# **FAST INVERSE SQUARE ROOT (****Pipelined)**

Engineer: Xianyou Li

Revision: 0.01 – File Created

Create Date: 9:21:00 10/20/2017

## Introduction

Computing reciprocal square roots is necessary in many applications, such as vector normalization in video games. Often, some loss of precision is acceptable for a large increase in speed.

This program uses Newton iteration method to calculate reciprocal square roots with the magic initial guess presented by John Carmack. We need to convert the fixed-point to a floating point because the initial guess can only be obtained in 32-bits floating point format. With the magic initial guess in fixed-pint format, we can get the reciprocal square roots by just one Newton iteration.

## Quick Facts

|  |  |  |
| --- | --- | --- |
| CoreRequirements | FPGA Families Supported | Lattice ECP5UM |
| Minimal Device Needed |  |
| ResourceUtilization | Targeted Device | LFE5UM-85FCABGA756 |
| LUTs | 690 |
| sysMEM EBRs | 0 |
| Registers | 597 |
| DSP MULT | 24 |
| DSP ALU | 12 |
| Design ToolSupport | Lattice Implementation | Lattice Diamond 3.9.0.99.2 |
| Synthesis | Synopsys Synplify Pro for Lattice D-2016 09L |
| Simulation | Aldec Active-HDL 10.3 Lattice Edition |
| Performance | clk(MHz) | 200 |
| Latency(clk) | 20 |

## Data Format

###  32-bits Fixed-point Data Format

|  |  |  |
| --- | --- | --- |
| S(Sign) | I(Integer Part) | F(Fractional Part) |
| 1bit | 8bits | 23bits |
| Bit31 | 30🡨Bits🡪23 | 22🡨Bits🡪0 |

Hence, the represented value is

$$x=\left(-1\right)^{s}\left(I+\frac{F}{2^{23}}\right)$$

###  32-bits Floating Point Data Format (IEEE 754-1985)

|  |  |  |
| --- | --- | --- |
| S(Sign) | E(Exponent) | M(Mantissa) |
| 1bit | 8bits | 23bits |
| Bit31 | 30🡨Bits🡪23 | 22🡨Bits🡪0 |

Hence, the represented value is

$$x=\left(-1\right)^{s}\left(1+\frac{M}{2^{23}}\right)2^{E-127}$$