| Names | LLVM Current Cost Model | Dan Weber (@omnisip) | Thomas Lively (@tlively) |
|---|---|---|---|
| Scoring Strategy | Every instruction is equally valued (?) | Bias based on implementations in V8 for each instruction relative to x64 and ARM64 for Skylake and Cortex-A76 | Egalitarian approach maximizing worst case performance for each architecture on each instruction |
| Instruction | | | |
| v128.load | | | |
| v128.load8x8_s | | | |
| v128.load8x8_u | | | |
| v128.load16x4_s | | | |
| v128.load16x4_u | | | |
| v128.load32x2_s | | | |
| v128.load32x2_u | | | |
| v128.load8_splat | | | |
| v128.load16_splat | | | |
| v128.load32_splat | | | |
| v128.load64_splat | | | |
| v128.store | | | |
| v128.const | | | |
| i8x16.shuffle | | | |
| i8x16.swizzle | | | |
| i8x16.splat | | | |
| i16x8.splat | | | |
| i32x4.splat | | | |
| i64x2.splat | | | |
| f32x4.splat | | | |
| f64x2.splat | | | |
| i8x16.extract_lane_s | | | |
| i8x16.extract_lane_u | | | |
| i8x16.replace_lane | | | |
| i16x8.extract_lane_s | | | |
| i16x8.extract_lane_u | | | |
| i16x8.replace_lane | | | |
| i32x4.extract_lane | | | |
| i32x4.replace_lane | | | |
| i64x2.extract_lane | | | |
| i64x2.replace_lane | | | |
| f32x4.extract_lane | | | |
| f32x4.replace_lane | | | |
| f64x2.extract_lane | | | |
| f64x2.replace_lane | | | |
| i8x16.eq | | | |
| i8x16.ne | | | |
| i8x16.lt_s | | | |
| i8x16.lt_u | | | |
| i8x16.gt_s | | | |
| i8x16.gt_u | | | |
| i8x16.le_s | | | |
| i8x16.le_u | | | |
| i8x16.ge_s | | | |
| i8x16.ge_u | | | |
| i16x8.eq | | | |
| i16x8.ne | | | |
| i16x8.lt_s | | | |
| i16x8.lt_u | | | |
| i16x8.gt_s | | | |
| i16x8.gt_u | | | |

| | | | |
|---|---|---|---|
| i16x8.le_s | | | |
| i16x8.le_u | | | |
| i16x8.ge_s | | | |
| i16x8.ge_u | | | |
| i32x4.eq | | | |
| i32x4.ne | | | |
| i32x4.lt_s | | | |
| i32x4.lt_u | | | |
| i32x4.gt_s | | | |
| i32x4.gt_u | | | |
| i32x4.le_s | | | |
| i32x4.le_u | | | |
| i32x4.ge_s | | | |
| i32x4.ge_u | | | |
| f32x4.eq | | | |
| f32x4.ne | | | |
| f32x4.lt | | | |
| f32x4.gt | | | |
| f32x4.le | | | |
| f32x4.ge | | | |
| f64x2.eq | | | |
| f64x2.ne | | | |
| f64x2.lt | | | |
| f64x2.gt | | | |
| f64x2.le | | | |
| f64x2.ge | | | |
| v128.not | | | |
| v128.and | | | |
| v128.andnot | | | |
| v128.or | | | |
| v128.xor | | | |
| v128.bitselect | | | |
| i8x16.abs | | | |
| i8x16.neg | | | |
| i8x16.any_true | | | |
| i8x16.all_true | | | |
| i8x16.bitmask | | | |
| i8x16.narrow_i16x8_s | | | |
| i8x16.narrow_i16x8_u | | | |
| i8x16.shl | | | |
| i8x16.shr_s | | | |
| i8x16.shr_u | | | |
| i8x16.add | | | |
| i8x16.add_sat_s | | | |
| i8x16.add_sat_u | | | |
| i8x16.sub | | | |
| i8x16.sub_sat_s | | | |
| i8x16.sub_sat_u | | | |
| i8x16.min_s | | | |
| i8x16.min_u | | | |
| i8x16.max_s | | | |
| i8x16.max_u | | | |
| i8x16.avgr_u | | | |
| i16x8.abs | | | |
| i16x8.neg | | | |
| i16x8.any_true | | | |
| i16x8.all_true | | | |

| | | | |
|---|---|---|---|
| i16x8.bitmask | | | |
| i16x8.narrow_i32x4_s | | | |
| i16x8.narrow_i32x4_u | | | |
| i16x8.widen_low_i8x16_s | | | |
| i16x8.widen_high_i8x16_s | | | |
| i16x8.widen_low_i8x16_u | | | |
| i16x8.widen_high_i8x16_u | | | |
| i16x8.shl | | | |
| i16x8.shr_s | | | |
| i16x8.shr_u | | | |
| i16x8.add | | | |
| i16x8.add_sat_s | | | |
| i16x8.add_sat_u | | | |
| i16x8.sub | | | |
| i16x8.sub_sat_s | | | |
| i16x8.sub_sat_u | | | |
| i16x8.mul | | | |
| i16x8.min_s | | | |
| i16x8.min_u | | | |
| i16x8.max_s | | | |
| i16x8.max_u | | | |
| i16x8.avgr_u | | | |
| i32x4.abs | | | |
| i32x4.neg | | | |
| i32x4.any_true | | | |
| i32x4.all_true | | | |
| i32x4.bitmask | | | |
| i32x4.widen_low_i16x8_s | | | |
| i32x4.widen_high_i16x8_s | | | |
| i32x4.widen_low_i16x8_u | | | |
| i32x4.widen_high_i16x8_u | | | |
| i32x4.shl | | | |
| i32x4.shr_s | | | |
| i32x4.shr_u | | | |
| i32x4.add | | | |
| i32x4.sub | | | |
| i32x4.mul | | | |
| i32x4.min_s | | | |
| i32x4.min_u | | | |
| i32x4.max_s | | | |
| i32x4.max_u | | | |
| i32x4.dot_i16x8_s | | | |
| i64x2.neg | | | |
| i64x2.shl | | | |
| i64x2.shr_s | | | |
| i64x2.shr_u | | | |
| i64x2.add | | | |
| i64x2.sub | | | |
| i64x2.mul | | | |
| f32x4.ceil | | | |
| f32x4.floor | | | |
| f32x4.trunc | | | |
| f32x4.nearest | | | |
| f64x2.ceil | | | |
| f64x2.floor | | | |
| f64x2.trunc | | | |
| f64x2.nearest | | | |

| | | | |
|---|---|---|---|
| f32x4.abs | | | |
| f32x4.neg | | | |
| f32x4.sqrt | | | |
| f32x4.add | | | |
| f32x4.sub | | | |
| f32x4.mul | | | |
| f32x4.div | | | |
| f32x4.min | | | |
| f32x4.max | | | |
| f32x4.pmin | | | |
| f32x4.pmax | | | |
| f64x2.abs | | | |
| f64x2.neg | | | |
| f64x2.sqrt | | | |
| f64x2.add | | | |
| f64x2.sub | | | |
| f64x2.mul | | | |
| f64x2.div | | | |
| f64x2.min | | | |
| f64x2.max | | | |
| f64x2.pmin | | | |
| f64x2.pmax | | | |
| i32x4.trunc_sat_f32x4_s | | | |
| i32x4.trunc_sat_f32x4_u | | | |
| f32x4.convert_i32x4_s | | | |
| f32x4.convert_i32x4_u | | | |
| v128.load32_zero | | | |
| v128.load64_zero | | | |
| i16x8.extmul_low_i8x16_s | | | |
| i16x8.extmul_high_i8x16_s | | | |
| i16x8.extmul_low_i8x16_u | | | |
| i16x8.extmul_high_i8x16_u | | | |
| i32x4.extmul_low_i16x8_s | | | |
| i32x4.extmul_high_i16x8_s | | | |
| i32x4.extmul_low_i16x8_u | | | |
| i32x4.extmul_high_i16x8_u | | | |
| i64x2.extmul_low_i32x4_s | | | |
| i64x2.extmul_high_i32x4_s | | | |
| i64x2.extmul_low_i32x4_u | | | |
| i64x2.extmul_high_i32x4_u | | | |