The DesignWare® ARC® FastMath Pack for ARC HS processors is a set of hardware extensions and an accompanying set of software wrapper functions that provide a collection of additional instructions supporting a range of mathematical functions that can be used with all ARC HS processors (Table 1). These include: basic saturating arithmetic, trigonometric functions, logarithmic and exponential functions, and fractional division and square root functions. The instructions support 16-bit and 32-bit data types. In addition to the extension instructions, the FastMath Pack includes a number of extension auxiliary registers.

All FastMath instructions are encoded in 32-bit formats and observe all of the established rules for encoding ARCv2 instructions for use with ARC HS processors. The instructions are implemented in the APEX extension space using the major opcode 0x07.

All FastMath instructions and register names have the prefix “FMP_” to ensure they do not create any namespace clashes with existing ARCv2 instructions or other customer-defined instructions.

The FastMath instructions significantly reduce cycle count compared to the same functions implemented in software. They are easy to use and implemented as standard C-functions in application code.

---

**Highlights**

- Mathematical functions for ARC HS processors
- Basic saturating arithmetic
- Trigonometric functions
- Logarithmic functions
- Exponential functions
- Square root functions
- Fractional division
- Support for 16-bit and 32-bit data types
- Significantly reduce cycle count and increase performance
- Implemented as standard C-functions in application code
<table>
<thead>
<tr>
<th>Function</th>
<th>Instruction(s)</th>
<th>Formats</th>
<th>Q15 cycles</th>
<th>Q31 cycles</th>
<th>Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Round16**</td>
<td>FMP_RNDH</td>
<td>Q15</td>
<td>1</td>
<td>NA</td>
<td>Round and saturate a 32-bit signed 2’s complement value to a 16-bit value</td>
</tr>
<tr>
<td>Saturate16**</td>
<td>FMP_SATH</td>
<td>Q15</td>
<td>1</td>
<td>NA</td>
<td>Saturate a 32-bit signed 2’s complement integer to a 16-bit value</td>
</tr>
<tr>
<td>Saturate Add**</td>
<td>FMP_ADDS</td>
<td>Q31</td>
<td>NA</td>
<td>2</td>
<td>32-bit signed addition with result saturation</td>
</tr>
<tr>
<td>Cosine</td>
<td>FMP_COS, FMP_COS15</td>
<td>Q15/Q31</td>
<td>15</td>
<td>15</td>
<td>Computes the cosine function on a fractional operand. The input operand represents an angle expressed in radians divided by pi.</td>
</tr>
<tr>
<td>Sine</td>
<td>FMP_SIN, FMP_SIN15</td>
<td>Q15/Q31</td>
<td>15</td>
<td>15</td>
<td>Computes the sine function on a fractional operand. The input operand represents an angle expressed in radians divided by pi.</td>
</tr>
<tr>
<td>Atan</td>
<td>FMP_ATAN, FMP_ATN15</td>
<td>Q15/Q31</td>
<td>25</td>
<td>25</td>
<td>Computes the inverse tangent function on a fractional operand. The input operand represents a real-valued tangent in the range (-1,0) and the result represents the angle expressed in radians divided by pi, for which the tangent is equal to the input operand.</td>
</tr>
<tr>
<td>Log2</td>
<td>FMP_LOG2, FMP_LOG215</td>
<td>Q15/Q31</td>
<td>13</td>
<td>13</td>
<td>Computes the base-2 logarithm of the fractional input operand X, when X is in the range (0.5,1). The result Y = log2(X) is a fraction in the range (-1,0).</td>
</tr>
<tr>
<td>Exponential (2^x)</td>
<td>FMP_EXP2, FMP_EXP215</td>
<td>Q15/Q31</td>
<td>11</td>
<td>11</td>
<td>Computes the base-2 exponentiation of the fractional input operand X, when X is in the range (-1,0). The result Y = 2^x is a fraction in the range (0.5,1).</td>
</tr>
<tr>
<td>Square root</td>
<td>FMP_SQRTF, FMP_SQRTF15</td>
<td>Q15/Q31</td>
<td>15</td>
<td>31</td>
<td>Computes the square root of a fractional argument X, when X ≥ 0</td>
</tr>
<tr>
<td>Division</td>
<td>FMP_DIVF, FMP_DIVF15</td>
<td>Q15/Q31</td>
<td>31</td>
<td>31</td>
<td>Fractional division of numerator X, but divisor Y, where abs (X) &lt; abs (Y) or X = -Y and Y ≠ 0</td>
</tr>
<tr>
<td>Reciprocal</td>
<td>FMP_RECIP, FMP_RECIP15</td>
<td>Q15/Q31</td>
<td>31</td>
<td>31</td>
<td>Computes the reciprocal of a fraction X, provided X ≠ 1</td>
</tr>
<tr>
<td>Log10**</td>
<td>FMP_LOG10, FMP_LOG1015</td>
<td>Q15/Q31</td>
<td>13</td>
<td>13</td>
<td>Computes the 10logX where X is the input operand. The input and output operands are both either 16-bit or 32-bit fractional numbers</td>
</tr>
<tr>
<td>Exponential** (10^x)</td>
<td>FMP_EXP10, FMP_EXP1015</td>
<td>Q15/Q31</td>
<td>11</td>
<td>11</td>
<td>Computes Ten to the power X, where X is the input operand. The input and output operands are both either 16-bit or 32-bit fractional numbers</td>
</tr>
</tbody>
</table>

Table 1. DesignWare ARC FastMath Pack Instructions

**16-Bit Signed Fractions (1Q15)**

When a 16-bit signed fraction is contained in a 32-bit register, it is located in the least significant 16 bits. The most significant 16 bits of the 32 bits are ignored by all operations using a 16-bit fraction as a source operand, and when a 16-bit fractional value is written to a destination register, bits [31:16] are cleared.

The FastMath extension pack contains instructions that use 16-bit fractions in 1Q15 format. Of the least significant 16 bits, bit [15] represents the sign, which is effectively also the integer portion, and bits [14:0] represent the fractional portion of the value. The maximum representable value is 0.999969 and the minimum representable value is -1.

**32-Bit Signed Fraction (1Q31)**

In a 32-bit signed fraction, the most significant bit, (bit 31), is the sign bit and the least significant bits (30:0) represent the 2’s complement of the fractional value.
Extension Auxiliary Register
The FastMath extension pack contains the following extension auxiliary register, that is also defined as part of the standard ARCv2 architecture and its FMP extensions.

- FMP Control Register, FMP_CTRL This register controls the rounding and saturation behavior of FastMath operations, and contains sticky saturation flag (SAT). (Address: 0x470, Access: rw)
  - RM (bits 0, 1) The RM field indicates the rounding mode.
    • 0: no rounding
    • 1: truncation, round down
    • 2: round up
    • 3: convergent rounding, round to nearest even
  - SAT (bit 16) The SAT field indicates the sticky saturation status.
    • 0: clear the flag
    • 1: indicates a DSP operation is saturated – this bit is sticky so you must explicitly write a 0 to clear the flag.

Complete Suite of Development Tools
The ARC HS FastMath Pack option is supported by a complete suite of development tools. This includes the MetaWare Development Toolkit that generates highly efficient code, ideal for deeply embedded applications, the ARC xCAM and nSIM simulators and the ARCHitect configuration tool.

Testing, Compliance and Quality
Verification of the DesignWare ARC Trace Interface follows a bottom-up verification methodology from block level through system level. Each functional block within the product follows a functional, coverage-driven test plan.

Deliverables
The DesignWare ARC FastMath Pack for HS is delivered as Verilog HDL in the ARCHitect IP Library. The HDL is configured and output from the ARCHitect IP Configurator tool.

About DesignWare IP
Synopsys is a leading provider of high-quality, silicon-proven IP solutions for SoC designs. The broad DesignWare IP portfolio includes logic libraries, embedded memories, embedded test, analog IP, wired interface IP, wireless interface IP, security IP, embedded processors, and subsystems. To accelerate prototyping, software development and integration of IP into SoCs, Synopsys’ IP Accelerated initiative offers IP Prototyping Kits, IP Virtual Development Kits and IP subsystems. Synopsys’ extensive investment in IP quality, comprehensive technical support and robust IP development methodology enables designers to reduce integration risk and accelerate time-to-market.

For more information on DesignWare IP, visit http://www.synopsys.com/designware. Follow us on Twitter at http://twitter.com/designware_ip.