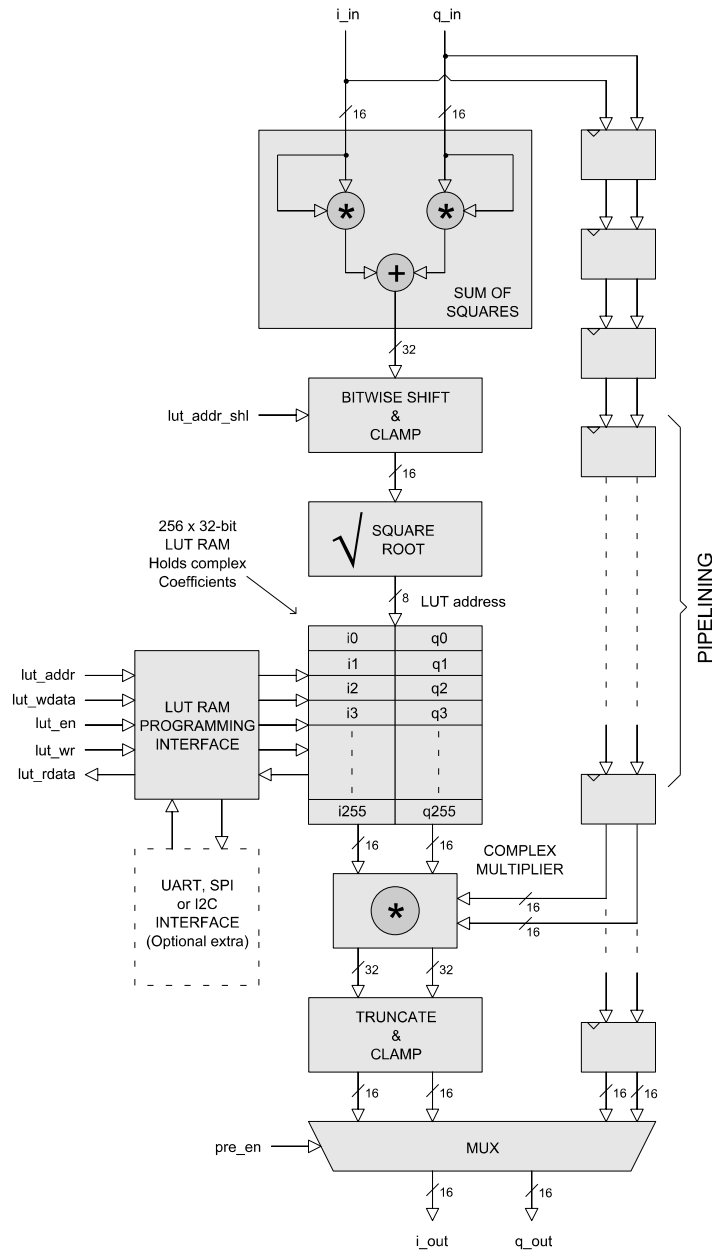
Block Diagram

Figure 1: Precorrection system architecture

1 Signal improvement will generally depend on how hard the PA is being driven
2 Xilinx Virtex 5 FPGA used as a benchmark

Generic Parameters

Generic name	Description	Type	Valid range
lut_addr_shl	Selects a 16-bit slice of the 32-bit result of the sum of squares (Resultant 16-bit slice used as an input value to square root)	integer	0 to 15 0 = select LSBs [15:0] 15 = select MSBs [30:15]

Pin-out Description

Pin name	I/O	Description	Active state
clk	in	Sample clock	rising edge
reset	in	Asynchronous reset	low
en	in	clock-enable	high
pre_en	in	Precorrection enable/disable	high = enable low = disable
lut_addr [7:0]	in	Complex LUT read/write address	address
lut_wdata [31:0]	in	Complex LUT write data I = [31:16], Q = [15:0]	data
lut_rdata [31:0]	out	Complex LUT read data I = [31:16], Q = [15:0]	data
lut_en	in	Complex LUT enable	high
lut_wr	in	Complex LUT write enable	high
i_in [15:0]	in	In-phase complex input	data
q_in [15:0]	in	Quadrature complex input	data
i_out [15:0]	out	In-phase complex output	data
q_out [15:0]	out	Quadrature complex output	data

General Description

DPSYS is a complete Digital Precorrection (Predistortion) system designed to compensate for the non-linear characteristic of a high-power RF Amplifier. The system is capable of adjusting both the gain and phase of a complex input signal. This is achieved by means of a complex multiplication of the input with a complex polynomial function stored in the LUT. Complex inputs are sampled on the rising edge of *clk* when *en* is high.

The LUT contains the inverse PA characteristic and is applied before the amplification stages (either at baseband or IF frequencies). By programming the LUT with the inverse gain/phase PA response, the resultant PA response is linearized. After linearization, the output signal is much cleaner with reduced intermodulation distortion. This is most readily observed by the absence of raised 'shoulders' at either side of the signal of interest.

The system may be used in open-loop or closed-loop configuration. For open loop operation, the LUT coefficients are static and programmed during initial setup of the PA precorrection system. For closed-loop operation, an external circuit may compare the baseband inputs and PA outputs and adjust the LUT coefficients dynamically in order to automate the linearization process.

Figure 1 shows the stages involved in the precorrection pipeline. The first stage of the pipeline computes the magnitude of the complex input signal. The magnitude serves as an address into a LUT which in turn indexes a pair of complex coefficients (I & Q). The complex input is multiplied by the complex coefficients in order to modify the gain and phase of the input accordingly. Finally the output is truncated and clamped to 16-bit precision.

Input Magnitude calculation

The PA gain and phase responses are a function of input signal power. Figure 2 shows a simplified PA characteristic of input power vs. gain and input power vs. phase. These plots are commonly known as the AM-AM and the AM-PM characteristics of the amplifier.
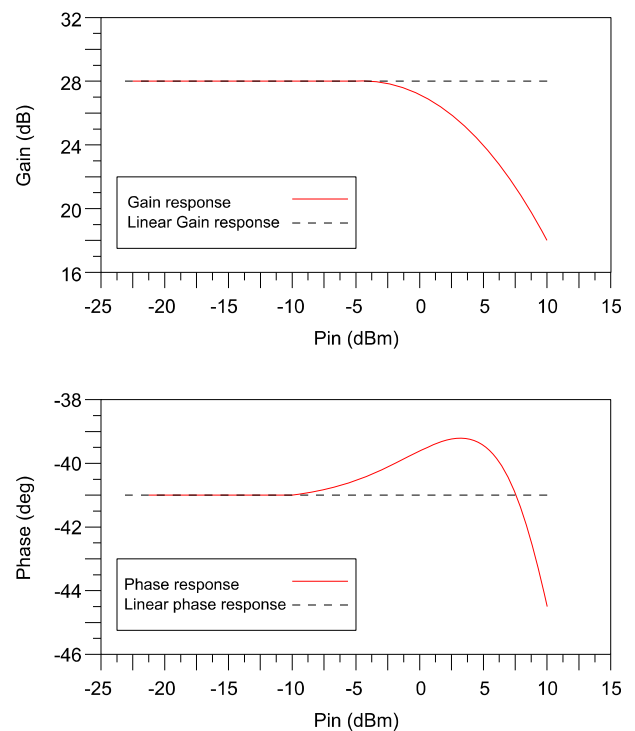

Figure 2: Simplified AM-AM and AM-PM responses for an RF Power Amplifier

In order to define an inverse characteristic to those shown in Figure 2, the complex input magnitude must first be calculated. This gives a relative measure of the input power and therefore a relative position on the x-axis for the AM-AM and AM-PM curves. The magnitude of the complex input is calculated by the formula:

$$Mag = \sqrt{I_{in}^2 + Q_{in}^2} \quad \text{... Equ (1)}$$

This operation is performed in two stages. The first stage computes the sum of squares (SOS) of the complex input generating a 32-bit unsigned result. A 16-bit slice of this result is then taken as an input into the square-root function. The slice-bits are selectable using the generic parameter *lut_addr_shl.* which may be adjusted (Figure 3) to ensure that the full dynamic range of the LUT is utilized.

As an example, setting *lut_addr_shl* to 13 will select bits [28:13] of the SOS result. Note that the sign-bit (bit-31) is redundant due to the previous squaring operation. Normally, the best selection of slice-bits are the most significant 16-bits of the SOS. However, for optimum results, the slice bits should be determined by simulation and/or experiment.
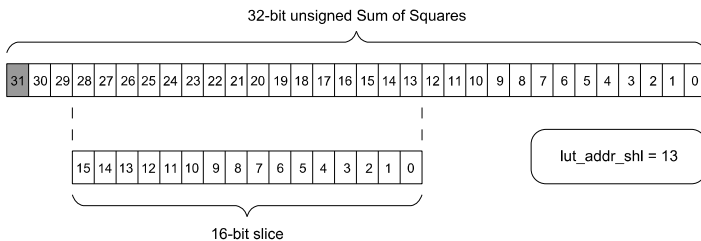

Figure 3: Sum of Squares bit-selection example

The output of the square-root function gives an 8-bit magnitude which is used as an address into the LUT.

Complex-valued Look-up Table

The LUT is a 256 x 32-bit dual-port RAM that contains the complex coefficients used to adjust the Gain and Phase of the input signal. The I coefficient is stored in bits [31:16] and the Q coefficient is stored in bits [15:0] for each location in the RAM.

The values in the LUT should be programmed with the inverse AM-AM and AM-PM curves in order to correct the PA response. Figure 4 demonstrates how this is done graphically.
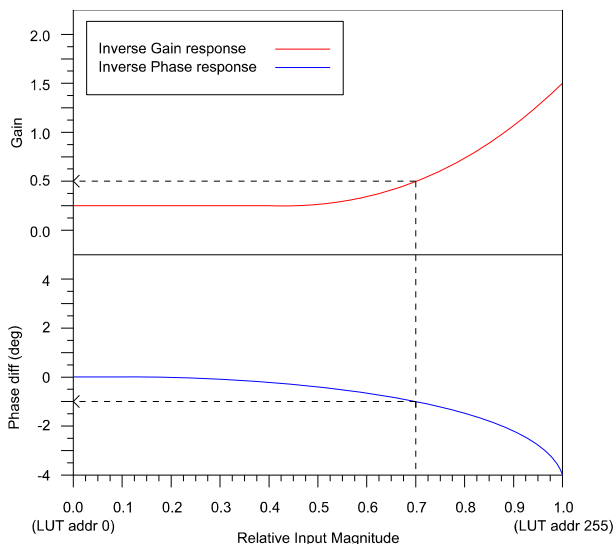

Figure 4: Inverse AM-AM and AM-PM curves

The relative input magnitude is defined in the range 0 to 1, with a gain of 0 representing location 0 in the LUT, a gain of 1/256 representing location 1 etc … up to a gain of 255/256 at location 255.

Once the inverse gain and phase curves have been defined, the user can read off the the gain and phase values for each address on the x-axis. The complex LUT coefficients can then be found using the formula:

$$I_n = Gain * [\,\cos\,(Phase)\,]$$
$$Q_n = Gain * [\,\sin\,(Phase)\,]$$

… Equ (2)

Gain is a multiplication factor between 0 and 2. Phase is measured in radians in the range -Pi/4 to Pi/4. I and Q are stored as signed [16 14] values with 1 sign bit, 1 integer bit and 14 fraction bits. As an example, consider the calculation of I_n and Q_n at the input magnitude of 0.7 as shown in Figure 4. Reading off the values we have a Gain of 0.5 and a Phase difference of -1 degrees or -0.01745 radians. The complex LUT values would be calculated as:

$$I_{179} = 0.5 * \cos\,(-0.01745) = 0.49992$$
$$Q_{179} = 0.5 * \sin\,(-0.01745) = -0.00873$$

These values would be placed in the LUT at location 179 (0.7 * 255). Converting to signed [16 14] format, these results in hex would be 0x1FFF and 0xFF71 respectively.

Programming of the LUT RAM

The LUT is programmable via a simple interface. Writes to the LUT RAM occur on the rising edge of *clk* when the signal *lut_en* and *lut_wr* are both high. A read from the RAM occurs on a rising clock edge when *lut_en* is high and *lut_wr* is low. The latency of a RAM read is 3 clock cycles. Note that the RAM interface works independently to the rest of the circuit and is not controlled by the clock-enable signal.

The LUT may also be programmed via a number of alternative serial interfaces. These interfaces are UART, I2C and SPI. Most microprocessors and microcontrollers come with at least one of these serial interfaces offering convenient programming. For high-bandwidth closed-loop operation, SPI is recommended as it supports the highest data rates[3]

Complex Multiplier

The complex multiplier performs the complex multiplication of the LUT values with the complex input signal. The result is a complex output that has been adjusted in gain and phase. The complex multiply is of the form:

$$I_{out} = (\,I_{in} * I_n\,) - (\,Q_{in} * Q_n\,)$$
$$Q_{out} = (\,I_{in} * Q_n\,) + (\,Q_{in} * I_n\,)$$

Where I_{in} and Q_{in} are the complex inputs, I_n and Q_n are the complex LUT values and I_{out}, Q_{out} are the complex outputs.

---

3   Serial interfaces are provided as an optional extra. SPI may be clocked up to 40MHz.

Output Truncation, Clamp and bypass Mux

After the complex multiply, the results are truncated and clamped to 16-bits resolution. Finally, the design offers a precorrection enable/disable feature. On setting the input *pre_en* to 0 the precorrection pipeline is disabled. The latency of 8 clock cycles is maintained whether precorrection is enabled or disabled.

Functional Timing

Figure 5 gives an example of the LUT programming. A write of the value 0x2DE7FF74 to address 0x9B is shown followed by a read from the same address. Notice that a LUT read has a latency of 3 clock cycles. A write occurs on a rising clock edge when *lut_en* and *lut_wr* are both high. A read occurs on a rising clock edge when *lut_en* is high and *lut_wr* is low.
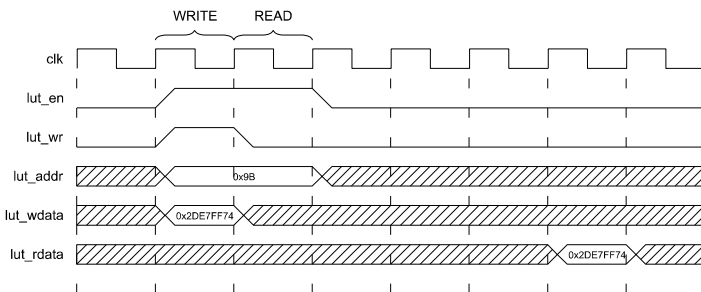

Figure 5: LUT programming timing waveforms

During normal operation the precorrection circuit samples the complex inputs i_in and q_in on the rising edge of *clk* when *en* is high. If *en* is low then the precorrection pipeline is stalled.

Figure 6 demonstrates the precorrection pipeline operating normally. As an example, the complex input at 'A' is sampled and the complex output is available 8 cycles later at 'B' - i.e. the pipeline has a latency of 8 clock cycles.
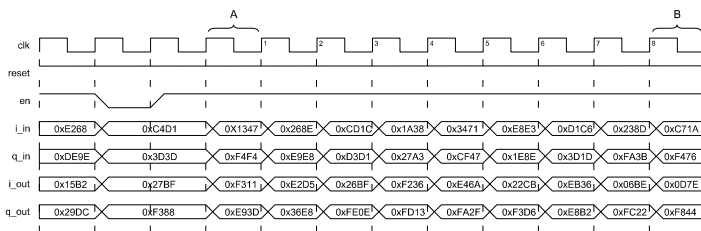

Figure 6: Precorrection timing waveforms

Source File Description

All source files are provided as text files coded in VHDL. The following table gives a brief description of each file.

Source file	Description
dpsys_lut.txt	Text file containing LUT coefficients - read in during simulation
dpsys_pack.vhd	Package containing default LUT coefficients
dpsys_sqrt_lut.vhd	Square-root function
dpsys_ram_256_32.vhd	Precorrection LUT RAM stores complex coefficients
dpsys_lutfile_reader.vhd	File reader component reads LUT coefficients during simulation
dpsys.vhd	Top-level component
dpsys_bench.vhd	Top-level testbench

Functional Testing

An example VHDL testbench is provided for use in a suitable VHDL simulator. The compilation order of the source code is as follows:

1. dpsys_pack.vhd
2. dpsys_sqrt_lut.vhd
3. dpsys_ram_256_32.vhd
4. dpsys.vhd
5. dpsys_lutfile_reader.vhd
6. dpsys_bench.vhd

The VHDL testbench instantiates the DPSYS component. The default LUT coefficients may be defined in the package 'dpsys_pack.vhd' or alternatively, defined in a text file which gets read at the start of simulation. The text file is called 'dpsys_lut.txt' and should be placed in the top-level simulation directory. The format of the file is as follows:

01 2CCCFFF8 1 1   # Write to address 0x01
02 000000000 1 0   # Read from address 0x2

etc.

The first two hex characters are the address, the next 8 are the 32-bit write data and the final two fields are the LUT enable and LUT write strobes. The test bench provided is programmed with example LUT coefficients to modify the gain and phase of the complex inputs. The complex inputs are generated randomly.

The complex inputs and outputs are captured during the course of the simulation to the files 'dpsys_in.txt' and 'dpsys_out.txt'. These files may be used to verify the gain and phase response of the Digital Precorrection System. Results of input magnitude vs. gain are shown in Figure 7.
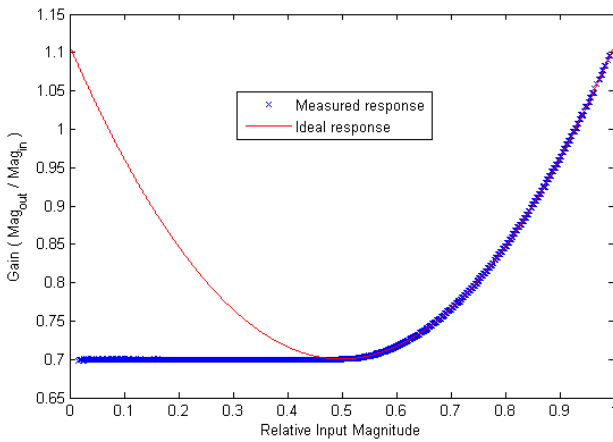
Figure 7: Relative input Magnitude vs. Gain for the testbench example

The red curve in Figure 7 (from Magnitude 0.5 onwards) shows the original polynomial used to model the gain response. Notice the exact correlation between this curve and the measured response.
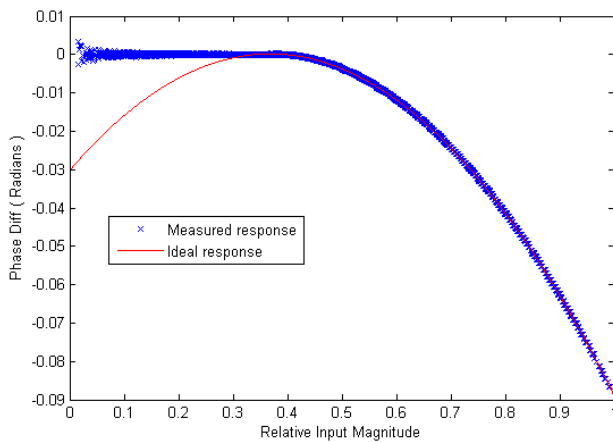

Figure 8: Relative input Magnitude vs. Phase difference for the testbench example

Figure 8 plots input magnitude vs. phase *difference*. Again, the original polynomial is shown against the measured response and again the plot shows very good correlation (the polynomial is valid from an input magnitude of 0.4 onwards). Notice that at very small input magnitudes, the square-root calculation is less accurate - resulting in a narrow zone of spreading close to 0.

Performance

The RF Power Amplifier Precorrection System was tested on a Xilinx® Virtex-II FPGA development platform clocking at 65MHz. The circuit was implemented at baseband frequencies with an 8MHz bandwidth OFDM source. The output of the precorrection circuit was up-converted so that the central carrier of the OFDM signal was positioned at 15MHz. Complex output samples were taken and passed through a high-power PA model. The PA model was configured to exhibit different degrees of non-linearity in both gain and phase[4]. Figure 9 shows a block diagram of the test arrangement.
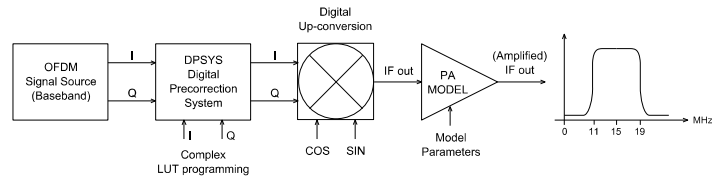

Figure 9: H/W test setup for Precorrection System

As a control, the output of an 'Ideal' PA was measured with the PA exhibiting a completely linear response. This was compared against the PA model with the highest degree of non-linearity. Figure 10 shows the resulting mean-square spectrum of the two control situations. The degraded output signal is shown in red and the characteristic raised 'shoulders' can clearly be observed. The signal in blue demonstrates the Ideal PA output.
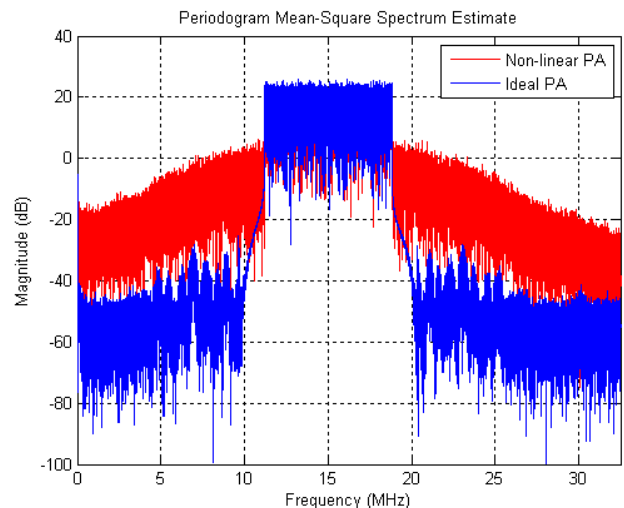

Figure 10: OFDM output spectra for the Ideal and non-linear PA cases

---

4   A 2nd-order polynomial was used to model the PA response with the non-linearity starting a different input magnitudes.

The next stage of testing was to use varying degrees of non-linearity in the PA with the precorrection system enabled. The LUT coefficients were adjusted in order to realize the inverse gain and phase characteristics for the particular PA model. The results were plotted as a Welch mean-square spectrum to average out the peaks. The precorrected case vs. the most non-linear PA model (model #3) represented an improvement of 20 dBs at the spectrum shoulders.
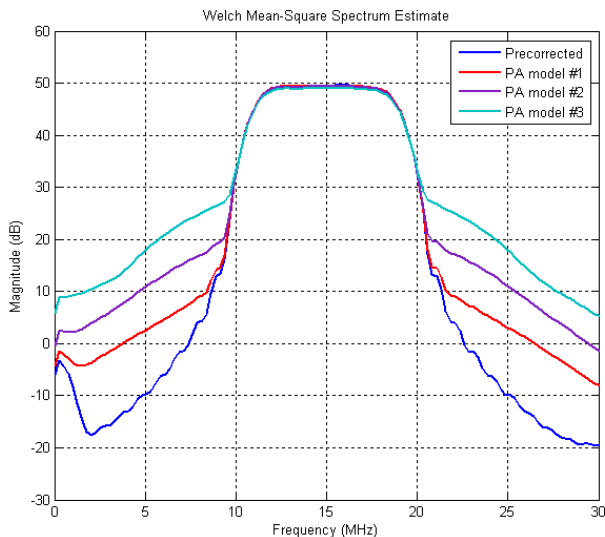

Figure 11: Precorrected vs. non-corrected spectra for different PA models

Synthesis

The files required for synthesis and the design hierarchy is shown below:

- dpsys_pack.vhd
- dpsys.vhd
  - dpsys_sqrt_lut.vhd
  - dpsys_ram_256_32.vhd

The VHDL core is designed to be technology independent. However, as a benchmark, synthesis results have been provided for the Xilinx Virtex 5 and the Altera Stratix III series of FPGA devices. The lowest and highest speed grade devices have been chosen in both cases for comparison.

Note that the square-root function is implemented as a ROM and uses a significant amount of Block RAM resource on the device. A pipelined implementation of the square-root function is available on request. In some synthesis tools and technologies, the default RAM values in dpsys_pack.vhd may not be supported.

Trial synthesis results are shown with the generic parameter *lut_addr_shl* = 15. Resource usage is specified after Place and Route.

VIRTEX 5

Resource type	Quantity used
Slice register	219
Slice LUT	435
Block RAM	16
DSP48	6
Clock frequency (worst case)	146 MHz
Clock frequency (best case)	212 MHz

STRATIX III

Resource type	Quantity used
Register	8492
ALUT	5835
Block Memory bit	524448
DSP block 18	6
Clock frequency (worse case)	166 MHz
Clock frequency (best case)	216 MHz

Revision History

Revision	Change description	Date
1.0	Initial revision	22/04/2009
1.1	Updated synthesis results in line with minor code changes	18/01/2012