8, 16 & 32 bit FPGA implementation of Unsigned integer SRT Radix2 division algorithm with comparative analysis of power, delay and area.

Diksha Choudhary, Ravi Singh Choudhary, ,Vinayak Awasthi

The comparative literature review of different division algorithms was done with their ease of implementation on hardware. Different algorithms such as booth division algorithm, nonrestoring division algorithm and restoring division algorithm were studied but none of them were found to be fast enough for hardware implementation. Finally the SRT (Sweeney, Robertson, Tocher) algorithm was found to be suitable for hardware implementation. Many different forms of the algorithm are already in practical use in current hardwares. The project was aimed at implementing a simple basic Radix2 SRT algorithm on Xilinx zynq zedboard. The algorithm was implemented for 8, 16 and 32 bit and a comparative study for power consumption, area in terms of LUTs and delay is presented. Finally a floating point division algorithm implementation was also attempted using the SRT Radix2 unsigned algorithm developed.

There were a number of design challenges met during the course of the project. Firstly during the synthesis a negative slack time error was incurred. After many solutions it was resolved by increasing the period. Secondly during the floating point implementation, design challenge regarding the IEEE 754 format number representation was faced. It was finally assumed that the user input as well as the output shall be taken in terms of IEEE 754 single precision format.

The algorithm used is simple.

- Firstly the dividend and the divisor are taken as a input from the user and stored in registers inA and inB respectively.
- If inB has K leading zeros, shift inB and inPA “K” bits left.
- Loop for n times where n is the total number of bits.

  A. If top 3 bits of inPA are equal set inA[0] = 0 and shift inPA 1 bit left.

  B. If top 3 bits of inPA are unequal and inPA is negative set inPA[0] = -1 and shift inPA left.

  C. Otherwise if inPA is positive set inPA[0] = 1 and shift inPA left 1 bit.

- If final remainder is negative (inPA is negative) correct remainder by adding B and subtract 1 from quotient (inA).
- Shift remainder K bits right.

The final comparative analysis for 8, 16 and 32 bit algorithm is as follows.

Parameter	8 bit	16 bit	32 bit	Floating point
Power	0.104 W	0.106 W	0.105 W	0.105 W
Area	1173	2382	7756	1688
Delay	27.09 ns	26.375 ns	25.294 ns	27.228 ns

Diksha Choudhary 2017H1030129T

Ravi Singh Choudhary 2018H1400174P

Vinayak Awasthi 2013HD400658P