COS_X

16-bit Cosine Function
Rev. 1.2

Key Design Features

- Synthesizable, technology independent VHDL Core
- Function y = cos(x)
- Input range 0 ≤ x ≤ π /2 (Quarter wave)
- Output range 0 ≤ y ≤ 1
- Based on a quadratic polynomial with dynamic coefficients
- Input values as 16-bit unsigned fractions
- Output values as 16-bit unsigned fractions in radians
- Accurate to within 0.0002
- High-speed fully pipelined architecture
- 3 clock-cycle latency

Applications

- Fixed-point mathematics
- Quadrature signal generation in digital communications
- Alternative to using a 65536 x 16-bit LUT (128kbytes)

Pin-out Description

Pin name	*I/O*	*Description*	*Active state*
clk	in	Synchronous clock	rising edge
en	in	Clock enable	high
x_in [15:0]	in	Input value in radians	data
y_out [15:0]	out	Output value	data

Functional Specification

Value	*Type*	*Valid range*
x_in [15:0]	16-bit unsigned fraction in [16 15] format	[0, π/2]
y_out [15:0]	16-bit unsigned fraction in [16 15] format	[0, 1] Accuracy to within 0.0002

Block Diagram

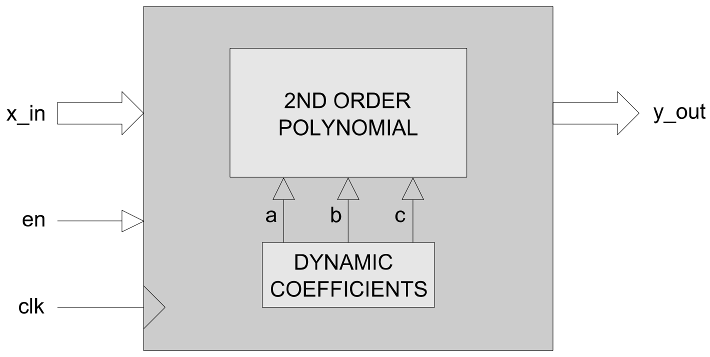

Figure 1: cos_x function architecture

General Description

COS_X (Figure 1) calculates the cosine of an angle in radians. It has a fully pipelined architecture and uses fixed-point mathematics throughout. Input values are accepted as 16-bit unsigned values in the range 0 to π/2. Output values are 16-bit unsigned values in the range 0 to 1. For input values greater than π/2, the output clips to 0. Both input and output values are in [16 15] format with 1 integer bit and 15 fraction bits. As an example the input value 0xC000 would signify the value 1.5.

Internally, the function uses a 2nd order polynomial of the form:

$$y=ax^2+bx+c$$

The coefficients a, b and c dynamically change with respect to the input value in order to generate a more accurate approximation. The output result is accurate to within 0.0002.

Values are sampled on the rising clock-edge of *clk* when *en* is high. The function has a 3 clock-cycle latency.

Functional Timing

Figure 2 demonstrates the computation of y = cos(x), where x = 0x37E4 (0.4366 as a decimal fraction). The result, 0x73FD (0.9062 in decimal) has a latency of 3 clock cycles.

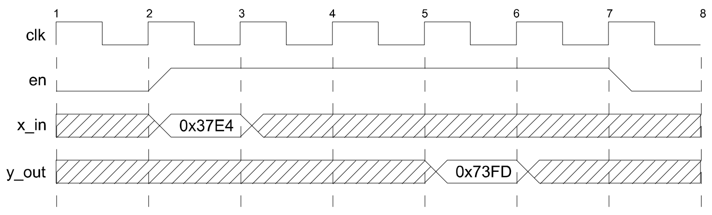

Figure 2: Calculation of y = cos(x)

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***
cos_x.vhd	Top-level block
cos_x_bench.vhd	Top-level test bench

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. cos_x.vhd
2. cos_x_bench.vhd

The simulation must be run for at least 1 ms during which time a 16-bit input stimulus in the range 0 to 65535 will be generated. The test terminates automatically.

The simulation generates a text file called *cos_x_out.txt.* This file contains the output results captured during the test. The results of the test are shown graphically in Figure 3 below:

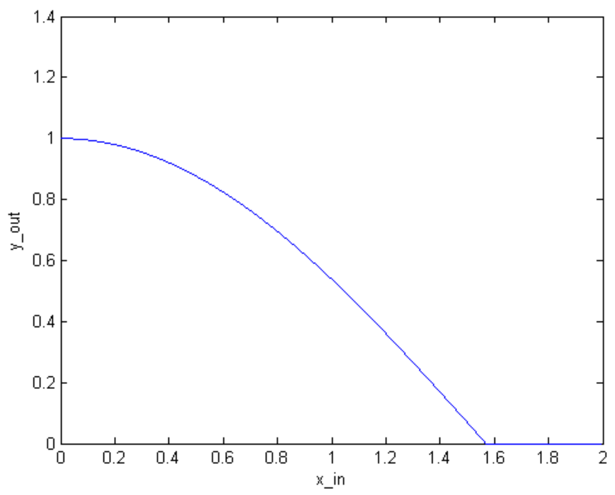

Figure 3: Plot of test results for cos_x function

Synthesis

The source file 'cos_x.vhd' is the only file required for synthesis. There are no sub-modules in the design.

The VHDL core is designed to be technology independent. However, as a benchmark, synthesis results have been provided for the Xilinx® Virtex 6 and Spartan 6 FPGA devices. Synthesis results for other FPGAs and technologies can be provided on request.

Resource usage is specified after Place and Route.

VIRTEX 6

Resource type	***Quantity used***
Slice register	40
Slice LUT	52
Block RAM	0
DSP48	3
Occupied slices	20
Clock frequency (approx)	300 MHz

SPARTAN 6

Resource type	***Quantity used***
Slice register	40
Slice LUT	52
Block RAM	0
DSP48	3
Occupied slices	21
Clock frequency (approx)	200 MHz

Revision History

Revision	***Change description***	***Date***
1.0	Initial revision	28/04/2008
1.1	Improved accuracy and updated synthesis results	07/04/2009
1.2	Updated synthesis results for Xilinx® 6 series FPGAs	06/06/2012