Literature Review Summary for EE-800
Name: Vishesh Kalra
ID: 11089943
Literature Review 1
Functionally Linear Decomposition and Synthesis (FLDS) of Logic Circuits for FPGAs
@Tomasz S. Czajkowski and Stephen D. Brown

Concept of Linearity-F(x)=∑G_i(y)*H_i(x-y) where X,Y are set of Variables(Y≤X),∑ represents XOR gate. F is weighted sum of functions of G_i (Basis), where weighting factors are determined by H_i (Selector). This retains ability to synthesize original function using Davio/Shanon’s as well as BDD. FLDS VS BDS-PGA VS ABC-XOR based logic functions can be significantly reduced in size and logic depth 18.8%/14.5% and 25.3% in area and 7.7% in depth respectively. FLDS VS ABC for Non-XOR based Circuits ABC produces circuits with 6.2% lower area results and 16.5 lower depth.

Literature Review 2
Floating-Point Division and Square Root Implementation using Taylor-Series Expansion Algorithm with Reduced Look-up Tables
@Taek-Jun Kwon, Jeff Draper

Division and Square root are open considered as infrequent in general purpose applications. But in modern applications like CAD tools and 3D Graphics DIV/SQRT become performance bottlenecks. A fused Floating point multiply/square root/divide unit has been presented. Algorithm used- Taylor Series Expansion with reduced lookup tables.

For Yo, generally two look up tables are used based upon exponent value of input operand— even or odd. To generate IEEE 754 standard 8 bit seed is used when exponent is even and 9 bit if odd. In addition to this Yo2 values are used via a small multiplier (9b*9b). To incorporate a Square root function with a modest 20 % increase is desirable and to incorporate it in entire FPU (Floating Point Unit) is a mere 10 %. Multiplier Approach for generating constants was done and resulted in area savings of 2.4 %.

Literature Review 3
Energy and Delay Improvement via Decimal Floating Point Units
@Hossam A.H.Fahmy, and RamyRaafat, Amira, M.Abdel-Majeed, Rodina, Samy, TarekElDeeb, YasminFarouk,
Decimal Adder

Decimal Multiplier - Two main paths - Significand path and Exponent path. Significand Path relies on a Parallel Multiplier to generate partial products in parallel and reduce them to two vectors (sum and carry) using a carry save addition tree. These two vectors are added using the new decimal adder proposed.

DIV/SQRT - Algorithm used - Modified Newton Raphson. Iterate on \( x_{i+1} = x_i \left(2 - b \frac{x_i}{x_i^2}\right)\) to find out reciprocal of \(b\) and \( x_{i+1} = x_i \left(3 - b \frac{x_i}{x_i^2}\right)/2 \) to find out reciprocal square root of \(b\). The author proposes new rounding scheme - truncates quotient to \(p\) digits and checks actual reminder to decide on the correct rounding.

Result-Delay in Divider is less than 1/3 times of 2300 FO4 for Decimal 64 design proposed by Wang and Schulte. Using the Power Play Estimator tool of Altera, the estimated average power/instruction is 109 mW, while FPGA running at 50 MHz. Energy Delay Product of over 500.

Literature Review 4
Real Time Image Feature Vector Generator Employing Functional Cache Memory for Edge Flags @Takuki Nakagawa and Tadashi Shibata

- Edge filtering is carried out in 4 directions i.e. Horizontal, Vertical, +45 and -45 degrees and four edge maps are generated from 64x64 recognition window. Then a Feature vector is generated by dividing each edge map into 16 bins and number of edge flags in each bin are counted and 64 dimension feature vector is generated by concatenating 4 different histograms. Function of this unit is to generate a 16 element Edge Histogram. The unit is composed of Functional Cache Memory for storing edge flag bits and Processing Element array for Edge Counting. The Functional Cache Memory includes four 64x65-SRAM banks and two Crossbar switches to be used for reordering of edge flag locations. The Architecture enables us to generate 3.9x10^7 feature vectors/second (@ 100 MHz) which is 5x10^3 times faster than software processing using 2.16-GHz processor.