

DFPAU

Floating Point Arithmetic Coprocessor

v. 2.08

OVERVIEW

DFPAU is a Floating Point Arithmetic Coprocessor, designed to assist CPU in performing the floating point arithmetic computations. DFPAU directly replaces C software functions, by equivalent, very fast hardware operations, which significantly accelerate system performance. It doesn't require any programming, so it also doesn't require any modifications made in the main software. Everything is done automatically, during software compilation, by the DFPAU C driver.

DFPAU was designed to operate with DCD's DP8051, but can also operate with any other 8-, 16- and 32-bit processor. Drivers for all popular 8051 C compilers are delivered together with the DFPAU package.

DFPAU uses the specialized algorithms to compute arithmetic functions. It supports addition, subtraction, multiplication, division, square root, comparison, absolute value and change sign of a number. The input numbers format is in accordance with IEEE-754 standard single precision real numbers. Trigonometric functions are supported indirectly, because they are computed as set of add, multiply and divide operations, by software subroutines. **Each floating point function can be turned on/off** at configuration level, providing the flexible scalability of DFPAU module. It allows to save silicon space and provides exact configuration required by certain application.

DFPAU is a technology independent design, that can be implemented in a variety of process technologies.

APPLICATIONS

- Math coprocessors
- DSP algorithms
- Embedded arithmetic coprocessor
- Fast data processing & control

KEY FEATURES

- **Direct replacement** for C float software functions such as: +, -, *, /, ==, !=, >=, <=, <, >
- C interface supplied for all popular compilers: GNU C/C++, 8051 compilers
- No programming required
- Configurability of all available functions
- IEEE-754 Single precision real format support – **float type**
- Flexible arguments and result registers location
- Performs the following functions:
 - *FADD, FSUB* – addition, subtraction
 - *FMUL, FDIV* – multiplication, division
 - *FSQRT* – square root
 - *FCHS, FABS* – change of sign, absolute value
 - *FXAM* – examine input data
 - *FUCOM* – comparison
- Exceptions built-in routines
- Masks each exception indicator:
 - *Precision lack PE*
 - *Underflow result UE*
 - *Overflow result OE*
 - *Invalid operand IE*
 - *Division by zero ZE*
 - *Denormal operand DE*
- Fully configurable

- Fully synthesizable, static synchronous design with no internal tri-states

- FPGA EDIF/NGO/NGD/QXP/VQM called Netlist

DELIVERABLES

- Source code:
 - VHDL Source Code or/and
 - VERILOG Source Code or/and
 - Encrypted Netlist or/and
 - plain text EDIF netlist
- VHDL & VERILOG test bench environment
 - Active-HDL automatic simulation macros
 - NCSim automatic simulation macros
 - ModelSim automatic simulation macros
 - Tests with reference responses
- Technical documentation
 - Installation notes
 - HDL core specification
 - Datasheet
- Synthesis scripts
- Example application
- Technical support
 - IP Core implementation support
 - 3 months maintenance
 - Delivery the IP Core updates, minor and major versions changes
 - Delivery the documentation updates
 - Phone & email support

LICENSING

Comprehensible and clearly defined licensing methods, without royalty-per-chip fees, make using of IP Core easy and simple.

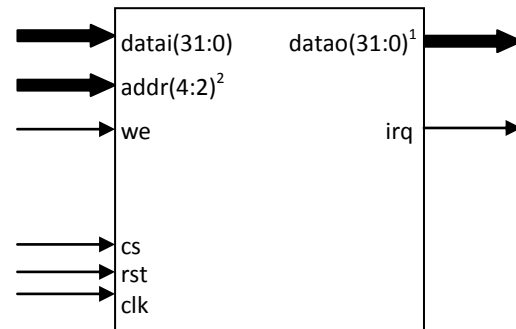
Single Site license option – it is dedicated for small and middle sized companies, running their business at one location.

Multi Sites license option – it is dedicated for corporate customers, running their business at several places. Licensed product can be used in selected company branches. In all cases, number of IP Core instantiation within a project and number of manufactured chips are unlimited. The license is royalty-per-chip free. There is no restrictions regarding the time of use.

There are two formats of delivered IP Core:

- VHDL, Verilog RTL synthesizable source code called HDL Source

SYMBOL



PINS DESCRIPTION

PIN	TYPE	DESCRIPTION
clk	Input	Global system clock
rst	Input	Global system reset
cs	Input	Chip select for read/write
datai[31:0] ¹	Input	Data bus input
addr[4:2] ²	Input	Register address to read/write
we	Input	Data write enable
datao[31:0] ¹	Output	Data bus output
int	Output	Interrupt request indicator

1 – data bus can be configured as 8-, 16- or 32- bit depends on processor's bus size

2 – address bus is aligned to work with 8- (3:0), 16- (3:1) or 32- (4:2) bit processors

BLOCK DIAGRAM

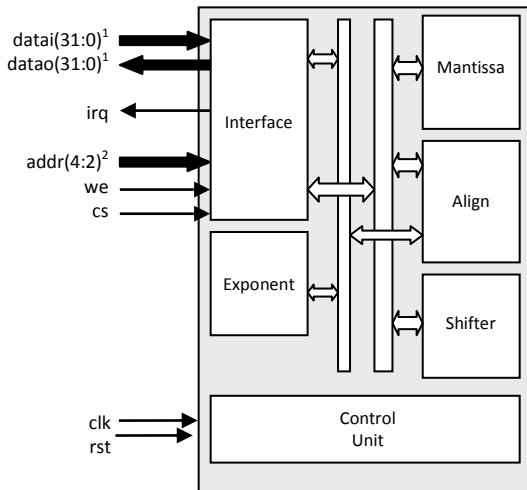
Mantissa – performs operations on mantissa part of number. The addition, subtraction, multiplication, division, square root, comparison and conversion operations, are executed in this module. It contains mantissas and work registers.

Exponent – performs operations on exponent part of number. The addition, subtraction, shifting, comparison and conversion operations, are executed in this module. It contains exponents and work registers.

Align – performs the numbers analysis against IEEE-754 standard compliance. Information about the data classes are passed, as result to appropriate internal module.

Shifter – performs mantissa shifting during normalization and denormalization operations. Information about shifted-out bits are stored for rounding process.

Control Unit – manages execution of all instructions and internal operation, required to execute particular function.



Interface – makes interface between external device and DFPAU internal 32-bit modules. It contains data, control and status registers. It can be configured to work with 8-, 16- and 32-bit processors.

PERFORMANCE

The following table gives a survey about the Core area and performance in ASIC devices (all key features have been included):

Technology	Optimization	Gates	F _{max}
0.25 typical	area	10000	100 MHz
0.25 typical	speed	12000	300 MHz

Core performance in ASIC devices

IMPROVEMENTS

The tables and figures below illustrates the system with DFPAU performance improvements for two typical CPU. DFPAU floating point instructions performance has been compared to standard C library functions delivered with every commercial C compiler. Each program was executed in the same system environments. Number of clock periods were measured between input

data loading into work registers and output result storing after operation. The results are placed in table below. Improvement has been computed as a number of clock cycles required by the CPU to compute FP operation, by the number of clocks required to compute the same operation by system of CPU with DFPAU:

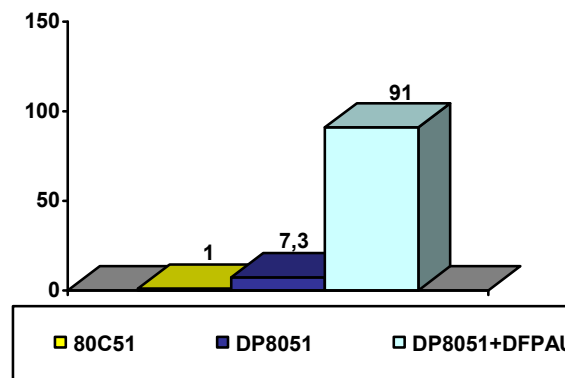
$$improvement = \frac{CPU_clocks}{CPU + DFPAU_clocks}$$

More details are available in core documentation.

The following table gives a survey about the DP8051+DFPAU performance compared to std 8051 microcontroller.

Device	Improvement
80C51	1.0
DP8051	7.3
DP8051+DFPAU	91.0

General performance improvements



IEEE-754 FP Instruction	Improvement
Addition	73
Subtraction	60
Multiplication	65
Division	182
Square Root	392
Sine	10
Cosine	10
Tangent	12
Arcs Tangent	17
Average speed improvement:	91

Improvements of particular operations



The table below shows performance improvements of the NIOS-II processor with DFPAU, compared to the same system without the DFPAU coprocessor.

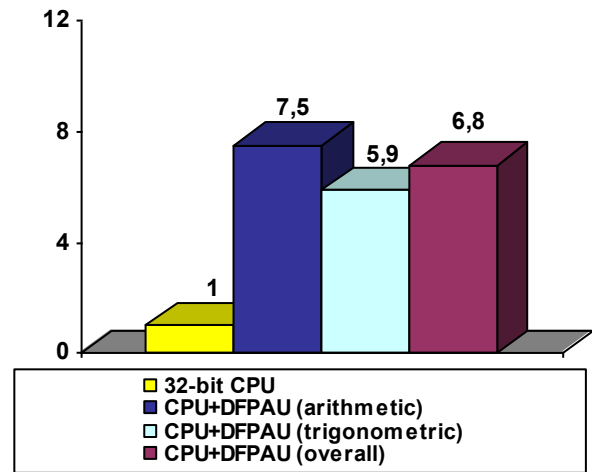
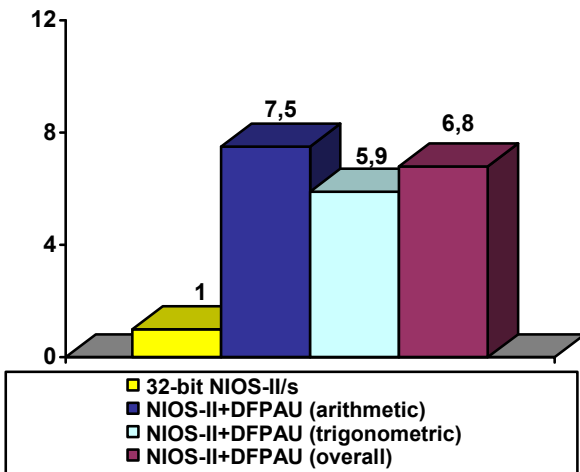
Device	Improvement
NIOS-II/s	1.0
NIOS-II+DFPAU (arithmetic)	7.5
NIOS-II+DFPAU (trigonometric)	5.9
NIOS-II+DFPAU (overall)	6.8

General performance improvements

The table below shows performance improvements of the sample 32-bit RISC CPU with DFPAU, compared to the same system without the DFPAU coprocessor.

Device	Improvement
CPU	1.0
CPU+DFPAU (arithmetic)	7.5
CPU+DFPAU (trigonometric)	5.9
CPU+DFPAU (overall)	6.8

General performance improvements



IEEE-754 FP Instruction	Improvement
Addition	6.4
Subtraction	6.5
Multiplication	5.1
Division	6.5
Square Root	12.9
Sine	5.2
Cosine	5.4
Tangent	5.8
Arcs Tangent	7.2
Average speed improvement:	6.8

Improvements of particular operations

IEEE-754 FP Instruction	Improvement
Addition	6.4
Subtraction	6.5
Multiplication	5.1
Division	6.5
Square Root	12.9
Sine	5.2
Cosine	5.4
Tangent	5.8
Arcs Tangent	7.2
Average speed improvement:	6.8

Improvements of particular operations

More details are available in core documentation.

More details are available in core documentation.

CONTACT

For any modification or special request, please contact Digital Core Design or local distributors.

Headquarters:

Wroclawska 94

41-902 Bytom, POLAND

e-mail: : info@dcd.pl

tel. : +48 32 282 82 66

fax : +48 32 282 74 37

Distributors:

Please check: <http://dcd.pl/sales/>