

```
%9:
%10 = shl i64 %6, 5
%11 = shl i64 %7, 3
%sub.i = add nsw i32 %3, -1
%sub4.i = add nsw i32 %4, -1
%12 = trunc i64 %7 to i32
%13 = mul i32 %12, %4
%14 = shl i32 %13, 3
%15 = trunc i64 %6 to i32
%16 = shl i32 %15, 5
%17 = add i32 %14, %16
%18 = zext i32 %4 to i64
%19 = or i32 %17, 1
%20 = shl i32 %12, 3
%21 = or i32 %20, 1
%22 = mul i32 %21, %4
%23 = add i32 %22, %16
%24 = trunc i64 %7 to i32
%25 = mul i32 %24, %4
%26 = shl i32 %25, 3
%27 = trunc i64 %6 to i32
%28 = shl i32 %27, 5
%29 = add i32 %26, %28
%30 = zext i32 %4 to i64
%31 = %30
%32 = zext i32 %63 to i64
%33 = %32
%34 = %33
%35 = %34
%36 = %35
%37 = %36
%38 = %37
%39 = %38
%40 = %39
%41 = %40
%42 = %41
%43 = %42
%44 = %43
%45 = %44
%46 = %45
%47 = %46
%48 = %47
%49 = %48
%50 = %49
%51 = %50
%52 = %51
%53 = %52
%54 = %53
%55 = %54
%56 = %55
%57 = %56
%58 = %57
%59 = %58
%60 = %59
%61 = %60
%62 = %61
%63 = %62
%64 = %63
%65 = %64
%66 = %65
%67 = %66
%68 = %67
%69 = %68
%70 = %69
%71 = %70
%72 = %71
%73 = %72
%74 = %73
%75 = %74
%76 = %75
%77 = %76
%78 = %77
%79 = %78
%80 = %79
%81 = %80
%82 = %81
%83 = %82
%84 = %83
%85 = %84
%86 = %85
%87 = %86
%88 = %87
%89 = %88
%90 = %89
%91 = %90
%92 = %91
%93 = %92
%94 = %93
%95 = %94
%96 = %95
%97 = %96
%98 = %97
%99 = %98
%100 = %99
%101 = %100
%102 = %101
%103 = %102
%104 = %103
%105 = %104
%106 = %105
%107 = %106
%108 = %107
%109 = %108
%110 = %109
%111 = %110
%112 = %111
%113 = %112
%114 = %113
%115 = %114
%116 = %115
%117 = %116
%118 = %117
%119 = %118
%120 = %119
%121 = %120
%122 = %121
%123 = %122
%124 = %123
%125 = %124
%126 = %125
%127 = %126
%128 = %127
%129 = %128
%130 = %129
%131 = %130
%132 = %131
%133 = %132
%134 = %133
%135 = %134
%136 = %135
%137 = %136
%138 = %137
%139 = %138
%140 = %139
%141 = %140
%142 = %141
%143 = %142
%144 = %143
%145 = %144
%146 = %145
%147 = %146
%148 = %147
%149 = %148
%150 = %149
%151 = %150
%152 = %151
%153 = %152
%154 = %153
%155 = %154
%156 = %155
%157 = %156
%158 = %157
%159 = %158
%160 = %159
%161 = %160
%162 = %161
%163 = %162
%164 = %163
%165 = %164
%166 = %165
%167 = %166
%168 = %167
%169 = %168
%170 = %169
%171 = %170
%172 = %171
%173 = %172
%174 = %173
%175 = %174
%176 = %175
%177 = %176
%178 = %177
%179 = %178
%180 = %179
%181 = %180
%182 = %181
%183 = %182
%184 = %183
%185 = %184
%186 = %185
%187 = %186
%188 = %187
%189 = %188
%190 = %189
%191 = %190
%192 = %191
%193 = %192
%194 = %193
%195 = %194
%196 = %195
%197 = %196
%198 = %197
%199 = %198
%200 = %199
%201 = %200
%202 = %201
%203 = %202
%204 = %203
%205 = %204
%206 = %205
%207 = %206
%208 = %207
%209 = %208
%210 = %209
%211 = %210
%212 = %211
%213 = %212
%214 = %213
%215 = %214
%216 = %215
%217 = %216
%218 = %217
%219 = %218
%220 = %219
%221 = %220
%222 = %221
%223 = %222
%224 = %223
%225 = %224
%226 = %225
%227 = %226
%228 = %227
%229 = %228
%230 = %229
%231 = %230
%232 = %231
%233 = %232
%234 = %233
%235 = %234
%236 = %235
%237 = %236
%238 = %237
%239 = %238
%240 = %239
%241 = %240
%242 = %241
%243 = %242
%244 = %243
%245 = %244
%246 = %245
%247 = %246
%248 = %247
%249 = %248
%250 = %249
%251 = %250
%252 = %251
%253 = %252
%254 = %253
%255 = %254
%256 = %255
%257 = %256
%258 = %257
%259 = %258
%260 = %259
%261 = %260
%262 = %261
%263 = %262
%264 = %263
%265 = %264
%266 = %265
%267 = %266
%268 = %267
%269 = %268
%270 = %269
%271 = %270
%272 = %271
%273 = %272
%274 = %273
%275 = %274
%276 = %275
%277 = %276
%278 = %277
%279 = %278
%280 = %279
%281 = %280
%282 = %281
%283 = %282
%284 = %283
%285 = %284
%286 = %285
%287 = %286
%288 = %287
%289 = %288
%290 = %289
%291 = %290
%292 = %291
%293 = %292
%294 = %293
%295 = %294
%296 = %295
%297 = %296
%298 = %297
%299 = %298
%300 = %299
%301 = %300
%302 = %301
%303 = %302
%304 = %303
%305 = %304
%306 = %305
%307 = %306
%308 = %307
%309 = %308
%310 = %309
%311 = %310
%312 = %311
%313 = %312
%314 = %313
%315 = %314
%316 = %315
%317 = %316
%318 = %317
%319 = %318
%320 = %319
%321 = %320
%322 = %321
%323 = %322
%324 = %323
%325 = %324
%326 = %325
%327 = %326
%328 = %327
%329 = %328
%330 = %329
%331 = %330
%332 = %331
%333 = %332
%334 = %333
%335 = %334
%336 = %335
%337 = %336
%338 = %337
%339 = %338
%340 = %339
%341 = %340
%342 = %341
%343 = %342
%344 = %343
%345 = %344
%346 = %345
%347 = %346
%348 = %347
%349 = %348
%350 = %349
%351 = %350
%352 = %351
%353 = %352
%354 = %353
%355 = %354
%356 = %355
%357 = %356
%358 = %357
%359 = %358
%360 = %359
%361 = %360
%362 = %361
%363 = %362
%364 = %363
%365 = %364
%366 = %365
%367 = %366
%368 = %367
%369 = %368
%370 = %369
%371 = %370
%372 = %371
%373 = %372
%374 = %373
%375 = %374
%376 = %375
%377 = %376
%378 = %377
%379 = %378
%380 = %379
%381 = %380
%382 = %381
%383 = %382
%384 = %383
%385 = %384
%386 = %385
%387 = %386
%388 = %387
%389 = %388
%390 = %389
%391 = %390
%392 = %391
%393 = %392
%394 = %393
%395 = %394
%396 = %395
%397 = %396
%398 = %397
%399 = %398
%400 = %399
%401 = %400
%402 = %401
%403 = %402
%404 = %403
%405 = %404
%406 = %405
%407 = %406
%408 = %407
%409 = %408
%410 = %409
%411 = %410
%412 = %411
%413 = %412
%414 = %413
%415 = %414
%416 = %415
%417 = %416

```

```

pre: region for entry:pre: region for init.i:
% local id y.0 = phi i64 [ 0, %9 ], [ %96, %region_for_end.i ]
%37 = mul i64 % local id y.0, %30
%38 = trunc i64 %37 to i32
%39 = add i32 %29, %38
%40 = sext i32 %39 to i64
%sevcgep10 = getelementptr float, float* %2, i64 %40
%sevcgep11 = getelementptr float, float* %sevcgep10, i64 %40
%41 = add i64 %37, %32
%sext = shl i64 %41, 32
%42 = ashr exact i64 %sext, 32
%sevcgep18 = getelementptr float, float* %0, i64 %42
%sevcgep21 = getelementptr float, float* %sevcgep20, i64 %42
%43 = trunc i64 %37 to i32
%44 = add i32 %36, %43
%45 = sext i32 %44 to i64
%sevcgep28 = getelementptr float, float* %1, i64 %45
%sevcgep31 = getelementptr float, float* %sevcgep30, i64 %45
%46 = mul i64 % local id y.0, %18
%47 = add nsw nsw i64 % local id y.0, %11
%conv2.i = trunc i64 %47 to i32
%cmp.i = icmp sgt i32 %sub.i, %conv2.i
%mul.i = mul nsw i32 %conv2.i, %4
%add18.i = add nsw i32 %conv2.i, 1
%mul19.i = mul nsw i32 %add18.i, %4
br i1 %cmp.i, label %vector.sevcgepcheck, label %region_for_end.i

```

```
vector::sveccheck;
%48 = trunc i64 %46 to i32
%49 = add i32 %23, %48
%50 = trunc i64 %46 to i32
%51 = add i32 i19, %50
%52 = trunc i64 %46 to i32
%53 = add i32 %17, %52
%54 = icmp sgt i32 %53, 2147483616
%55 = icmp sgt i32 %51, 2147483616
%56 = or i1 %54, %55
%57 = icmp sgt i32 %49, 2147483616
%58 = or i1 %56, %57
br i1 %58, label %prejoin_for_entry.entry.i.us.preheader, label
```

```
vector.memcheck:
%bound0 = icmp ugt float* %scevgep15, %2
%bound1 = icmp ugt float* %scevgep10, %0
%found.conflict = and i1 %bound0, %bound1
%bound033 = icmp ult float* %scevgep, %scevgep21
%bound134 = icmp ult float* %scevgep18, %scevgep11
%found.conflict35 = and i1 %bound033, %bound134
%conflict.rdx = or i1 %found.conflict, %found.conflict35
%bound036 = icmp ugt float* %scevgep25, %2
%bound137 = icmp ugt float* %scevgep10, %1
%found.conflict38 = and i1 %bound036, %bound137
%conflict.rdx39 = or i1 %conflict.rdx, %found.conflict38
%bound040 = icmp ult float* %scevgep, %scevgep31
%bound141 = icmp ult float* %scevgep28, %scevgep11
%found.conflict42 = and i1 %bound040, %bound141
%conflict.rdx43 = or i1 %conflict.rdx39, %found.conflict42
br i1 %conflict.rdx43, label %prejoin_for_entry.entry.i.us.preheader, label
%%vector.ph
```

```
vector.ph:
%broadcast.splatinert = insertelement <8 x i64> undef, i64 %i0, i32 0
%broadcast.splat = shufflevector <8 x i64> %broadcast.splatinert, <8 x i64>
... undef, <8 x i32> %zeroinitializer
%broadcast.splatinert44 = insertelement <8 x i32> undef, i32 %sub4.i, i32 0
%broadcast.splat45 = shufflevector <8 x i32> %broadcast.splatinert44, <8 x
... i32> undef, <8 x i32> %zeroinitializer
br label %vector.body
```

```

vector.body:
%index = phi i64 [ 0, %vector.ph ], [ %index.next, %vector.body ]
%vec.ind = phi <8 x i64> [ <i64 0, i64 1, i64 2, i64 3, i64 4, i64 5, i64 6,
... i64 7>, %vector.ph ], [ %vec.ind.next, %vector.body ]
%59 = add nuw nsw <8 x i64> %vec.ind, %broadcast.splat
%60 = trunc <8 x i64> %59 to <8 x i32>
%61 = icmp sgt <8 x i32> %broadcast.splat45, %60
%62 = extractelement <8 x i32> %60, i32 0
%63 = add i32 %mul1, %62
%64 = sext i32 %63 to i64
%65 = getelementptr inbounds float, float* %2, i64 %64
%66 = bitcast float* %65 to <8 x float>*
%wide.masked.load = call <8 x float> @llvm.masked.load.v8f32.p0v8f32i<8 x
... float>* %66, i32 4, <8 x i1> %61, <8 x float> undef, !tbaa !12, !alias.scope
... !16, !inalias !19
%67 = fpext <8 x float> %wide.masked.load to <8 x double>
%68 = add i32 %63, 1
%69 = sext i32 %68 to i64
%70 = getelementptr inbounds float, float* %0, i64 %69
%71 = bitcast float* %70 to <8 x float>*
%wide.masked.load46 = call <8 x float> @llvm.masked.load.v8f32.p0v8f32i<8 x
... float>* %71, i32 4, <8 x i1> %61, <8 x float> undef, !tbaa !12, !alias.scope
... !24
%72 = getelementptr inbounds float, float* %0, i64 %64
%73 = bitcast float* %72 to <8 x float>*
%wide.masked.load47 = call <8 x float> @llvm.masked.load.v8f32.p0v8f32i<8 x
... float>* %73, i32 4, <8 x i1> %61, <8 x float> undef, !tbaa !12, !alias.scope
... !25
%74 = fsub <8 x float> %wide.masked.load46, %wide.masked.load47
%75 = add nsw i32 %mul19.i, %62
%76 = sext i32 %75 to i64
%77 = getelementptr inbounds float, float* %1, i64 %76
%78 = bitcast float* %77 to <8 x float>*
%wide.masked.load48 = call <8 x float> @llvm.masked.load.v8f32.p0v8f32i<8 x
... float>* %78, i32 4, <8 x i1> %61, <8 x float> undef, !tbaa !12, !alias.scope
... !26
%79 = fadd <8 x float> %74, %wide.masked.load48
%80 = getelementptr inbounds float, float* %1, i64 %64
%81 = bitcast float* %80 to <8 x float>*
%wide.masked.load49 = call <8 x float> @llvm.masked.load.v8f32.p0v8f32i<8 x
... float>* %81, i32 4, <8 x i1> %61, <8 x float> undef, !tbaa !12, !alias.scope
... !27
%82 = fsub <8 x float> %79, %wide.masked.load49
%83 = fpext <8 x float> %82 to <8 x double>
%84 = call <8 x double> @llvm.fmuladd.v8f64i<8 x double> %83, <8 x double>
... <double 0xBF6E666666666666, double 0xBF6E666666666666, double
... 0xBF6E666666666666, double 0xBF6E666666666666, double 0xBF6E666666666666,
... double 0xBF6E666666666666, double 0xBF6E666666666666, double
... 0xBF6E666666666666>, <8 x double> %67)
%85 = fptrunc <8 x double> %84 to <8 x float>
%86 = bitcast float* %65 to <8 x float>*
call void @llvm.masked.store.v8f32.p0v8f32i<8 x float> %85, <8 x float>*
... %66, i32 4, <8 x i1> %61, !tbaa !12, !alias.scope !16, !inalias !19,
... !llvm.access_group !28
%index.next = add i64 %index, 8
%vec.ind.next = add <8 x i64> %vec.ind, <i64 8, i64 8, i64 8, i64 8,
... i64 8, i64 8, i64 8>
%87 = icmp eq i64 %index.next, 32
br i1 %87, label %region.for_end.i.loopepit51, label %vector.body,
... !llvm.loop !31

```

```

%region for_entryentry.i.us:
% local id x.0.us = phi i64 [%95,%if.end.r_exit.i.us ], [ 0,
... %region for_entryentry.i.us.preheader ]
%388 = add nuw nsw i64 %local id x.0.us,%10
%conv.i.us = trunc i64 %88 to i32
%cmp5.i.us = icmp slt i32 %sub4.i,%conv.i.us
br i1 %cmp5.i.us,label %if.then.i.us,label %if.end.r_exit.i.us

```

```
if then i.us:
%addi.i.us = add i32 %mul.i, %conv.i.us
%idxprom.i.us = sext i32 %add.i.us to i64
%arrayidx.i.us = getelementptr inbounds float, float* %2, i64 %idxprom.i.us
%89 = load float, float* %arrayidx.i.us, align 4, !tbaa !12
%conv7.i.us = fpxext float %89 to double
%add10.i.us = add i32 %add.i.us, 1
%idxprom11.i.us = sext i32 %add10.i.us to i64
%arrayidx12.i.us = getelementptr inbounds float, float* %0, i64
... %idxprom11.i.us
%90 = load float, float* %arrayidx12.i.us, align 4, !tbaa !12
%arrayidx16.i.us = getelementptr inbounds float, float* %0, i64 %idxprom.i.us
%91 = load float, float* %arrayidx16.i.us, align 4, !tbaa !12
%sub17.i.us = fsub float %90, %91
%add20.i.us = add nsw i32 %mul19.i, %conv.i.us
%idxprom21.i.us = sext i32 %add20.i.us to i64
%arrayidx22.i.us = getelementptr inbounds float, float* %1, i64
... %idxprom21.i.us
%92 = load float, float* %arrayidx22.i.us, align 4, !tbaa !12
%add23.i.us = fadd float %sub17.i.us, %92
%arrayidx27.i.us = getelementptr inbounds float, float* %1, i64 %idxprom.i.us
%93 = load float, float* %arrayidx27.i.us, align 4, !tbaa !12
%sub28.i.us = fsub float %add23.i.us, %93
%conv29.i.us = fpxext float %sub28.i.us to double
%94 = tail call @llvm.fmuladd.f64(double %conv29.i.us, double
... 0xBF6E666666666666, %conv7.i.us) #5
%conv31.i.us = fptrunc double %94 to float
store float %conv31.i.us, float* %arrayidx.i.us, align 4, !tbaa !12,
... !lvm.access_group 128
br label %if.end.r.exit.i.us
```

```
if.end.r_exit.i.us:
%95 = add nuw nsw i64 % local_id_x.0.us, 1
%exitcond = icmp eq i64 %95, 32
br i1 %exitcond, label %pregon for_end.i.loopexit, label
%pregon for_entry.i.us, !llvm.loop !34
```

```
preregion for_end.i.loopexit51:
br label %preregion for_end.i
```

```

pregion_for_end.i:
%96 = add_nuw nsw i64 % local_id_y.0, 1
%exitcond2 = icmp eq i64 %96, 8
br i1 %exitcond2, label %fddt_kernel3.exit, label
    %pregion_for_entry.pregion_for_init.i.llvm.loop.ll

```

```
fddt_kernel3.exit:
    ret void
```