

CAST

C387I

Math Coprocessor Core

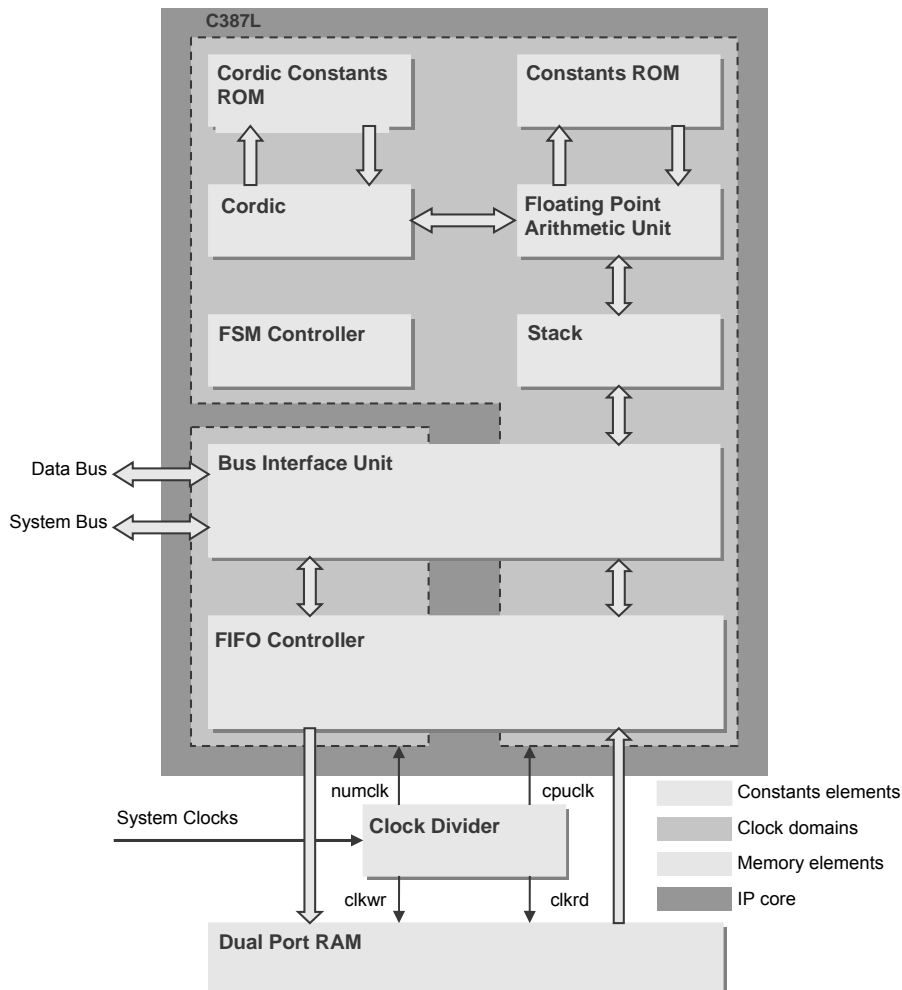
The C387L implements a math coprocessor and is derived from the Intel® i387SX. The C387L extends the architecture of the Intel® 386 processor with floating-point, extended integer and BCD data types.

A computing system that includes the C387L fully conforms to the IEEE 754-1985 Floating-Point Standard. The C387L adds over 70 mnemonics to the instruction set of the Intel® 386, including support for arithmetic, logarithmic, exponential, and trigonometric mathematical operations. The C387L are upward object-code compatible from the 8087-math coprocessor and will execute code written for the i387DX and i387SX math coprocessors.

Typically the core is delivered as VHDL source code for ASIC implementations. The following options may be ordered according to user's requirements:

- EDIF netlist for FPGA
- One-year maintenance
- On-site support

Block Diagram



Features

- High performance 80-bit internal architecture
- Implements ANSI/IEEE Standard 754-1985 for binary floating point arithmetic
- Fully compatible instruction set of i387DX and i387SX math coprocessors
- Implemented all i387SX architectural enhancements over 8087
- Full range transcendental operations for SINE, COSINE, TANGENT, ARCTANGENT and LOGARITHM
- Directly extends Intel® 386's instruction set to trigonometric, logarithmic, exponential, and arithmetic instructions for all data types
- Built-in exception handling
- Eight 80-bit numeric registers
- Expands Intel® 386 data types to include 32-/64-/80-bit floating point, 32-/64-bit integers, 18-digit BCD operands
- Sophisticated self-checking Testbench (Verilog versions use Verilog 2001)

Applications

The C387I can be utilized in a variety of applications including:

- floating-point computing applications
- generated fractal applications

Functional Description

The C387L core is partitioned into modules as shown in the block diagram above and described below.

FSM Controller

Main control state machine. It controls all execution units in C387L core. All built-in algorithms are coded in this unit. This module contains also instruction decoder.

Floating-Point Arithmetic Unit

Unit designed to calculate all non transcendental arithmetic operation. It is internally divided into exponent arithmetic unit and mantissa arithmetic unit. There is also additional align sub-block used to operand format conversions. Used to calculate addition, subtraction, multiplication, division and square root.

Cordic

Unit CORDIC is separated from the main arithmetic unit. It implements floating point operations on numbers formatted according to the IEEE STD 754. CORDIC module arguments may be normalized or denormalized numbers. The CORDIC is used for implementation of the following instructions of the C387L core: FSIN, FCOS, FSINCOS, FPTAN, FPATAN, F2XM1, FYL2X, FYL2XP1.

The CORDIC algorithm is implemented in HDL as a design entity, which is composed of several components. It uses 3 adders and 2 shift registers for mantissa computation, exponent unit, denormalization registers, normalization registers, coefficient ROM and controller.

Bus Interface Unit

The Bus Interface Unit (BIU) is an interface between internal part of coprocessor and external ports. A dedicated communication protocol was implemented to allow high-speed transfers of opcodes and operands between the host CPU and C387L coprocessor. The BIU contains the Data Registers, Instruction Registers, Status and Control Registers and Controller. BIU contains two subcomponents working on two different clocks, both are connected inside BIU.

Clock Divider

Clock control unit generates internal clocks for BIU interface part (CPUCLK) and other logic (NUMCLK). Additionally two clock signals for Dual Port RAM (CLKWR and CLKRD) are generated in this unit. Depending on the clock mode selection with CKM input the whole coprocessor can be triggered by two clocks (CPUCLK and CLKNUM) or by only one (CPUCLK)

FIFO Controller

Unit designed to load and store data from/to CPU. It's used for clock domain separation. FIFO size is fixed to 8x16 bits. FIFO is build on standard synchronous Dual Port RAM memory 1W1R. The core includes control unit for this memory. FIFO size is no restricted, depending on size PEREQ signal will be active often or rare.

Cordic constant ROM

Set of constants required by a Cordic unit for transcendental operations. ROM size is 64 address by 134 bits. The ROM memory should be asynchronous. All constants are put in the separate block to allow efficient synthesis optimization and use of technology depended dedicated ROM blocks.

Constants ROM

Set of constants used in non transcendental operations provided by floating-point arithmetic unit. Constants size is 64 address by 68-bits wise mantissa and 16-bits wise exponent. All constants are put in the separate block to allow efficient synthesis optimization and use of technology depended dedicated ROM blocks. Read access for constant ROM is fully synchronous and belongs to the system clk clock domain.

Stack

Eight 80-bits wise registers joined into the stack structure. It is used by C387L architecture as a store of all sources and results. Every operation takes one source operand from the top of stack and the second from the other position. Each of 80-bits wise line on the stack includes 64-bits wise mantissa, 15-bits wise exponent and 1-bit of sign.

Dual Port RAM

Dual Port synchronous RAM memory is used as a data storage buffer.

Implementation Results

C387L reference designs have been evaluated in a variety of technologies. The following are sample ASIC results using optimization for area.

Technology	Parameters	Speed	Area
UMC 0.18 μ m (typ)	Min area	100 MHz	94.2k gates
UMC 0.18 μ m (typ)	Max speed	160 MHz	98.3k gates

Support

The core as delivered is warranted against defects for ninety days from purchase. Thirty days of phone and email technical support are included, starting with the first interaction. Additional maintenance and support options are available.

Verification

The C387L core's functionality is based on the Intel® i387SX device. To prove the full compliance between these two devices the test environment is comparing the core behavior with the expected one captured in the pattern files. Pattern files were captured from the original Intel® i387SX device in the hardware test environment by means of a personal hardware modeler. The personal hardware modeler works with Mentor's ModelSim simulator.

The hardware environment was comprised of original Am386 and Intel® i387SX devices. The same set of the test cases was run on the original devices as the one delivered with the C387L core. The original device behavior was taken as a reference. All reference bus transactions are gathered in the pattern file.

Deliverables

The core is available in ASIC (synthesizable HDL) and FPGA (netlist) forms, and includes everything required for successful implementation. The ASIC version includes:

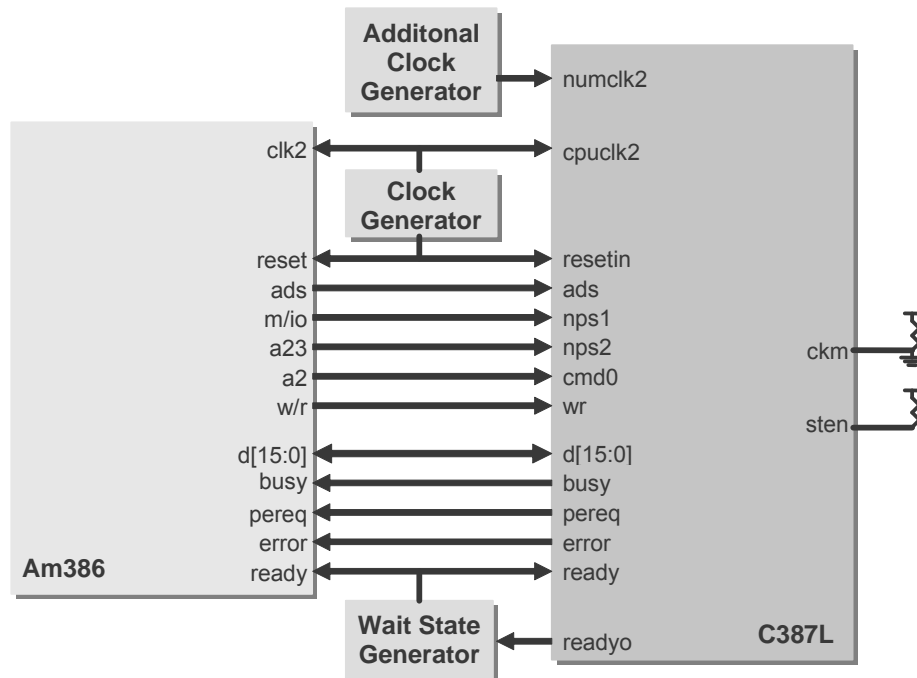
- HDL RTL source code

Example Application

The C387L core can be connected to the AMD386 or Intel® 386 CPU and it can support floating point computing.

The C387L's WR, NPS1, NPS2, ADS, CMD0, BUSY, ERORRN, PEREQ and D[15:0] pins are connected directly to the corresponding pins of the AMD386. The Clock Generator provides system clock for both clocks C387L when CKM pin is strapped to high or the Additional Clock Generator provides faster clock numclk2 for the floating-point unit when CKM pin is strapped to low.

The Clock Generator provides also the same reset signal for CPU and Math Coprocessor.



- Example C387L_CHIP -- this design uses the C387L and illustrates how to build and connect memories DPRAM, Clock control unit and three-state buffer.
- Sophisticated self-checking Testbench (Verilog versions use Verilog 2001) that instantiates example design C387L_CHIP, clock generator, process that stimulates external input signals, process that emulates the communication behavior between processor Am386 and C387L, and process that compares your simulation results with the expected results.
- A collection of C387L reference bus transaction are captured of original AMD386 and Intel® i387SX device which are executed directly by the Test Bench
- Simulation script, vectors, expected results, and comparison utility
- Synthesis script
- Comprehensive user documentation, including design specification, verification specification, test plan, and an integration manual