Overview of GSLP in VeGen

1 Analysis

- AAResults: This pass provides alias analysis information, which determines whether two pointers can point to the same memory location or not. It also provides mod-ref information, which determines whether a pointer or a function can modify or read a memory location or not.

- ScalarEvolution: This pass provides scalar evolution information, which analyzes the evolution of scalar expressions in loops. It can compute the number of iterations, the trip count, and the exit condition of loops, as well as the value of induction variables and other expressions at any point in the loop.

- DominatorTree: This pass provides dominator tree information, which represents the dominance relation between basic blocks in a function. A basic block A dominates another basic block B if every path from the entry of the function to B goes through A. The dominator tree is useful for many optimizations and analyses, such as dead code elimination, loop detection, and reaching definitions.

- LoopInfo: This pass provides natural loop information, which identifies the loops in a function and their nesting structure. A natrual loop is a set of basic blocks that has a single entry (the header) and a back edge from a latch to the header. The loop info pass also computes the loop depth, the loop preheader, and the loop exit blocks for each loop.

- DependenceInfo: This pass provides dependence analysis information, which determines whether two memory accesses in a loop are dependent or not. Two memory accesses are dependent if they access the same memory location and at least one of them is a write. The dependence analysis can classify the dependence into defferent types, such as flow, anti, output, or input dependence, and can compute the distance and direction vectors for each dependence.

- LazyValueInfo: This pass provides lazy value information, which computes the possible values of an instruction at a given point in the program. It uses a demand-driven algorithm that only analyzes the values that are requested by the client. It can handle simple constant values, ranges of values, and bitwise operations.

- TargetTransformInfo: This pass provides target transform information, which estimates the cost and profitability of various code transformations for a specific target. It can provide information such as the instruction cost, the intrinsic cost, the vectorization factor, and the unrolling factor for defferent types of operations and data types.

- BlockFrequencyInfo: This pass provides block frequency information, which estimates the execution frequency of each basic block in a function. It uses a branch probability analysis to compute the relative probabilities of taking different branches, and then propagates these probabilities along the control flow graph to obtain the block frequencies.

2 Irreducible Control Flow

VeGen does not handle irreducible control flow graphs, switches, invocations, or infinite loops.

3 Supported Intrinsics

The IRVec header file defines a set of supported intrinsics. However, our requirements extend beyond the intrinsics listed in the intrinsic guide. Specifically, we require nine categories of intrinsics: Binary, Unary, Truncate, Extension, Select, UnaryMath, IntToFloat, FloatToInt, and BinaryMath.

In VeGen, an instruction is a list of lanes, each characterized by a BoundOperation. A BoundOperation binds a vector instruction to its inputs. These inputs are represented by InputSlice, which specifies the input ID and range (i.e., the positions in the input vector).

Now, let me introduce the nine categories of intrinsics and explain their origins.

4 Loop Unroll

VeGen computes the unroll factor for every loop and then unrolls loops based on the unroll factor.

5 Get Seed Packs

SeedOperands is a vector of OperandPack, which is a vector of values with extended links to their producers. It is associated with a FixedVectorType.

6 BottomUp Optimization

7 CodeGen