

### Key Design Features

- Synthesizable, technology independent VHDL Core
- Combined functions  $y = \sin(x)$ ,  $y = \cos(x)$
- Input range  $-\pi \leq x \leq \pi$
- Output range  $-1 \leq y \leq 1$
- Based on a quadratic polynomial with dynamic coefficients
- Input values as 18-bit signed fractions in radians
- Output values as 17-bit signed fractions
- Accurate to within 0.00017
- High-speed fully pipelined architecture
- Tiny implementation
- Only 5 clock-cycles latency

### Applications

- Fixed-point mathematics
- Quadrature signal generation in digital communications
- Cheaper alternative to using a 65k x 16-bit LUT (128kbytes)
- Cheaper alternative to CORDIC

### Pin-out Description

Pin name	I/O	Description	Active state
clk	in	Synchronous clock	rising edge
en	in	Clock enable	high
x_in [17:0]	in	Input value in radians	data
sin_out [16:0]	out	Output value	data
cos_out [16:0]	out	Output value	data

### Functional Specification

Value	Type	Valid range
x_in [17:0]	18-bit signed fraction in [18 15] format	$[-\pi, \pi]$
sin_out [16:0]	17-bit signed fraction in [17 15] format	$[-1, 1]$ Accuracy to within 0.0002
cos_out [16:0]	17-bit signed fraction in [17 15] format	$[0, 1]$ Accuracy to within 0.0002

### Block Diagram

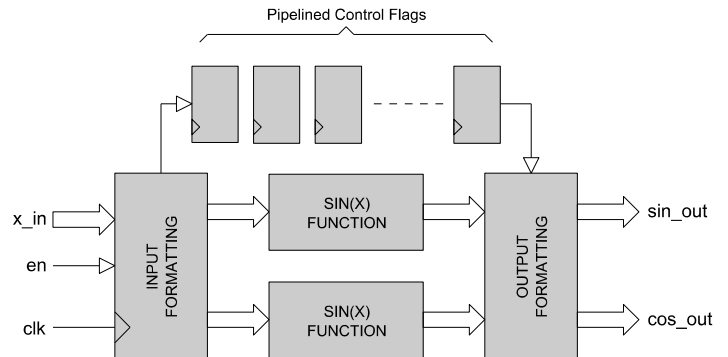


Figure 1: sincos\_x function architecture

### General Description

SINCOS\_X (Figure 1) calculates the sine and cosine of an angle in radians. It has a fully pipelined architecture and uses fixed-point mathematics throughout. Input values are accepted as 18-bit signed values in the range  $-\pi$  to  $\pi$ . Output values are 17-bit signed values in the range  $-1$  to  $1$ . For input values outside the specified range, *sin\_out* defaults to 0 and *cos\_out* defaults to  $-1$ .

Input and output values are specified in [18 15] format with 1 sign bit, 2 integer bits and 15 fraction bits. Output values are in [17 15] format with 1 sign bit, 1 integer bit and 15 fraction bits. Internally, the function uses a quarter-wave SIN function core implemented as a 2<sup>nd</sup> order polynomial with dynamic coefficients. Values are sampled on the rising clock-edge of *clk* when *en* is high. The function has a 5 clock-cycle latency.

### Functional Timing

Figure 2 demonstrates the computation of  $y = \sin(x)$  and  $y = \cos(x)$  where  $x$  is respectively 0x04872 and 0x30066 (0.56598 and -1.99689 as a decimal fraction). The results for the SIN calculation are 0x044A4 and 0x18B72 (0.53625 and -0.91058 in decimal). For the COS calculation the results are 0x06C09 and 0x1CB1B (0.84402 and -0.41324 in decimal).

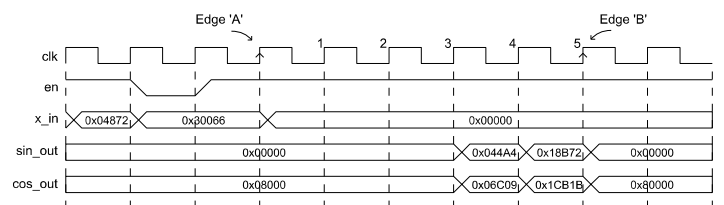


Figure 2: Timing waveform for the sincos function

Note that the second value of  $x$  is stalled by the clock-enable signal being de-asserted for one clock cycle. In the next cycle, the clock-enable is asserted high and normal operation continues. The function has a latency of 5 clock cycles as indicated by edge 'A' and 'B' in the timing waveform.

## 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
sin_x.vhd	Quarter-wave SIN function
sincos_x.vhd	Top-level block
sincos_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. sin\_x.vhd
2. sincos\_x.vhd
3. sincos\_x\_bench.vhd

The simulation must be run for at least 3 ms during which time an 18-bit input stimulus in the range 0 to 262143 will be generated. The test terminates automatically.

The simulation generates two text files called *sincos\_x\_in.txt* and *sincos\_x\_out.txt*. These files contain the input and output samples captured during the course of the test. The results of the test are shown graphically in Figure 3 below:

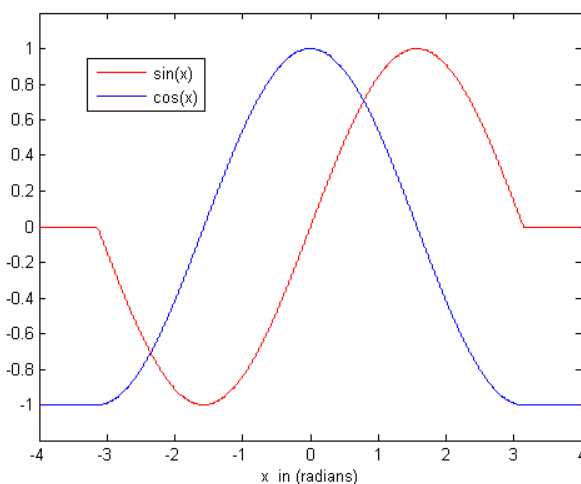


Figure 3: Plot of test results for sincos\_x function

Note the valid input range from  $-\pi$  to  $\pi$ . Values outside this range default to 0 for  $\sin(x)$  and -1 for  $\cos(x)$ .

## Synthesis

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

- sincos\_x.vhd
  - sin\_x.vhd

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	153
Slice LUT	328
Block RAM	0
DSP48	6
Occupied slices	101
Clock frequency (approx)	200 MHz

### SPARTAN 6

Resource type	Quantity used
Slice register	138
Slice LUT	332
Block RAM	0
DSP48	6
Occupied slices	112
Clock frequency (approx)	150 MHz

## Revision History

Revision	Change description	Date
1.0	Initial revision	04/05/2009
1.1	Updated synthesis results for Xilinx® 6 series FPGAs	07/06/2012

## X-ON Electronics

Largest Supplier of Electrical and Electronic Components

*Click to view similar products for [Development Software](#) category:*

*Click to view products by [Zipcores](#) manufacturer:*

Other Similar products are found below :

[SRP004001-01](#) [SW163052](#) [SYSWINEV21](#) [WS01NCTF1E](#) [W128E13](#) [SW89CN0-ZCC](#) [IP-UART-16550](#) [MPROG-PRO535E](#) [AFLCF-08-LX-CE060-R21](#) [WS02-CFSC1-EV3-UP](#) [SYSMAC-STUDIO-EIPCPLR](#) [1120270005](#) [SW006021-2H](#) [ATATMELSTUDIO](#) [2400573](#) [2702579](#) [2988609](#) [SW006022-DGL](#) [2400303](#) [88970111](#) [DG-ACC-NET-CD](#) [55195101-101](#) [55195101-102](#) [SW1A-W1C](#) [MDK-ARM](#) [SW006021-2NH](#) [B10443](#) [SW006021-1H](#) [SW006021-2](#) [SW006022-2](#) [SW006023-2](#) [SW007023](#) [MIKROE-730](#) [MIKROE-2401](#) [MIKROE-499](#) [MIKROE-722](#) [MIKROE-724](#) [MIKROE-726](#) [MIKROE-728](#) [MIKROE-732](#) [MIKROE-734](#) [MIKROE-736](#) [MIKROE-738](#) [MIKROE-744](#) [MIKROE-928](#) [MIKROE-936](#) [1120270002](#) [1120270003](#) [1120275015](#) [NT-ZJCAT1-EV4](#)