

Wave Division Multiplexing Routing on FPGAs

Louis Pochet
Rome Research Site, AFRL

This paper describes an FPGA based graph coloring acceleration architecture. The approach described maximizes the use of local communication while still ensuring a complete search of the solution domain usually only ensured through more global approaches. The graph coloring accelerator will be demonstrated to speed up routing for Wave Division Multiplexing fiber optic communications systems.

Wave Division Multiplexing (WDM) can decrease the cost of fiber optic cable in a communication system. WDM is a technique used to increase the effective bandwidth of fiber optic communication lines by allowing multiple wavelengths (colors) to be used in a single channel. Each color can exist independently of any other colors in the same channel. The only constraint on use of WDM comes at routing nodes. Each color can only be represented at each node once. This constraint maps to an N-P complete problem known as the Latin Squares or quasigroup problem.

The quasigroup graph coloring problem can be visualized as an $N \times N$ matrix, with each position row of the matrix representing a routing node. With N colors available to color the graph, the rule for coloring the graph is that each color can appear in each row and each column only once. It has been shown that for graphs with very few constraints (below 20% of nodes pre-selected), a solution appears rather quickly. It has also been shown that for graphs with many constraints (over 80% of nodes pre-selected) a solution, or proof that there is no solution, also appears quickly. Long compute times are required to color or prove that no mapping exists when some intermediate level of constraints is given.

The graph coloring acceleration architecture consists of; small generic graph nodes that actually perform the graph coloring, graph edges that keep track of all backtrack information, and a graph master that handles host-FPGA interaction. Each node of the graph has a corresponding processor on the FPGA. Once initial constraints have been set, the color of selected nodes will be guessed from the available set of colors. This update then causes implications in other nodes that can no longer be the guessed color. Guessing continues until a contradiction is arrived at or the graph is colored. When a node's color has been established, the FPGA can be reconfigured to remove the finished node. When the FPGA's resources are

not large enough to fit an entire graph problem, a windowing technique can be employed to simulate a larger FPGA than available.

The global communications requirements in this implementation are for backtracking from incorrect guesses and making new guesses. Backtracks and guesses are handled as ripples in the matrix fabric. This ripple begins where the contradiction was discovered and continues until all nodes have backtracked. Guessing must be handled globally to keep multiple nodes from guessing at the same time. This would make discerning the incorrect guess causing a contradiction difficult.

Graph coloring does not map well to general purpose processors because it consists of many simple integer comparisons, where each color is designated an integer representation. There is also little use for the global memory available to general purpose processors. A node is only required to communicate with other nodes in the same row and column. Nodes in different rows and columns need no direct communications. Field Programmable Gate Arrays (FPGAs) have strengths that correspond to the requirements of the quasigroup problem. FPGAs lack efficient means for global communication, but excel when local communications are required. An FPGA can also be partitioned into many simple processors running in parallel to be able to perform more comparisons in a given span of time.