基于多边形三角分割法的近红外成像多边形填充软件

```cpp
#include <iostream>
#include <GL/glut.h>
#include <glm/glm.hpp>
#include <vector>
#include "triangulate.h"

/*---常量---*/
const int HEIGHT(800);
const int WIDTH(800);

/*---全局变量---*/
glm::vec2 StartPoint{ 0, 0 };
glm::vec2 EndPoint{ 0, 0 };
//绘线变量
std::vector<glm::vec2> VertexArray;
//顶点数组

/*---函数---*/
void Display();                                 //mian中显示子程序
void Reshape(int, int);                         //投影变换函数
void MouseMove(int, int);                       //鼠标移动事件
void MouseClick(int, int, int, int);            //鼠标点击事件
void DrawLine(int, int);                        //绘制直线
void DrawEdge();                                //绘制多边形的边
void DrawPolygon();                             //绘制多边形
Vector2dVector PolygonToTriangle();             //剪裁多边形

int main(int argc, char** argv)
{
    glutInit(&argc, argv);
    //GLUT初始化

```

```
33      glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGB);
34      //双缓存模式
35
36      glutInitWindowSize(WIDTH, HEIGHT);
37      glutInitWindowPosition(50, 100);
38      //窗口初始化
39
40      glClearColor(0, 0, 0, 1);
41      //窗口颜色
42
43      glutCreateWindow("Draw and Filled Ploygon");
44
45      glutDisplayFunc(Display);
46      glutMouseFunc(MouseClick);
47      glutReshapeFunc(Reshape);
48
49      glutMainLoop();
50  }
51
52  void Display()
53  {
54      glClear(GL_COLOR_BUFFER_BIT);
55  }
56
57  void Reshape(int w, int h)
58  {
59      glMatrixMode(GL_PROJECTION);
60      glLoadIdentity();
61      gluOrtho2D(0, WIDTH, 0, HEIGHT);
62  }
63
64  void MouseMove(int x, int y)
65  {
66      std::cout << "(" << x << ", " << y << ")" << std::endl;
67
68      int slope = ((y - StartPoint.y) / (x - StartPoint.x));
69      //斜率
70      int Shift = glutGetModifiers();
71      //获取特殊按键
72      glm::vec2 GoalPoint{ 0,0 };
```

```
    if (GLUT_ACTIVE_SHIFT == Shift)
    {
        if (slope == 1)                         //斜率为1，绘制 45°斜线
        {
            GoalPoint.x = x;
            GoalPoint.y = (HEIGHT - x);
        }
        else if (slope == -1)                   //斜率为-1，绘制 -45°斜线
        {
            GoalPoint.x = x;
            GoalPoint.y = x;
        }
        else if ((slope > -1) && (slope < 1))   //斜率在-1到1之间，绘制 x 方向平行线
        {
            GoalPoint.x = x;
            GoalPoint.y = StartPoint.y;
        }
        else if ((slope < -1) || (slope > 1))   //斜率小于-1 或 大于1，绘制 y 方向平行线
        {
            GoalPoint.x = StartPoint.x;
            GoalPoint.y = (HEIGHT - y);
        }
    }
    else if (GLUT_ACTIVE_SHIFT != Shift)
    {
        GoalPoint.x = x;
        GoalPoint.y = (HEIGHT - y);
    }
    DrawLine(GoalPoint.x, GoalPoint.y);
}

void MouseClick(int button, int state, int x, int y)
{
    if (GLUT_DOWN == state && GLUT_LEFT_BUTTON == button)
    {
        //左键按下：动态画线
        if (VertexArray.empty())
        {
```

```
111             StartPoint.x = x;
112             StartPoint.y = (HEIGHT - y);
113             VertexArray.push_back(StartPoint);
114         }
115         //数组空：新建顶点存入数组，否则直接以多边形最后一个点作为顶点
116 
117         StartPoint.x = VertexArray.back().x;
118         StartPoint.y = VertexArray.back().y;
119         glutMotionFunc(MouseMove);
120     }
121     else if (GLUT_UP == state && GLUT_LEFT_BUTTON == button)
122     {
123         int Shift = glutGetModifiers();
124         int slope = ((y - StartPoint.y) / (x - StartPoint.x));
125         if (GLUT_ACTIVE_SHIFT == Shift)
126         {
127             if (slope == 1)                         //斜率为1，绘制 45°斜
线
128             {
129                 EndPoint.x = x;
130                 EndPoint.y = (HEIGHT - x);
131             }
132             else if (slope == -1)                   //斜率为-1，绘制 -45°
斜线
133             {
134                 EndPoint.x = x;
135                 EndPoint.y = x;
136             }
137             else if ((slope > -1) && (slope < 1))   //斜率在-1到1之
间，绘制 x 方向平行线
138             {
139                 EndPoint.x = x;
140                 EndPoint.y = StartPoint.y;
141             }
142             else if ((slope < -1) || (slope > 1))   //斜率小于-1 或
大于1，绘制 y 方向平行线
143             {
144                 EndPoint.x = StartPoint.x;
145                 EndPoint.y = (HEIGHT - y);
146             }
```

```
        }
        else if (GLUT_ACTIVE_SHIFT != Shift)
        {
            EndPoint.x = x;
            EndPoint.y = (HEIGHT - y);
        }
        VertexArray.push_back(EndPoint);
        //终点存入

        StartPoint = glm::vec2{ 0,0 };
        EndPoint = glm::vec2{ 0,0 };
        //Clear

        DrawEdge();
    }
    else if (GLUT_DOWN == state && GLUT_RIGHT_BUTTON == button)
    {
        //右键把线段转换为多边形
        DrawPolygon();
    }
}

void DrawLine(int x, int y)
{
    DrawEdge();

    glColor3f(0, 1.0, 1.0);
    //颜色选择cyan

    glClear(GL_COLOR_BUFFER_BIT);
    //刷新缓冲区

    glLineWidth(1);
    //线条宽度：1个像素

    glBegin(GL_LINES);
    glVertex2i(StartPoint.x, StartPoint.y);
    glVertex2i(x, y);
    glEnd();
    //画线
```

```
187
188     glutSwapBuffers();
189 }
190
191 void DrawEdge()
192 {
193     glColor3f(0, 1.0, 1.0);
194     glClear(GL_COLOR_BUFFER_BIT);
195     glLineWidth(1);
196     //基本设置同上
197
198     glBegin(GL_LINE_STRIP);
199     for (auto i : VertexArray)
200     {
201         glVertex2i(i.x, i.y);
202     }
203     glEnd();
204     //画图
205
206     glutSwapBuffers();
207 }
208
209 void DrawPolygon()
210 {
211     glColor3f(0, 0.5, 1.0);
212     glClear(GL_COLOR_BUFFER_BIT);
213     glLineWidth(1);
214     //基本设置同上
215
216     Vector2dVector TriangleVertexArray = PolygonToTriangle();
217
218     glBegin(GL_TRIANGLES);
219     for (auto i : TriangleVertexArray)
220     {
221         glVertex2i(i.GetX(), i.GetY());
222     }
223     glEnd();
224     //画图
225
226     glutSwapBuffers();
```

```
}

Vector2dVector PolygonToTriangle()
{
    Vector2dVector PolygonVertexArray;
    Vector2dVector TriangleVertexArray;
    for (auto i : VertexArray)
    {
        Vector2d TheVertex(i.x, i.y);
        PolygonVertexArray.push_back(TheVertex);
    }

    Triangulate::Process(PolygonVertexArray, TriangleVertexArray);

    return TriangleVertexArray;
}

#ifndef TRIANGULATE_H

#define TRIANGULATE_H

#include <vector>  // Include STL vector class.

class Vector2d
{
public:
  Vector2d(float x,float y)
  {
    Set(x,y);
  };

  float GetX(void) const { return mX; };

  float GetY(void) const { return mY; };

  void  Set(float x,float y)
  {
    mX = x;
    mY = y;
  };
```

```
private:
  float mX;
  float mY;
};

// Typedef an STL vector of vertices which are used to represent
// a polygon/contour and a series of triangles.
typedef std::vector< Vector2d > Vector2dVector;


class Triangulate
{
public:

  // triangulate a contour/polygon, places results in STL vector
  // as series of triangles.
  static bool Process(const Vector2dVector &contour,
                      Vector2dVector &result);

  // compute area of a contour/polygon
  static float Area(const Vector2dVector &contour);

  // decide if point Px/Py is inside triangle defined by
  // (Ax,Ay) (Bx,By) (Cx,Cy)
  static bool InsideTriangle(float Ax, float Ay,
                      float Bx, float By,
                      float Cx, float Cy,
                      float Px, float Py);


private:
  static bool Snip(const Vector2dVector &contour,int u,int v,int
w,int n,int *V);

};


#endif
```

```
/*********************************************************************
*******/
/*** END OF HEADER FILE TRIANGULATE.H BEGINNING OF CODE
TRIANGULATE.CPP ***/
/*********************************************************************
*******/


#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <assert.h>

#include "triangulate.h"

static const float EPSILON=0.0000000001f;

float Triangulate::Area(const Vector2dVector &contour)
{

  int n = contour.size();

  float A=0.0f;

  for(int p=n-1,q=0; q<n; p=q++)
  {
    A+= contour[p].GetX()*contour[q].GetY() -
contour[q].GetX()*contour[p].GetY();
  }
  return A*0.5f;
}

   /*
     InsideTriangle decides if a point P is Inside of the triangle
     defined by A, B, C.
   */
bool Triangulate::InsideTriangle(float Ax, float Ay,
                      float Bx, float By,
                      float Cx, float Cy,
                      float Px, float Py)
```

```
341
342 {
343   float ax, ay, bx, by, cx, cy, apx, apy, bpx, bpy, cpx, cpy;
344   float cCROSSap, bCROSScp, aCROSSbp;
345
346   ax = Cx - Bx;  ay = Cy - By;
347   bx = Ax - Cx;  by = Ay - Cy;
348   cx = Bx - Ax;  cy = By - Ay;
349   apx= Px - Ax;  apy= Py - Ay;
350   bpx= Px - Bx;  bpy= Py - By;
351   cpx= Px - Cx;  cpy= Py - Cy;
352
353   aCROSSbp = ax*bpy - ay*bpx;
354   cCROSSap = cx*apy - cy*apx;
355   bCROSScp = bx*cpy - by*cpx;
356
357   return ((aCROSSbp >= 0.0f) && (bCROSScp >= 0.0f) && (cCROSSap >=
    0.0f));
358 };
359
360 bool Triangulate::Snip(const Vector2dVector &contour,int u,int v,int
    w,int n,int *V)
361 {
362   int p;
363   float Ax, Ay, Bx, By, Cx, Cy, Px, Py;
364
365   Ax = contour[V[u]].GetX();
366   Ay = contour[V[u]].GetY();
367
368   Bx = contour[V[v]].GetX();
369   By = contour[V[v]].GetY();
370
371   Cx = contour[V[w]].GetX();
372   Cy = contour[V[w]].GetY();
373
374   if ( EPSILON > (((Bx-Ax)*(Cy-Ay)) - ((By-Ay)*(Cx-Ax))) ) return
    false;
375
376   for (p=0;p<n;p++)
377   {
```

```
    if( (p == u) || (p == v) || (p == w) ) continue;
    Px = contour[V[p]].GetX();
    Py = contour[V[p]].GetY();
    if (InsideTriangle(Ax,Ay,Bx,By,Cx,Cy,Px,Py)) return false;
  }

  return true;
}

bool Triangulate::Process(const Vector2dVector
&contour,Vector2dVector &result)
{
  /* allocate and initialize list of Vertices in polygon */

  int n = contour.size();
  if ( n < 3 ) return false;

  int *V = new int[n];

  /* we want a counter-clockwise polygon in V */

  if ( 0.0f < Area(contour) )
    for (int v=0; v<n; v++) V[v] = v;
  else
    for(int v=0; v<n; v++) V[v] = (n-1)-v;

  int nv = n;

  /*  remove nv-2 Vertices, creating 1 triangle every time */
  int count = 2*nv;   /* error detection */

  for(int m=0, v=nv-1; nv>2; )
  {
    /* if we loop, it is probably a non-simple polygon */
    if (0 >= (count--))
    {
      //** Triangulate: ERROR - probable bad polygon!
      return false;
    }

```

```
    /* three consecutive vertices in current polygon, <u,v,w> */
    int u = v  ; if (nv <= u) u = 0;     /* previous */
    v = u+1; if (nv <= v) v = 0;     /* new v    */
    int w = v+1; if (nv <= w) w = 0;     /* next     */

    if ( Snip(contour,u,v,w,nv,V) )
    {
      int a,b,c,s,t;

      /* true names of the vertices */
      a = V[u]; b = V[v]; c = V[w];

      /* output Triangle */
      result.push_back( contour[a] );
      result.push_back( contour[b] );
      result.push_back( contour[c] );

      m++;

      /* remove v from remaining polygon */
      for(s=v,t=v+1;t<nv;s++,t++) V[s] = V[t]; nv--;

      /* resest error detection counter */
      count = 2*nv;
    }
  }



  delete V;

  return true;
}

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <assert.h>
```

```
#include "triangulate.h"

void main(int argc,char **argv)
{

  // Small test application demonstrating the usage of the
triangulate
  // class.


  // Create a pretty complicated little contour by pushing them onto
  // an stl vector.

  Vector2dVector a;

  a.push_back( Vector2d(0,6));
  a.push_back( Vector2d(0,0));
  a.push_back( Vector2d(3,0));
  a.push_back( Vector2d(4,1));
  a.push_back( Vector2d(6,1));
  a.push_back( Vector2d(8,0));
  a.push_back( Vector2d(12,0));
  a.push_back( Vector2d(13,2));
  a.push_back( Vector2d(8,2));
  a.push_back( Vector2d(8,4));
  a.push_back( Vector2d(11,4));
  a.push_back( Vector2d(11,6));
  a.push_back( Vector2d(6,6));
  a.push_back( Vector2d(4,3));
  a.push_back( Vector2d(2,6));

  // allocate an STL vector to hold the answer.

  Vector2dVector result;

  //  Invoke the triangulator to triangulate this polygon.
  Triangulate::Process(a,result);

  // print out the results.
  int tcount = result.size()/3;
```

```
496
497   for (int i=0; i<tcount; i++)
498   {
499     const Vector2d &p1 = result[i*3+0];
500     const Vector2d &p2 = result[i*3+1];
501     const Vector2d &p3 = result[i*3+2];
502     printf("Triangle %d => (%0.0f,%0.0f) (%0.0f,%0.0f)
(%0.0f,%0.0f)\n",i+1,p1.GetX(),p1.GetY(),p2.GetX(),p2.GetY(),p3.GetX
(),p3.GetY());
503   }
504 }
505
506 #include "core/_fixes.hpp"
507
508 #ifndef glm_glm
509 #define glm_glm
510
511 #include <cmath>
512 #include <climits>
513 #include <cfloat>
514 #include <limits>
515 #include <cstdio>
516 //#include <type_traits>
517 #include "core/setup.hpp"
518
519 #if(defined(GLM_MESSAGES) &&
!defined(GLM_MESSAGE_CORE_INCLUDED_DISPLAYED))
520 #   define GLM_MESSAGE_CORE_INCLUDED_DISPLAYED
521 #   pragma message("GLM: Core library included")
522 #endif//GLM_MESSAGE
523
524 #include "./core/_detail.hpp"
525 #include "./core/type.hpp"
526
527 #include "./core/func_trigonometric.hpp"
528 #include "./core/func_exponential.hpp"
529 #include "./core/func_common.hpp"
530 #include "./core/func_packing.hpp"
531 #include "./core/func_geometric.hpp"
532 #include "./core/func_matrix.hpp"
```

```
533 #include "./core/func_vector_relational.hpp"
534 #include "./core/func_integer.hpp"
535 #include "./core/func_noise.hpp"
536 #include "./core/_swizzle.hpp"
537
538 ////////////////////
539 // check type sizes
540 #ifndef GLM_STATIC_ASSERT_NULL
541     GLM_STATIC_ASSERT(sizeof(glm::detail::int8) == 1, "int8 size
    isn't 1 byte on this platform");
542     GLM_STATIC_ASSERT(sizeof(glm::detail::int16) == 2, "int16 size
    isn't 2 bytes on this platform");
543     GLM_STATIC_ASSERT(sizeof(glm::detail::int32) == 4, "int32 size
    isn't 4 bytes on this platform");
544     GLM_STATIC_ASSERT(sizeof(glm::detail::int64) == 8, "int64 size
    isn't 8 bytes on this platform");
545
546     GLM_STATIC_ASSERT(sizeof(glm::detail::uint8) == 1, "uint8 size
    isn't 1 byte on this platform");
547     GLM_STATIC_ASSERT(sizeof(glm::detail::uint16) == 2, "uint16 size
    isn't 2 bytes on this platform");
548     GLM_STATIC_ASSERT(sizeof(glm::detail::uint32) == 4, "uint32 size
    isn't 4 bytes on this platform");
549     GLM_STATIC_ASSERT(sizeof(glm::detail::uint64) == 8, "uint64 size
    isn't 8 bytes on this platform");
550
551     GLM_STATIC_ASSERT(sizeof(glm::detail::float16) == 2, "float16
    size isn't 2 bytes on this platform");
552     GLM_STATIC_ASSERT(sizeof(glm::detail::float32) == 4, "float32
    size isn't 4 bytes on this platform");
553     GLM_STATIC_ASSERT(sizeof(glm::detail::float64) == 8, "float64
    size isn't 8 bytes on this platform");
554 #endif//GLM_STATIC_ASSERT_NULL
555
556 #endif//glm_glm
557
558 #ifndef glm_core_detail
559 #define glm_core_detail
560
561 #include "setup.hpp"
```

```
562 #include <cassert>
563 #if(defined(__STDC_VERSION__) && (__STDC_VERSION__ >= 199901L))
564 #include <cstdint>
565 #endif
566
567 namespace glm{
568 namespace detail
569 {
570     class half;
571
572 #if(defined(__STDC_VERSION__) && (__STDC_VERSION__ >= 199901L)) //
    C99 detected, 64 bit types available
573     typedef int64_t                             sint64;
574     typedef uint64_t                            uint64;
575 #elif(GLM_COMPILER & GLM_COMPILER_VC)
576     typedef signed __int64                      sint64;
577     typedef unsigned __int64                    uint64;
578 #elif(GLM_COMPILER & (GLM_COMPILER_GCC | GLM_COMPILER_LLVM_GCC |
    GLM_COMPILER_CLANG))
579     __extension__ typedef signed long long      sint64;
580     __extension__ typedef unsigned long long    uint64;
581 #elif(GLM_COMPILER & GLM_COMPILER_BC)
582     typedef Int64                               sint64;
583     typedef Uint64                              uint64;
584 #else//unknown compiler
585     typedef signed long long                    sint64;
586     typedef unsigned long long                  uint64;
587 #endif//GLM_COMPILER
588
589     template<bool C>
590     struct If
591     {
592         template<typename F, typename T>
593         static GLM_FUNC_QUALIFIER T apply(F functor, const T& val)
594         {
595             return functor(val);
596         }
597     };
598
599     template<>
```

```
600     struct If<false>
601     {
602         template<typename F, typename T>
603         static GLM_FUNC_QUALIFIER T apply(F, const T& val)
604         {
605             return val;
606         }
607     };
608
609     union uif32
610     {
611         GLM_FUNC_QUALIFIER uif32() :
612             i(0)
613         {}
614
615         GLM_FUNC_QUALIFIER uif32(float f) :
616             f(f)
617         {}
618
619         GLM_FUNC_QUALIFIER uif32(unsigned int i) :
620             i(i)
621         {}
622
623         float f;
624         unsigned int i;
625     };
626
627     union uif64
628     {
629         GLM_FUNC_QUALIFIER uif64() :
630             i(0)
631         {}
632
633         GLM_FUNC_QUALIFIER uif64(double f) :
634             f(f)
635         {}
636
637         GLM_FUNC_QUALIFIER uif64(uint64 i) :
638             i(i)
639         {}
```

```
640
641         double f;
642         uint64 i;
643     };
644
645     typedef uif32 uif;
646
647     /////////////////
648     // int
649
650     template <typename T>
651     struct is_int
652     {
653         enum is_int_enum
654         {
655             _YES = 0,
656             _NO = 1
657         };
658     };
659
660 #define GLM_DETAIL_IS_INT(T)    \
661     template <>                 \
662     struct is_int<T>            \
663     {                           \
664         enum is_int_enum        \
665         {                       \
666             _YES = 1,           \
667             _NO = 0             \
668         };                      \
669     }
670
671     /////////////////
672     // uint
673
674     template <typename T>
675     struct is_uint
676     {
677         enum is_uint_enum
678         {
679             _YES = 0,
```

```
680             _NO = 1
681         };
682     };
683
684 #define GLM_DETAIL_IS_UINT(T)   \
685     template <>                 \
686     struct is_uint<T>           \
687     {                           \
688         enum is_uint_enum       \
689         {                       \
690             _YES = 1,           \
691             _NO = 0             \
692         };                      \
693     }
694
695     //GLM_DETAIL_IS_UINT(unsigned long long)
696
697     //////////////////
698     // float
699
700     template <typename T>
701     struct is_float
702     {
703         enum is_float_enum
704         {
705             _YES = 0,
706             _NO = 1
707         };
708     };
709
710 #define GLM_DETAIL_IS_FLOAT(T)  \
711     template <>                 \
712     struct is_float<T>          \
713     {                           \
714         enum is_float_enum      \
715         {                       \
716             _YES = 1,           \
717             _NO = 0             \
718         };                      \
719     }
```

```
720
721     GLM_DETAIL_IS_FLOAT(detail::half);
722     GLM_DETAIL_IS_FLOAT(float);
723     GLM_DETAIL_IS_FLOAT(double);
724     GLM_DETAIL_IS_FLOAT(long double);
725
726     //////////////////
727     // bool
728
729     template <typename T>
730     struct is_bool
731     {
732         enum is_bool_enum
733         {
734             _YES = 0,
735             _NO = 1
736         };
737     };
738
739     template <>
740     struct is_bool<bool>
741     {
742         enum is_bool_enum
743         {
744             _YES = 1,
745             _NO = 0
746         };
747     };
748
749     //////////////////
750     // vector
751
752     template <typename T>
753     struct is_vector
754     {
755         enum is_vector_enum
756         {
757             _YES = 0,
758             _NO = 1
759         };
```

```
760     };
761
762 #   define GLM_DETAIL_IS_VECTOR(TYPE) \
763         template <typename T> \
764         struct is_vector<TYPE<T> > \
765         { \
766             enum is_vector_enum \
767             { \
768                 _YES = 1, \
769                 _NO = 0 \
770             }; \
771         }
772
773     //////////////////
774     // matrix
775
776     template <typename T>
777     struct is_matrix
778     {
779         enum is_matrix_enum
780         {
781             _YES = 0,
782             _NO = 1
783         };
784     };
785
786 #define GLM_DETAIL_IS_MATRIX(T) \
787     template <>                 \
788     struct is_matrix            \
789     {                           \
790         enum is_matrix_enum     \
791         {                       \
792             _YES = 1,           \
793             _NO = 0             \
794         };                      \
795     }
796
797     //////////////////
798     // type
799
```

```
800     template <typename T>
801     struct type
802     {
803         enum type_enum
804         {
805             is_float = is_float<T>::_YES,
806             is_int = is_int<T>::_YES,
807             is_uint = is_uint<T>::_YES,
808             is_bool = is_bool<T>::_YES
809         };
810     };
811
812     //////////////////
813     // type
814
815     typedef signed char                         int8;
816     typedef signed short                        int16;
817     typedef signed int                          int32;
818     typedef detail::sint64                      int64;
819
820     typedef unsigned char                       uint8;
821     typedef unsigned short                      uint16;
822     typedef unsigned int                        uint32;
823     typedef detail::uint64                      uint64;
824
825     typedef detail::half                        float16;
826     typedef float                               float32;
827     typedef double                              float64;
828
829     //////////////////
830     // float_or_int_trait
831
832     struct float_or_int_value
833     {
834         enum
835         {
836             GLM_ERROR,
837             GLM_FLOAT,
838             GLM_INT
839         };
```

```
840     };
841
842     template <typename T>
843     struct float_or_int_trait
844     {
845         enum{ID = float_or_int_value::GLM_ERROR};
846     };
847
848     template <>
849     struct float_or_int_trait<int8>
850     {
851         enum{ID = float_or_int_value::GLM_INT};
852     };
853
854     template <>
855     struct float_or_int_trait<int16>
856     {
857         enum{ID = float_or_int_value::GLM_INT};
858     };
859
860     template <>
861     struct float_or_int_trait<int32>
862     {
863         enum{ID = float_or_int_value::GLM_INT};
864     };
865
866     template <>
867     struct float_or_int_trait<int64>
868     {
869         enum{ID = float_or_int_value::GLM_INT};
870     };
871
872     template <>
873     struct float_or_int_trait<uint8>
874     {
875         enum{ID = float_or_int_value::GLM_INT};
876     };
877
878     template <>
879     struct float_or_int_trait<uint16>
```

```
880     {
881         enum{ID = float_or_int_value::GLM_INT};
882     };
883
884     template <>
885     struct float_or_int_trait<uint32>
886     {
887         enum{ID = float_or_int_value::GLM_INT};
888     };
889
890     template <>
891     struct float_or_int_trait<uint64>
892     {
893         enum{ID = float_or_int_value::GLM_INT};
894     };
895
896     template <>
897     struct float_or_int_trait<float16>
898     {
899         enum{ID = float_or_int_value::GLM_FLOAT};
900     };
901
902     template <>
903     struct float_or_int_trait<float32>
904     {
905         enum{ID = float_or_int_value::GLM_FLOAT};
906     };
907
908     template <>
909     struct float_or_int_trait<float64>
910     {
911         enum{ID = float_or_int_value::GLM_FLOAT};
912     };
913
914 }//namespace detail
915 }//namespace glm
916
917 #if((GLM_COMPILER & GLM_COMPILER_VC) && (GLM_COMPILER >=
    GLM_COMPILER_VC2005))
918 #   define GLM_DEPRECATED __declspec(deprecated)
```

```
919 #   define GLM_ALIGN(x) __declspec(align(x))
920 #   define GLM_ALIGNED_STRUCT(x) __declspec(align(x)) struct
921 #   define GLM_RESTRICT __declspec(restrict)
922 #   define GLM_RESTRICT_VAR __restrict
923 #elif((GLM_COMPILER & (GLM_COMPILER_GCC | GLM_COMPILER_LLVM_GCC)) &&
    (GLM_COMPILER >= GLM_COMPILER_GCC31))
924 #   define GLM_DEPRECATED __attribute__((__deprecated__))
925 #   define GLM_ALIGN(x) __attribute__((aligned(x)))
926 #   define GLM_ALIGNED_STRUCT(x) struct __attribute__((aligned(x)))
927 #   if(GLM_COMPILER >= GLM_COMPILER_GCC33)
928 #       define GLM_RESTRICT __restrict__
929 #       define GLM_RESTRICT_VAR __restrict__
930 #   else
931 #       define GLM_RESTRICT
932 #       define GLM_RESTRICT_VAR
933 #   endif
934 #   define GLM_RESTRICT __restrict__
935 #   define GLM_RESTRICT_VAR __restrict__
936 #else
937 #   define GLM_DEPRECATED
938 #   define GLM_ALIGN
939 #   define GLM_ALIGNED_STRUCT(x)
940 #   define GLM_RESTRICT
941 #   define GLM_RESTRICT_VAR
942 #endif//GLM_COMPILER
943
944 #endif//glm_core_detail
945
946 #include <cmath>
947
948 //! Workaround for compatibility with other libraries
949 #ifdef max
950 #undef max
951 #endif
952
953 //! Workaround for compatibility with other libraries
954 #ifdef min
955 #undef min
956 #endif
957
```

```
958 //! Workaround for Android
959 #ifdef isnan
960 #undef isnan
961 #endif
962
963 //! Workaround for Android
964 #ifdef isinf
965 #undef isinf
966 #endif
967
968 //! Workaround for Chrone Native Client
969 #ifdef log2
970 #undef log2
971 #endif
972
973 ////////////////////////////////////////////////////////////////
    ///////////////
974 /// OpenGL Mathematics (glm.g-truc.net)
975 ///
976 /// Copyright (c) 2005 - 2012 G-Truc Creation (www.g-truc.net)
977 /// Permission is hereby granted, free of charge, to any person
    obtaining a copy
978 /// of this software and associated documentation files (the
    "Software"), to deal
979 /// in the Software without restriction, including without
    limitation the rights
980 /// to use, copy, modify, merge, publish, distribute, sublicense,
    and/or sell
981 /// copies of the Software, and to permit persons to whom the
    Software is
982 /// furnished to do so, subject to the following conditions:
983 ///
984 /// The above copyright notice and this permission notice shall be
    included in
985 /// all copies or substantial portions of the Software.
986 ///
987 /// THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
    EXPRESS OR
988 /// IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
    MERCHANTABILITY,
```

```
989 /// FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO
    EVENT SHALL THE
990 /// AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR
    OTHER
991 /// LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE,
    ARISING FROM,
992 /// OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER
    DEALINGS IN
993 /// THE SOFTWARE.
994 ///
995 /// @ref core
996 /// @file glm/core/_swizzle.hpp
997 /// @date 2006-04-20 / 2011-02-16
998 /// @author Christophe Riccio
999 ////////////////////////////////////////////////////////////////////
    ///////////////
100
100 #ifndef glm_core_swizzle
101 #define glm_core_swizzle
102
103 #include "_swizzle_func.hpp"
104
105 namespace glm
106 {
107     enum comp
108     {
109         X = 0,
110         R = 0,
111         S = 0,
112         Y = 1,
113         G = 1,
114         T = 1,
115         Z = 2,
116         B = 2,
117         P = 2,
118         W = 3,
119         A = 3,
120         Q = 3
121     };
122 }//namespace glm
```

```
namespace glm{
namespace detail
{
    // Internal class for implementing swizzle operators
    template <typename T, int N>
    struct _swizzle_base0
    {
        typedef T       value_type;

    protected:
        value_type&         elem   (size_t i)       { return (reinterpret_cast<value_type*>(_buffer))[i]; }
        const value_type&   elem   (size_t i) const { return (reinterpret_cast<const value_type*>(_buffer))[i]; }

        // Use an opaque buffer to *ensure* the compiler doesn't call a constructor.
        // The size 1 buffer is assumed to aligned to the actual members so that the
        // elem()
        char    _buffer[1];
    };

    template <typename T, typename V, int E0, int E1, int E2, int E3, int N>
    struct _swizzle_base1 : public _swizzle_base0<T,N>
    {
    };

    template <typename T, typename V, int E0, int E1>
    struct _swizzle_base1<T,V,E0,E1,-1,-2,2> : public _swizzle_base0<T,2>
    {
        V operator ()()  const { return V(this->elem(E0), this->elem(E1)); }
    };

    template <typename T, typename V, int E0, int E1, int E2>
```

```
    struct _swizzle_base1<T,V,E0,E1,E2,-1,3> : public _swizzle_base0<T,3>
    {
        V operator ()()  const { return V(this->elem(E0), this->elem(E1), this->elem(E2)); }
    };

    template <typename T, typename V, int E0, int E1, int E2, int E3>
    struct _swizzle_base1<T,V,E0,E1,E2,E3,4> : public _swizzle_base0<T,4>
    {
        V operator ()()  const { return V(this->elem(E0), this->elem(E1), this->elem(E2), this->elem(E3)); }
    };

    // Internal class for implementing swizzle operators
    /*
        Template parameters:

        ValueType = type of scalar values (e.g. float, double)
        VecType   = class the swizzle is applies to (e.g. tvec3<float>)
        N         = number of components in the vector (e.g. 3)
        E0...3    = what index the n-th element of this swizzle refers to in the unswizzled vec

        DUPLICATE_ELEMENTS = 1 if there is a repeated element, 0 otherwise (used to specialize swizzles
                containing duplicate elements so that they cannot be used as r-values).
    */
    template <typename ValueType, typename VecType, int N, int E0, int E1, int E2, int E3, int DUPLICATE_ELEMENTS>
    struct _swizzle_base2 : public _swizzle_base1<ValueType,VecType,E0,E1,E2,E3,N>
    {
        typedef VecType vec_type;
        typedef ValueType value_type;
```

```
        _swizzle_base2& operator= (const ValueType& t)
        {
            for (int i = 0; i < N; ++i)
                (*this)[i] = t;
            return *this;
        }

        _swizzle_base2& operator= (const VecType& that)
        {
            struct op {
                void operator() (value_type& e, value_type& t) { e = t; }
            };
            _apply_op(that, op());
            return *this;
        }

        void operator -= (const VecType& that)
        {
            struct op {
                void operator() (value_type& e, value_type& t) { e -= t; }
            };
            _apply_op(that, op());
        }

        void operator += (const VecType& that)
        {
            struct op {
                void operator() (value_type& e, value_type& t) { e += t; }
            };
            _apply_op(that, op());
        }

        void operator *= (const VecType& that)
        {
            struct op {
                void operator() (value_type& e, value_type& t) { e *= t; }
```

```
            };
            _apply_op(that, op());
        }

        void operator /= (const VecType& that)
        {
            struct op {
                void operator() (value_type& e, value_type& t) { e
/= t; }
            };
            _apply_op(that, op());
        }

        value_type& operator[]  (size_t i)
        {
            static const int offset_dst[4] = { E0, E1, E2, E3 };
            return this->elem(offset_dst[i]);
        }
        value_type  operator[]  (size_t i) const
        {
            static const int offset_dst[4] = { E0, E1, E2, E3 };
            return this->elem(offset_dst[i]);
        }
    protected:
        template <typename T>
        void _apply_op(const VecType& that, T op)
        {
            // Make a copy of the data in this == &that.
            // The copier should optimize out the copy in cases
where the function is
            // properly inlined and the copy is not necessary.
            ValueType t[N];
            for (int i = 0; i < N; ++i)
                t[i] = that[i];
            for (int i = 0; i < N; ++i)
                op( (*this)[i], t[i] );
        }
    };
```

```
    // Specialization for swizzles containing duplicate elements. These cannot be modified.
    template <typename ValueType, typename VecType, int N, int E0, int E1, int E2, int E3>
    struct _swizzle_base2<ValueType,VecType,N,E0,E1,E2,E3,1> : public _swizzle_base1<ValueType,VecType,E0,E1,E2,E3,N>
    {
        typedef VecType         vec_type;
        typedef ValueType       value_type;

        struct Stub {};
        _swizzle_base2& operator= (Stub const &) {}

        value_type  operator[]  (size_t i) const
        {
            static const int offset_dst[4] = { E0, E1, E2, E3 };
            return this->elem(offset_dst[i]);
        }
    };

    template <int N,typename ValueType, typename VecType, int E0,int E1,int E2,int E3>
    struct swizzle : public _swizzle_base2<ValueType,VecType,N,E0,E1,E2,E3,
    (E0==E1||E0==E2||E0==E3||E1==E2||E1==E3||E2==E3)>
    {
        typedef _swizzle_base2<ValueType,VecType,N,E0,E1,E2,E3,(E0==E1||E0==E2||E0==E3||E1==E2||E1==E3||E2==E3)> base_type;

        using base_type::operator=;

        operator VecType () const { return (*this)(); }
    };

//
// To prevent the C++ syntax from getting entirely overwhelming, define some alias macros
//
#define _GLM_SWIZZLE_TEMPLATE1   template <int N, typename T, typename V, int E0, int E1, int E2, int E3>
```

```
118 #define _GLM_SWIZZLE_TEMPLATE2   template <int N, typename T,
  9 typename V, int E0, int E1, int E2, int E3, int F0, int F1, int F2,
    int F3>
119 #define _GLM_SWIZZLE_TYPE1
  0 glm::detail::swizzle<N,T,V,E0,E1,E2,E3>
119 #define _GLM_SWIZZLE_TYPE2
  1 glm::detail::swizzle<N,T,V,F0,F1,F2,F3>
119
112 //
113 // Wrapper for a binary operator (e.g. u.yy + v.zy)
114 //
115 #define _GLM_SWIZZLE_VECTOR_BINARY_OPERATOR_IMPLEMENTATION(OPERAND)
  6                 \
119     _GLM_SWIZZLE_TEMPLATE2
  7             \
119     V operator OPERAND ( const _GLM_SWIZZLE_TYPE1& a, const
  8 _GLM_SWIZZLE_TYPE2& b)  \
119     {
  9                 \
120         return a() OPERAND b();
  0                 \
120     }
  1                 \
120     _GLM_SWIZZLE_TEMPLATE1
  2             \
120     V operator OPERAND ( const _GLM_SWIZZLE_TYPE1& a, const V& b)
  3                 \
120     {
  4                 \
120         return a() OPERAND b;
  5                 \
120     }
  6                 \
120     _GLM_SWIZZLE_TEMPLATE1
  7             \
120     V operator OPERAND ( const V& a, const _GLM_SWIZZLE_TYPE1& b)
  8                 \
120     {
  9                 \
```

```
        return a OPERAND b();
                \
    }

//
// Wrapper for a operand between a swizzle and a binary (e.g. 1.0f - u.xyz)
//
#define _GLM_SWIZZLE_SCALAR_BINARY_OPERATOR_IMPLEMENTATION(OPERAND)
                \
    _GLM_SWIZZLE_TEMPLATE1
                \
    V operator OPERAND ( const _GLM_SWIZZLE_TYPE1& a, const T& b)
                \
    {
                \
        return a() OPERAND b;
                \
    }
                \
    _GLM_SWIZZLE_TEMPLATE1
                \
    V operator OPERAND ( const T& a, const _GLM_SWIZZLE_TYPE1& b)
                \
    {
                \
        return a OPERAND b();
                \
    }

//
// Macro for wrapping a function taking one argument (e.g. abs())
//
#define _GLM_SWIZZLE_FUNCTION_1_ARGS(RETURN_TYPE,FUNCTION)
                \
    _GLM_SWIZZLE_TEMPLATE1
                \
    typename _GLM_SWIZZLE_TYPE1::RETURN_TYPE FUNCTION(const _GLM_SWIZZLE_TYPE1& a)  \
```

```
123     {
  4                 \
123         return FUNCTION(a());
  5                 \
123     }
128
127 //
128 // Macro for wrapping a function taking two vector arguments (e.g.
  9 dot()).
124 //
120 #define _GLM_SWIZZLE_FUNCTION_2_ARGS(RETURN_TYPE,FUNCTION)
  1                                                   \
124     _GLM_SWIZZLE_TEMPLATE2
  2                                                   \
124     typename _GLM_SWIZZLE_TYPE1::RETURN_TYPE FUNCTION(const
  3 _GLM_SWIZZLE_TYPE1& a, const _GLM_SWIZZLE_TYPE2& b) \
124     {
  4                                                    \
124         return FUNCTION(a(), b());
  5                                                   \
124     }
  6                                                    \
124     _GLM_SWIZZLE_TEMPLATE1
  7                                                   \
124     typename _GLM_SWIZZLE_TYPE1::RETURN_TYPE FUNCTION(const
  8 _GLM_SWIZZLE_TYPE1& a, const _GLM_SWIZZLE_TYPE1& b) \
124     {
  9                                                    \
125         return FUNCTION(a(), b());
  0                                                   \
125     }
  1                                                    \
125     _GLM_SWIZZLE_TEMPLATE1
  2                                                   \
125     typename _GLM_SWIZZLE_TYPE1::RETURN_TYPE FUNCTION(const
  3 _GLM_SWIZZLE_TYPE1& a, const typename V& b)         \
125     {
  4                                                    \
125         return FUNCTION(a(), b);
  5                                                   \
```

```
125     }
  6                                                   \
125     _GLM_SWIZZLE_TEMPLATE1
  7                                                  \
125     typename _GLM_SWIZZLE_TYPE1::RETURN_TYPE FUNCTION(const V& a,
  8 const _GLM_SWIZZLE_TYPE1& b)                     \
125     {
  9                                                   \
126         return FUNCTION(a, b());
  0                                                  \
126     }
126
126 //
126 // Macro for wrapping a function take 2 vec arguments followed by a
  4 scalar (e.g. mix()).
126 //
126 #define _GLM_SWIZZLE_FUNCTION_2_ARGS_SCALAR(RETURN_TYPE,FUNCTION)
  6                                                                  \
126     _GLM_SWIZZLE_TEMPLATE2
  7                                                                 \
126     typename _GLM_SWIZZLE_TYPE1::RETURN_TYPE FUNCTION(const
  8 _GLM_SWIZZLE_TYPE1& a, const _GLM_SWIZZLE_TYPE2& b, const T& c)   \
126     {
  9                                                                  \
127         return FUNCTION(a(), b(), c);
  0                                                                  \
127     }
  1                                                                  \
127     _GLM_SWIZZLE_TEMPLATE1
  2                                                                 \
127     typename _GLM_SWIZZLE_TYPE1::RETURN_TYPE FUNCTION(const
  3 _GLM_SWIZZLE_TYPE1& a, const _GLM_SWIZZLE_TYPE1& b, const T& c)   \
127     {
  4                                                                  \
127         return FUNCTION(a(), b(), c);
  5                                                                  \
127     }
  6                                                                  \
127     _GLM_SWIZZLE_TEMPLATE1
  7                                                                 \
```

```
127     typename _GLM_SWIZZLE_TYPE1::RETURN_TYPE FUNCTION(const
  8 _GLM_SWIZZLE_TYPE1& a, const typename S0::vec_type& b, const T& c)\
127     {
  9                                                                   \
128         return FUNCTION(a(), b, c);
  0                                                                   \
128     }
  1                                                                   \
128     _GLM_SWIZZLE_TEMPLATE1
  2                                                                  \
128     typename _GLM_SWIZZLE_TYPE1::RETURN_TYPE FUNCTION(const typename
  3 V& a, const _GLM_SWIZZLE_TYPE1& b, const T& c)           \
128     {
  4                                                                   \
128         return FUNCTION(a, b(), c);
  5                                                                   \
128     }
126
127 }//namespace detail
128 }//namespace glm
129
129 namespace glm
131 {
132     namespace detail
133     {
134         _GLM_SWIZZLE_SCALAR_BINARY_OPERATOR_IMPLEMENTATION(-)
135         _GLM_SWIZZLE_SCALAR_BINARY_OPERATOR_IMPLEMENTATION(*)
136
137         _GLM_SWIZZLE_VECTOR_BINARY_OPERATOR_IMPLEMENTATION(+)
138         _GLM_SWIZZLE_VECTOR_BINARY_OPERATOR_IMPLEMENTATION(-)
139         _GLM_SWIZZLE_VECTOR_BINARY_OPERATOR_IMPLEMENTATION(*)
130         _GLM_SWIZZLE_VECTOR_BINARY_OPERATOR_IMPLEMENTATION(/)
131     }
132
133     //
134     // Swizzles are distinct types from the unswizzled type.  The
  5 below macros will
130     // provide template specializations for the swizzle types for
  6 the given functions
```

```
    // so that the compiler does not have any ambiguity to choosing how to handle
    // the function.
    //
    // The alternative is to use the operator()() when calling the function in order
    // to explicitly convert the swizzled type to the unswizzled type.
    //

    //_GLM_SWIZZLE_FUNCTION_1_ARGS(vec_type,    abs);
    //_GLM_SWIZZLE_FUNCTION_1_ARGS(vec_type,    acos);
    //_GLM_SWIZZLE_FUNCTION_1_ARGS(vec_type,    acosh);
    //_GLM_SWIZZLE_FUNCTION_1_ARGS(vec_type,    all);
    //_GLM_SWIZZLE_FUNCTION_1_ARGS(vec_type,    any);

    //_GLM_SWIZZLE_FUNCTION_2_ARGS(value_type,  dot);
    //_GLM_SWIZZLE_FUNCTION_2_ARGS(vec_type,    cross);
    //_GLM_SWIZZLE_FUNCTION_2_ARGS(vec_type,    step);
    //_GLM_SWIZZLE_FUNCTION_2_ARGS_SCALAR(vec_type, mix);
}

#define _GLM_SWIZZLE2_2_MEMBERS(T,P,E0,E1) \
    struct { glm::detail::swizzle<2,T,P,0,0,-1,-2> E0 ## E0; }; \
    struct { glm::detail::swizzle<2,T,P,0,1,-1,-2> E0 ## E1; }; \
    struct { glm::detail::swizzle<2,T,P,1,0,-1,-2> E1 ## E0; }; \
    struct { glm::detail::swizzle<2,T,P,1,1,-1,-2> E1 ## E1; };

#define _GLM_SWIZZLE2_3_MEMBERS(T,P2,E0,E1) \
    struct { glm::detail::swizzle<3,T,P2,0,0,0,-1> E0 ## E0 ## E0; }; \
    struct { glm::detail::swizzle<3,T,P2,0,0,1,-1> E0 ## E0 ## E1; }; \
    struct { glm::detail::swizzle<3,T,P2,0,1,0,-1> E0 ## E1 ## E0; }; \
    struct { glm::detail::swizzle<3,T,P2,0,1,1,-1> E0 ## E1 ## E1; }; \
    struct { glm::detail::swizzle<3,T,P2,1,0,0,-1> E1 ## E0 ## E0; }; \
```

```
133     struct { glm::detail::swizzle<3,T,P2,1,0,1,-1> E1 ## E0 ## E1;
 8 }; \
133     struct { glm::detail::swizzle<3,T,P2,1,1,0,-1> E1 ## E1 ## E0;
 9 }; \
134     struct { glm::detail::swizzle<3,T,P2,1,1,1,-1> E1 ## E1 ## E1;
 0 };
134
134 #define _GLM_SWIZZLE2_4_MEMBERS(T,P2,E0,E1) \
132     struct { glm::detail::swizzle<4,T,P2,0,0,0,0> E0 ## E0 ## E0 ##
 3 E0; }; \
134     struct { glm::detail::swizzle<4,T,P2,0,0,0,1> E0 ## E0 ## E0 ##
 4 E1; }; \
134     struct { glm::detail::swizzle<4,T,P2,0,0,1,0> E0 ## E0 ## E1 ##
 5 E0; }; \
134     struct { glm::detail::swizzle<4,T,P2,0,0,1,1> E0 ## E0 ## E1 ##
 6 E1; }; \
134     struct { glm::detail::swizzle<4,T,P2,0,1,0,0> E0 ## E1 ## E0 ##
 7 E0; }; \
134     struct { glm::detail::swizzle<4,T,P2,0,1,0,1> E0 ## E1 ## E0 ##
 8 E1; }; \
134     struct { glm::detail::swizzle<4,T,P2,0,1,1,0> E0 ## E1 ## E1 ##
 9 E0; }; \
135     struct { glm::detail::swizzle<4,T,P2,0,1,1,1> E0 ## E1 ## E1 ##
 0 E1; }; \
135     struct { glm::detail::swizzle<4,T,P2,1,0,0,0> E1 ## E0 ## E0 ##
 1 E0; }; \
135     struct { glm::detail::swizzle<4,T,P2,1,0,0,1> E1 ## E0 ## E0 ##
 2 E1; }; \
135     struct { glm::detail::swizzle<4,T,P2,1,0,1,0> E1 ## E0 ## E1 ##
 3 E0; }; \
135     struct { glm::detail::swizzle<4,T,P2,1,0,1,1> E1 ## E0 ## E1 ##
 4 E1; }; \
135     struct { glm::detail::swizzle<4,T,P2,1,1,0,0> E1 ## E1 ## E0 ##
 5 E0; }; \
135     struct { glm::detail::swizzle<4,T,P2,1,1,0,1> E1 ## E1 ## E0 ##
 6 E1; }; \
135     struct { glm::detail::swizzle<4,T,P2,1,1,1,0> E1 ## E1 ## E1 ##
 7 E0; }; \
135     struct { glm::detail::swizzle<4,T,P2,1,1,1,1> E1 ## E1 ## E1 ##
 8 E1; };
```

```
135
130 #define _GLM_SWIZZLE3_2_MEMBERS(T,P2,E0,E1,E2) \
130     struct { glm::detail::swizzle<2,T,P2,0,0,-1,-2> E0 ## E0; }; \
131     struct { glm::detail::swizzle<2,T,P2,0,1,-1,-2> E0 ## E1; }; \
132     struct { glm::detail::swizzle<2,T,P2,0,2,-1,-2> E0 ## E2; }; \
133     struct { glm::detail::swizzle<2,T,P2,1,0,-1,-2> E1 ## E0; }; \
134     struct { glm::detail::swizzle<2,T,P2,1,1,-1,-2> E1 ## E1; }; \
135     struct { glm::detail::swizzle<2,T,P2,1,2,-1,-2> E1 ## E2; }; \
136     struct { glm::detail::swizzle<2,T,P2,2,0,-1,-2> E2 ## E0; }; \
137     struct { glm::detail::swizzle<2,T,P2,2,1,-1,-2> E2 ## E1; }; \
138     struct { glm::detail::swizzle<2,T,P2,2,2,-1,-2> E2 ## E2; };
139
130 #define _GLM_SWIZZLE3_3_MEMBERS(T,P,E0,E1,E2) \
131     struct { glm::detail::swizzle<3,T,P,0,0,0,-1> E0 ## E0 ## E0; };
  2 \
137     struct { glm::detail::swizzle<3,T,P,0,0,1,-1> E0 ## E0 ## E1; };
  3 \
137     struct { glm::detail::swizzle<3,T,P,0,0,2,-1> E0 ## E0 ## E2; };
  4 \
137     struct { glm::detail::swizzle<3,T,P,0,1,0,-1> E0 ## E1 ## E0; };
  5 \
137     struct { glm::detail::swizzle<3,T,P,0,1,1,-1> E0 ## E1 ## E1; };
  6 \
137     struct { glm::detail::swizzle<3,T,P,0,1,2,-1> E0 ## E1 ## E2; };
  7 \
137     struct { glm::detail::swizzle<3,T,P,0,2,0,-1> E0 ## E2 ## E0; };
  8 \
137     struct { glm::detail::swizzle<3,T,P,0,2,1,-1> E0 ## E2 ## E1; };
  9 \
138     struct { glm::detail::swizzle<3,T,P,0,2,2,-1> E0 ## E2 ## E2; };
  0 \
138     struct { glm::detail::swizzle<3,T,P,1,0,0,-1> E1 ## E0 ## E0; };
  1 \
138     struct { glm::detail::swizzle<3,T,P,1,0,1,-1> E1 ## E0 ## E1; };
  2 \
138     struct { glm::detail::swizzle<3,T,P,1,0,2,-1> E1 ## E0 ## E2; };
  3 \
138     struct { glm::detail::swizzle<3,T,P,1,1,0,-1> E1 ## E1 ## E0; };
  4 \
```

```
138     struct { glm::detail::swizzle<3,T,P,1,1,1,-1> E1 ## E1 ## E1; };
  5 \
138     struct { glm::detail::swizzle<3,T,P,1,1,2,-1> E1 ## E1 ## E2; };
  6 \
138     struct { glm::detail::swizzle<3,T,P,1,2,0,-1> E1 ## E2 ## E0; };
  7 \
138     struct { glm::detail::swizzle<3,T,P,1,2,1,-1> E1 ## E2 ## E1; };
  8 \
138     struct { glm::detail::swizzle<3,T,P,1,2,2,-1> E1 ## E2 ## E2; };
  9 \
139     struct { glm::detail::swizzle<3,T,P,2,0,0,-1> E2 ## E0 ## E0; };
  0 \
139     struct { glm::detail::swizzle<3,T,P,2,0,1,-1> E2 ## E0 ## E1; };
  1 \
139     struct { glm::detail::swizzle<3,T,P,2,0,2,-1> E2 ## E0 ## E2; };
  2 \
139     struct { glm::detail::swizzle<3,T,P,2,1,0,-1> E2 ## E1 ## E0; };
  3 \
139     struct { glm::detail::swizzle<3,T,P,2,1,1,-1> E2 ## E1 ## E1; };
  4 \
139     struct { glm::detail::swizzle<3,T,P,2,1,2,-1> E2 ## E1 ## E2; };
  5 \
139     struct { glm::detail::swizzle<3,T,P,2,2,0,-1> E2 ## E2 ## E0; };
  6 \
139     struct { glm::detail::swizzle<3,T,P,2,2,1,-1> E2 ## E2 ## E1; };
  7 \
139     struct { glm::detail::swizzle<3,T,P,2,2,2,-1> E2 ## E2 ## E2; };
138 
140 #define _GLM_SWIZZLE3_4_MEMBERS(T,P2,E0,E1,E2) \
140     struct { glm::detail::swizzle<4,T,P2,0,0,0,0> E0 ## E0 ## E0 ##
  1 E0; }; \
140     struct { glm::detail::swizzle<4,T,P2,0,0,0,1> E0 ## E0 ## E0 ##
  2 E1; }; \
140     struct { glm::detail::swizzle<4,T,P2,0,0,0,2> E0 ## E0 ## E0 ##
  3 E2; }; \
140     struct { glm::detail::swizzle<4,T,P2,0,0,1,0> E0 ## E0 ## E1 ##
  4 E0; }; \
140     struct { glm::detail::swizzle<4,T,P2,0,0,1,1> E0 ## E0 ## E1 ##
  5 E1; }; \
```

```
140     struct { glm::detail::swizzle<4,T,P2,0,0,1,2> E0 ## E0 ## E1 ##
  6 E2; }; \
140     struct { glm::detail::swizzle<4,T,P2,0,0,2,0> E0 ## E0 ## E2 ##
  7 E0; }; \
140     struct { glm::detail::swizzle<4,T,P2,0,0,2,1> E0 ## E0 ## E2 ##
  8 E1; }; \
140     struct { glm::detail::swizzle<4,T,P2,0,0,2,2> E0 ## E0 ## E2 ##
  9 E2; }; \
141     struct { glm::detail::swizzle<4,T,P2,0,1,0,0> E0 ## E1 ## E0 ##
  0 E0; }; \
141     struct { glm::detail::swizzle<4,T,P2,0,1,0,1> E0 ## E1 ## E0 ##
  1 E1; }; \
141     struct { glm::detail::swizzle<4,T,P2,0,1,0,2> E0 ## E1 ## E0 ##
  2 E2; }; \
141     struct { glm::detail::swizzle<4,T,P2,0,1,1,0> E0 ## E1 ## E1 ##
  3 E0; }; \
141     struct { glm::detail::swizzle<4,T,P2,0,1,1,1> E0 ## E1 ## E1 ##
  4 E1; }; \
141     struct { glm::detail::swizzle<4,T,P2,0,1,1,2> E0 ## E1 ## E1 ##
  5 E2; }; \
141     struct { glm::detail::swizzle<4,T,P2,0,1,2,0> E0 ## E1 ## E2 ##
  6 E0; }; \
141     struct { glm::detail::swizzle<4,T,P2,0,1,2,1> E0 ## E1 ## E2 ##
  7 E1; }; \
141     struct { glm::detail::swizzle<4,T,P2,0,1,2,2> E0 ## E1 ## E2 ##
  8 E2; }; \
141     struct { glm::detail::swizzle<4,T,P2,0,2,0,0> E0 ## E2 ## E0 ##
  9 E0; }; \
142     struct { glm::detail::swizzle<4,T,P2,0,2,0,1> E0 ## E2 ## E0 ##
  0 E1; }; \
142     struct { glm::detail::swizzle<4,T,P2,0,2,0,2> E0 ## E2 ## E0 ##
  1 E2; }; \
142     struct { glm::detail::swizzle<4,T,P2,0,2,1,0> E0 ## E2 ## E1 ##
  2 E0; }; \
142     struct { glm::detail::swizzle<4,T,P2,0,2,1,1> E0 ## E2 ## E1 ##
  3 E1; }; \
142     struct { glm::detail::swizzle<4,T,P2,0,2,1,2> E0 ## E2 ## E1 ##
  4 E2; }; \
142     struct { glm::detail::swizzle<4,T,P2,0,2,2,0> E0 ## E2 ## E2 ##
  5 E0; }; \
```

```
142     struct { glm::detail::swizzle<4,T,P2,0,2,2,1> E0 ## E2 ## E2 ##
  6 E1; }; \
142     struct { glm::detail::swizzle<4,T,P2,0,2,2,2> E0 ## E2 ## E2 ##
  7 E2; }; \
142     struct { glm::detail::swizzle<4,T,P2,1,0,0,0> E1 ## E0 ## E0 ##
  8 E0; }; \
142     struct { glm::detail::swizzle<4,T,P2,1,0,0,1> E1 ## E0 ## E0 ##
  9 E1; }; \
143     struct { glm::detail::swizzle<4,T,P2,1,0,0,2> E1 ## E0 ## E0 ##
  0 E2; }; \
143     struct { glm::detail::swizzle<4,T,P2,1,0,1,0> E1 ## E0 ## E1 ##
  1 E0; }; \
143     struct { glm::detail::swizzle<4,T,P2,1,0,1,1> E1 ## E0 ## E1 ##
  2 E1; }; \
143     struct { glm::detail::swizzle<4,T,P2,1,0,1,2> E1 ## E0 ## E1 ##
  3 E2; }; \
143     struct { glm::detail::swizzle<4,T,P2,1,0,2,0> E1 ## E0 ## E2 ##
  4 E0; }; \
143     struct { glm::detail::swizzle<4,T,P2,1,0,2,1> E1 ## E0 ## E2 ##
  5 E1; }; \
143     struct { glm::detail::swizzle<4,T,P2,1,0,2,2> E1 ## E0 ## E2 ##
  6 E2; }; \
143     struct { glm::detail::swizzle<4,T,P2,1,1,0,0> E1 ## E1 ## E0 ##
  7 E0; }; \
143     struct { glm::detail::swizzle<4,T,P2,1,1,0,1> E1 ## E1 ## E0 ##
  8 E1; }; \
143     struct { glm::detail::swizzle<4,T,P2,1,1,0,2> E1 ## E1 ## E0 ##
  9 E2; }; \
144     struct { glm::detail::swizzle<4,T,P2,1,1,1,0> E1 ## E1 ## E1 ##
  0 E0; }; \
144     struct { glm::detail::swizzle<4,T,P2,1,1,1,1> E1 ## E1 ## E1 ##
  1 E1; }; \
144     struct { glm::detail::swizzle<4,T,P2,1,1,1,2> E1 ## E1 ## E1 ##
  2 E2; }; \
144     struct { glm::detail::swizzle<4,T,P2,1,1,2,0> E1 ## E1 ## E2 ##
  3 E0; }; \
144     struct { glm::detail::swizzle<4,T,P2,1,1,2,1> E1 ## E1 ## E2 ##
  4 E1; }; \
144     struct { glm::detail::swizzle<4,T,P2,1,1,2,2> E1 ## E1 ## E2 ##
  5 E2; }; \
```

```
144     struct { glm::detail::swizzle<4,T,P2,1,2,0,0> E1 ## E2 ## E0 ##
  6 E0; }; \
144     struct { glm::detail::swizzle<4,T,P2,1,2,0,1> E1 ## E2 ## E0 ##
  7 E1; }; \
144     struct { glm::detail::swizzle<4,T,P2,1,2,0,2> E1 ## E2 ## E0 ##
  8 E2; }; \
144     struct { glm::detail::swizzle<4,T,P2,1,2,1,0> E1 ## E2 ## E1 ##
  9 E0; }; \
145     struct { glm::detail::swizzle<4,T,P2,1,2,1,1> E1 ## E2 ## E1 ##
  0 E1; }; \
145     struct { glm::detail::swizzle<4,T,P2,1,2,1,2> E1 ## E2 ## E1 ##
  1 E2; }; \
145     struct { glm::detail::swizzle<4,T,P2,1,2,2,0> E1 ## E2 ## E2 ##
  2 E0; }; \
145     struct { glm::detail::swizzle<4,T,P2,1,2,2,1> E1 ## E2 ## E2 ##
  3 E1; }; \
145     struct { glm::detail::swizzle<4,T,P2,1,2,2,2> E1 ## E2 ## E2 ##
  4 E2; }; \
145     struct { glm::detail::swizzle<4,T,P2,2,0,0,0> E2 ## E0 ## E0 ##
  5 E0; }; \
145     struct { glm::detail::swizzle<4,T,P2,2,0,0,1> E2 ## E0 ## E0 ##
  6 E1; }; \
145     struct { glm::detail::swizzle<4,T,P2,2,0,0,2> E2 ## E0 ## E0 ##
  7 E2; }; \
145     struct { glm::detail::swizzle<4,T,P2,2,0,1,0> E2 ## E0 ## E1 ##
  8 E0; }; \
145     struct { glm::detail::swizzle<4,T,P2,2,0,1,1> E2 ## E0 ## E1 ##
  9 E1; }; \
146     struct { glm::detail::swizzle<4,T,P2,2,0,1,2> E2 ## E0 ## E1 ##
  0 E2; }; \
146     struct { glm::detail::swizzle<4,T,P2,2,0,2,0> E2 ## E0 ## E2 ##
  1 E0; }; \
146     struct { glm::detail::swizzle<4,T,P2,2,0,2,1> E2 ## E0 ## E2 ##
  2 E1; }; \
146     struct { glm::detail::swizzle<4,T,P2,2,0,2,2> E2 ## E0 ## E2 ##
  3 E2; }; \
146     struct { glm::detail::swizzle<4,T,P2,2,1,0,0> E2 ## E1 ## E0 ##
  4 E0; }; \
146     struct { glm::detail::swizzle<4,T,P2,2,1,0,1> E2 ## E1 ## E0 ##
  5 E1; }; \
```

```
146     struct { glm::detail::swizzle<4,T,P2,2,1,0,2> E2 ## E1 ## E0 ##
  6 E2; }; \
146     struct { glm::detail::swizzle<4,T,P2,2,1,1,0> E2 ## E1 ## E1 ##
  7 E0; }; \
146     struct { glm::detail::swizzle<4,T,P2,2,1,1,1> E2 ## E1 ## E1 ##
  8 E1; }; \
146     struct { glm::detail::swizzle<4,T,P2,2,1,1,2> E2 ## E1 ## E1 ##
  9 E2; }; \
147     struct { glm::detail::swizzle<4,T,P2,2,1,2,0> E2 ## E1 ## E2 ##
  0 E0; }; \
147     struct { glm::detail::swizzle<4,T,P2,2,1,2,1> E2 ## E1 ## E2 ##
  1 E1; }; \
147     struct { glm::detail::swizzle<4,T,P2,2,1,2,2> E2 ## E1 ## E2 ##
  2 E2; }; \
147     struct { glm::detail::swizzle<4,T,P2,2,2,0,0> E2 ## E2 ## E0 ##
  3 E0; }; \
147     struct { glm::detail::swizzle<4,T,P2,2,2,0,1> E2 ## E2 ## E0 ##
  4 E1; }; \
147     struct { glm::detail::swizzle<4,T,P2,2,2,0,2> E2 ## E2 ## E0 ##
  5 E2; }; \
147     struct { glm::detail::swizzle<4,T,P2,2,2,1,0> E2 ## E2 ## E1 ##
  6 E0; }; \
147     struct { glm::detail::swizzle<4,T,P2,2,2,1,1> E2 ## E2 ## E1 ##
  7 E1; }; \
147     struct { glm::detail::swizzle<4,T,P2,2,2,1,2> E2 ## E2 ## E1 ##
  8 E2; }; \
147     struct { glm::detail::swizzle<4,T,P2,2,2,2,0> E2 ## E2 ## E2 ##
  9 E0; }; \
148     struct { glm::detail::swizzle<4,T,P2,2,2,2,1> E2 ## E2 ## E2 ##
  0 E1; }; \
148     struct { glm::detail::swizzle<4,T,P2,2,2,2,2> E2 ## E2 ## E2 ##
  1 E2; };
148
148 #define _GLM_SWIZZLE4_2_MEMBERS(T,P,E0,E1,E2,E3) \
148     struct { glm::detail::swizzle<2,T,P,0,0,-1,-2> E0 ## E0; }; \
148     struct { glm::detail::swizzle<2,T,P,0,1,-1,-2> E0 ## E1; }; \
148     struct { glm::detail::swizzle<2,T,P,0,2,-1,-2> E0 ## E2; }; \
148     struct { glm::detail::swizzle<2,T,P,0,3,-1,-2> E0 ## E3; }; \
148     struct { glm::detail::swizzle<2,T,P,1,0,-1,-2> E1 ## E0; }; \
148     struct { glm::detail::swizzle<2,T,P,1,1,-1,-2> E1 ## E1; }; \
```

```
    struct { glm::detail::swizzle<2,T,P,1,2,-1,-2> E1 ## E2; }; \
    struct { glm::detail::swizzle<2,T,P,1,3,-1,-2> E1 ## E3; }; \
    struct { glm::detail::swizzle<2,T,P,2,0,-1,-2> E2 ## E0; }; \
    struct { glm::detail::swizzle<2,T,P,2,1,-1,-2> E2 ## E1; }; \
    struct { glm::detail::swizzle<2,T,P,2,2,-1,-2> E2 ## E2; }; \
    struct { glm::detail::swizzle<2,T,P,2,3,-1,-2> E2 ## E3; }; \
    struct { glm::detail::swizzle<2,T,P,3,0,-1,-2> E3 ## E0; }; \
    struct { glm::detail::swizzle<2,T,P,3,1,-1,-2> E3 ## E1; }; \
    struct { glm::detail::swizzle<2,T,P,3,2,-1,-2> E3 ## E2; }; \
    struct { glm::detail::swizzle<2,T,P,3,3,-1,-2> E3 ## E3; };

#define _GLM_SWIZZLE4_3_MEMBERS(T,P,E0,E1,E2,E3) \
    struct { glm::detail::swizzle<3,T,P,0,0,0,-1> E0 ## E0 ## E0; }; \
    struct { glm::detail::swizzle<3,T,P,0,0,1,-1> E0 ## E0 ## E1; }; \
    struct { glm::detail::swizzle<3,T,P,0,0,2,-1> E0 ## E0 ## E2; }; \
    struct { glm::detail::swizzle<3,T,P,0,0,3,-1> E0 ## E0 ## E3; }; \
    struct { glm::detail::swizzle<3,T,P,0,1,0,-1> E0 ## E1 ## E0; }; \
    struct { glm::detail::swizzle<3,T,P,0,1,1,-1> E0 ## E1 ## E1; }; \
    struct { glm::detail::swizzle<3,T,P,0,1,2,-1> E0 ## E1 ## E2; }; \
    struct { glm::detail::swizzle<3,T,P,0,1,3,-1> E0 ## E1 ## E3; }; \
    struct { glm::detail::swizzle<3,T,P,0,2,0,-1> E0 ## E2 ## E0; }; \
    struct { glm::detail::swizzle<3,T,P,0,2,1,-1> E0 ## E2 ## E1; }; \
    struct { glm::detail::swizzle<3,T,P,0,2,2,-1> E0 ## E2 ## E2; }; \
    struct { glm::detail::swizzle<3,T,P,0,2,3,-1> E0 ## E2 ## E3; }; \
    struct { glm::detail::swizzle<3,T,P,0,3,0,-1> E0 ## E3 ## E0; }; \
    struct { glm::detail::swizzle<3,T,P,0,3,1,-1> E0 ## E3 ## E1; }; \
```

```
151     struct { glm::detail::swizzle<3,T,P,0,3,2,-1> E0 ## E3 ## E2; };
  6 \
151     struct { glm::detail::swizzle<3,T,P,0,3,3,-1> E0 ## E3 ## E3; };
  7 \
151     struct { glm::detail::swizzle<3,T,P,1,0,0,-1> E1 ## E0 ## E0; };
  8 \
151     struct { glm::detail::swizzle<3,T,P,1,0,1,-1> E1 ## E0 ## E1; };
  9 \
152     struct { glm::detail::swizzle<3,T,P,1,0,2,-1> E1 ## E0 ## E2; };
  0 \
152     struct { glm::detail::swizzle<3,T,P,1,0,3,-1> E1 ## E0 ## E3; };
  1 \
152     struct { glm::detail::swizzle<3,T,P,1,1,0,-1> E1 ## E1 ## E0; };
  2 \
152     struct { glm::detail::swizzle<3,T,P,1,1,1,-1> E1 ## E1 ## E1; };
  3 \
152     struct { glm::detail::swizzle<3,T,P,1,1,2,-1> E1 ## E1 ## E2; };
  4 \
152     struct { glm::detail::swizzle<3,T,P,1,1,3,-1> E1 ## E1 ## E3; };
  5 \
152     struct { glm::detail::swizzle<3,T,P,1,2,0,-1> E1 ## E2 ## E0; };
  6 \
152     struct { glm::detail::swizzle<3,T,P,1,2,1,-1> E1 ## E2 ## E1; };
  7 \
152     struct { glm::detail::swizzle<3,T,P,1,2,2,-1> E1 ## E2 ## E2; };
  8 \
152     struct { glm::detail::swizzle<3,T,P,1,2,3,-1> E1 ## E2 ## E3; };
  9 \
153     struct { glm::detail::swizzle<3,T,P,1,3,0,-1> E1 ## E3 ## E0; };
  0 \
153     struct { glm::detail::swizzle<3,T,P,1,3,1,-1> E1 ## E3 ## E1; };
  1 \
153     struct { glm::detail::swizzle<3,T,P,1,3,2,-1> E1 ## E3 ## E2; };
  2 \
153     struct { glm::detail::swizzle<3,T,P,1,3,3,-1> E1 ## E3 ## E3; };
  3 \
153     struct { glm::detail::swizzle<3,T,P,2,0,0,-1> E2 ## E0 ## E0; };
  4 \
153     struct { glm::detail::swizzle<3,T,P,2,0,1,-1> E2 ## E0 ## E1; };
  5 \
```

```
153     struct { glm::detail::swizzle<3,T,P,2,0,2,-1> E2 ## E0 ## E2; };
  6 \
153     struct { glm::detail::swizzle<3,T,P,2,0,3,-1> E2 ## E0 ## E3; };
  7 \
153     struct { glm::detail::swizzle<3,T,P,2,1,0,-1> E2 ## E1 ## E0; };
  8 \
153     struct { glm::detail::swizzle<3,T,P,2,1,1,-1> E2 ## E1 ## E1; };
  9 \
154     struct { glm::detail::swizzle<3,T,P,2,1,2,-1> E2 ## E1 ## E2; };
  0 \
154     struct { glm::detail::swizzle<3,T,P,2,1,3,-1> E2 ## E1 ## E3; };
  1 \
154     struct { glm::detail::swizzle<3,T,P,2,2,0,-1> E2 ## E2 ## E0; };
  2 \
154     struct { glm::detail::swizzle<3,T,P,2,2,1,-1> E2 ## E2 ## E1; };
  3 \
154     struct { glm::detail::swizzle<3,T,P,2,2,2,-1> E2 ## E2 ## E2; };
  4 \
154     struct { glm::detail::swizzle<3,T,P,2,2,3,-1> E2 ## E2 ## E3; };
  5 \
154     struct { glm::detail::swizzle<3,T,P,2,3,0,-1> E2 ## E3 ## E0; };
  6 \
154     struct { glm::detail::swizzle<3,T,P,2,3,1,-1> E2 ## E3 ## E1; };
  7 \
154     struct { glm::detail::swizzle<3,T,P,2,3,2,-1> E2 ## E3 ## E2; };
  8 \
154     struct { glm::detail::swizzle<3,T,P,2,3,3,-1> E2 ## E3 ## E3; };
  9 \
155     struct { glm::detail::swizzle<3,T,P,3,0,0,-1> E3 ## E0 ## E0; };
  0 \
155     struct { glm::detail::swizzle<3,T,P,3,0,1,-1> E3 ## E0 ## E1; };
  1 \
155     struct { glm::detail::swizzle<3,T,P,3,0,2,-1> E3 ## E0 ## E2; };
  2 \
155     struct { glm::detail::swizzle<3,T,P,3,0,3,-1> E3 ## E0 ## E3; };
  3 \
155     struct { glm::detail::swizzle<3,T,P,3,1,0,-1> E3 ## E1 ## E0; };
  4 \
155     struct { glm::detail::swizzle<3,T,P,3,1,1,-1> E3 ## E1 ## E1; };
  5 \
```

```
155     struct { glm::detail::swizzle<3,T,P,3,1,2,-1> E3 ## E1 ## E2; };
 6 \
155     struct { glm::detail::swizzle<3,T,P,3,1,3,-1> E3 ## E1 ## E3; };
 7 \
155     struct { glm::detail::swizzle<3,T,P,3,2,0,-1> E3 ## E2 ## E0; };
 8 \
155     struct { glm::detail::swizzle<3,T,P,3,2,1,-1> E3 ## E2 ## E1; };
 9 \
156     struct { glm::detail::swizzle<3,T,P,3,2,2,-1> E3 ## E2 ## E2; };
 0 \
156     struct { glm::detail::swizzle<3,T,P,3,2,3,-1> E3 ## E2 ## E3; };
 1 \
156     struct { glm::detail::swizzle<3,T,P,3,3,0,-1> E3 ## E3 ## E0; };
 2 \
156     struct { glm::detail::swizzle<3,T,P,3,3,1,-1> E3 ## E3 ## E1; };
 3 \
156     struct { glm::detail::swizzle<3,T,P,3,3,2,-1> E3 ## E3 ## E2; };
 4 \
156     struct { glm::detail::swizzle<3,T,P,3,3,3,-1> E3 ## E3 ## E3; };
 5
156
156 #define _GLM_SWIZZLE4_4_MEMBERS(T,P,E0,E1,E2,E3) \
157     struct { glm::detail::swizzle<4,T,P,0,0,0,0> E0 ## E0 ## E0 ##
 8 E0; }; \
156     struct { glm::detail::swizzle<4,T,P,0,0,0,1> E0 ## E0 ## E0 ##
 9 E1; }; \
157     struct { glm::detail::swizzle<4,T,P,0,0,0,2> E0 ## E0 ## E0 ##
 0 E2; }; \
157     struct { glm::detail::swizzle<4,T,P,0,0,0,3> E0 ## E0 ## E0 ##
 1 E3; }; \
157     struct { glm::detail::swizzle<4,T,P,0,0,1,0> E0 ## E0 ## E1 ##
 2 E0; }; \
157     struct { glm::detail::swizzle<4,T,P,0,0,1,1> E0 ## E0 ## E1 ##
 3 E1; }; \
157     struct { glm::detail::swizzle<4,T,P,0,0,1,2> E0 ## E0 ## E1 ##
 4 E2; }; \
157     struct { glm::detail::swizzle<4,T,P,0,0,1,3> E0 ## E0 ## E1 ##
 5 E3; }; \
157     struct { glm::detail::swizzle<4,T,P,0,0,2,0> E0 ## E0 ## E2 ##
 6 E0; }; \
```

```
157     struct { glm::detail::swizzle<4,T,P,0,0,2,1> E0 ## E0 ## E2 ##
  7 E1; }; \
157     struct { glm::detail::swizzle<4,T,P,0,0,2,2> E0 ## E0 ## E2 ##
  8 E2; }; \
157     struct { glm::detail::swizzle<4,T,P,0,0,2,3> E0 ## E0 ## E2 ##
  9 E3; }; \
158     struct { glm::detail::swizzle<4,T,P,0,0,3,0> E0 ## E0 ## E3 ##
  0 E0; }; \
158     struct { glm::detail::swizzle<4,T,P,0,0,3,1> E0 ## E0 ## E3 ##
  1 E1; }; \
158     struct { glm::detail::swizzle<4,T,P,0,0,3,2> E0 ## E0 ## E3 ##
  2 E2; }; \
158     struct { glm::detail::swizzle<4,T,P,0,0,3,3> E0 ## E0 ## E3 ##
  3 E3; }; \
158     struct { glm::detail::swizzle<4,T,P,0,1,0,0> E0 ## E1 ## E0 ##
  4 E0; }; \
158     struct { glm::detail::swizzle<4,T,P,0,1,0,1> E0 ## E1 ## E0 ##
  5 E1; }; \
158     struct { glm::detail::swizzle<4,T,P,0,1,0,2> E0 ## E1 ## E0 ##
  6 E2; }; \
158     struct { glm::detail::swizzle<4,T,P,0,1,0,3> E0 ## E1 ## E0 ##
  7 E3; }; \
158     struct { glm::detail::swizzle<4,T,P,0,1,1,0> E0 ## E1 ## E1 ##
  8 E0; }; \
158     struct { glm::detail::swizzle<4,T,P,0,1,1,1> E0 ## E1 ## E1 ##
  9 E1; }; \
159     struct { glm::detail::swizzle<4,T,P,0,1,1,2> E0 ## E1 ## E1 ##
  0 E2; }; \
159     struct { glm::detail::swizzle<4,T,P,0,1,1,3> E0 ## E1 ## E1 ##
  1 E3; }; \
159     struct { glm::detail::swizzle<4,T,P,0,1,2,0> E0 ## E1 ## E2 ##
  2 E0; }; \
159     struct { glm::detail::swizzle<4,T,P,0,1,2,1> E0 ## E1 ## E2 ##
  3 E1; }; \
159     struct { glm::detail::swizzle<4,T,P,0,1,2,2> E0 ## E1 ## E2 ##
  4 E2; }; \
159     struct { glm::detail::swizzle<4,T,P,0,1,2,3> E0 ## E1 ## E2 ##
  5 E3; }; \
159     struct { glm::detail::swizzle<4,T,P,0,1,3,0> E0 ## E1 ## E3 ##
  6 E0; }; \
```

```
159     struct { glm::detail::swizzle<4,T,P,0,1,3,1> E0 ## E1 ## E3 ##
  7 E1; }; \
159     struct { glm::detail::swizzle<4,T,P,0,1,3,2> E0 ## E1 ## E3 ##
  8 E2; }; \
159     struct { glm::detail::swizzle<4,T,P,0,1,3,3> E0 ## E1 ## E3 ##
  9 E3; }; \
160     struct { glm::detail::swizzle<4,T,P,0,2,0,0> E0 ## E2 ## E0 ##
  0 E0; }; \
160     struct { glm::detail::swizzle<4,T,P,0,2,0,1> E0 ## E2 ## E0 ##
  1 E1; }; \
160     struct { glm::detail::swizzle<4,T,P,0,2,0,2> E0 ## E2 ## E0 ##
  2 E2; }; \
160     struct { glm::detail::swizzle<4,T,P,0,2,0,3> E0 ## E2 ## E0 ##
  3 E3; }; \
160     struct { glm::detail::swizzle<4,T,P,0,2,1,0> E0 ## E2 ## E1 ##
  4 E0; }; \
160     struct { glm::detail::swizzle<4,T,P,0,2,1,1> E0 ## E2 ## E1 ##
  5 E1; }; \
160     struct { glm::detail::swizzle<4,T,P,0,2,1,2> E0 ## E2 ## E1 ##
  6 E2; }; \
160     struct { glm::detail::swizzle<4,T,P,0,2,1,3> E0 ## E2 ## E1 ##
  7 E3; }; \
160     struct { glm::detail::swizzle<4,T,P,0,2,2,0> E0 ## E2 ## E2 ##
  8 E0; }; \
160     struct { glm::detail::swizzle<4,T,P,0,2,2,1> E0 ## E2 ## E2 ##
  9 E1; }; \
161     struct { glm::detail::swizzle<4,T,P,0,2,2,2> E0 ## E2 ## E2 ##
  0 E2; }; \
161     struct { glm::detail::swizzle<4,T,P,0,2,2,3> E0 ## E2 ## E2 ##
  1 E3; }; \
161     struct { glm::detail::swizzle<4,T,P,0,2,3,0> E0 ## E2 ## E3 ##
  2 E0; }; \
161     struct { glm::detail::swizzle<4,T,P,0,2,3,1> E0 ## E2 ## E3 ##
  3 E1; }; \
161     struct { glm::detail::swizzle<4,T,P,0,2,3,2> E0 ## E2 ## E3 ##
  4 E2; }; \
161     struct { glm::detail::swizzle<4,T,P,0,2,3,3> E0 ## E2 ## E3 ##
  5 E3; }; \
161     struct { glm::detail::swizzle<4,T,P,1,0,0,0> E1 ## E0 ## E0 ##
  6 E0; }; \
```

```
161     struct { glm::detail::swizzle<4,T,P,1,0,0,1> E1 ## E0 ## E0 ##
  7 E1; }; \
161     struct { glm::detail::swizzle<4,T,P,1,0,0,2> E1 ## E0 ## E0 ##
  8 E2; }; \
161     struct { glm::detail::swizzle<4,T,P,1,0,0,3> E1 ## E0 ## E0 ##
  9 E3; }; \
162     struct { glm::detail::swizzle<4,T,P,1,0,1,0> E1 ## E0 ## E1 ##
  0 E0; }; \
162     struct { glm::detail::swizzle<4,T,P,1,0,1,1> E1 ## E0 ## E1 ##
  1 E1; }; \
162     struct { glm::detail::swizzle<4,T,P,1,0,1,2> E1 ## E0 ## E1 ##
  2 E2; }; \
162     struct { glm::detail::swizzle<4,T,P,1,0,1,3> E1 ## E0 ## E1 ##
  3 E3; }; \
162     struct { glm::detail::swizzle<4,T,P,1,0,2,0> E1 ## E0 ## E2 ##
  4 E0; }; \
162     struct { glm::detail::swizzle<4,T,P,1,0,2,1> E1 ## E0 ## E2 ##
  5 E1; }; \
162     struct { glm::detail::swizzle<4,T,P,1,0,2,2> E1 ## E0 ## E2 ##
  6 E2; }; \
162     struct { glm::detail::swizzle<4,T,P,1,0,2,3> E1 ## E0 ## E2 ##
  7 E3; }; \
162     struct { glm::detail::swizzle<4,T,P,1,0,3,0> E1 ## E0 ## E3 ##
  8 E0; }; \
162     struct { glm::detail::swizzle<4,T,P,1,0,3,1> E1 ## E0 ## E3 ##
  9 E1; }; \
163     struct { glm::detail::swizzle<4,T,P,1,0,3,2> E1 ## E0 ## E3 ##
  0 E2; }; \
163     struct { glm::detail::swizzle<4,T,P,1,0,3,3> E1 ## E0 ## E3 ##
  1 E3; }; \
163     struct { glm::detail::swizzle<4,T,P,1,1,0,0> E1 ## E1 ## E0 ##
  2 E0; }; \
163     struct { glm::detail::swizzle<4,T,P,1,1,0,1> E1 ## E1 ## E0 ##
  3 E1; }; \
163     struct { glm::detail::swizzle<4,T,P,1,1,0,2> E1 ## E1 ## E0 ##
  4 E2; }; \
163     struct { glm::detail::swizzle<4,T,P,1,1,0,3> E1 ## E1 ## E0 ##
  5 E3; }; \
163     struct { glm::detail::swizzle<4,T,P,1,1,1,0> E1 ## E1 ## E1 ##
  6 E0; }; \
```

```
163     struct { glm::detail::swizzle<4,T,P,1,1,1,1> E1 ## E1 ## E1 ##
  7 E1; }; \
163     struct { glm::detail::swizzle<4,T,P,1,1,1,2> E1 ## E1 ## E1 ##
  8 E2; }; \
163     struct { glm::detail::swizzle<4,T,P,1,1,1,3> E1 ## E1 ## E1 ##
  9 E3; }; \
164     struct { glm::detail::swizzle<4,T,P,1,1,2,0> E1 ## E1 ## E2 ##
  0 E0; }; \
164     struct { glm::detail::swizzle<4,T,P,1,1,2,1> E1 ## E1 ## E2 ##
  1 E1; }; \
164     struct { glm::detail::swizzle<4,T,P,1,1,2,2> E1 ## E1 ## E2 ##
  2 E2; }; \
164     struct { glm::detail::swizzle<4,T,P,1,1,2,3> E1 ## E1 ## E2 ##
  3 E3; }; \
164     struct { glm::detail::swizzle<4,T,P,1,1,3,0> E1 ## E1 ## E3 ##
  4 E0; }; \
164     struct { glm::detail::swizzle<4,T,P,1,1,3,1> E1 ## E1 ## E3 ##
  5 E1; }; \
164     struct { glm::detail::swizzle<4,T,P,1,1,3,2> E1 ## E1 ## E3 ##
  6 E2; }; \
164     struct { glm::detail::swizzle<4,T,P,1,1,3,3> E1 ## E1 ## E3 ##
  7 E3; }; \
164     struct { glm::detail::swizzle<4,T,P,1,2,0,0> E1 ## E2 ## E0 ##
  8 E0; }; \
164     struct { glm::detail::swizzle<4,T,P,1,2,0,1> E1 ## E2 ## E0 ##
  9 E1; }; \
165     struct { glm::detail::swizzle<4,T,P,1,2,0,2> E1 ## E2 ## E0 ##
  0 E2; }; \
165     struct { glm::detail::swizzle<4,T,P,1,2,0,3> E1 ## E2 ## E0 ##
  1 E3; }; \
165     struct { glm::detail::swizzle<4,T,P,1,2,1,0> E1 ## E2 ## E1 ##
  2 E0; }; \
165     struct { glm::detail::swizzle<4,T,P,1,2,1,1> E1 ## E2 ## E1 ##
  3 E1; }; \
165     struct { glm::detail::swizzle<4,T,P,1,2,1,2> E1 ## E2 ## E1 ##
  4 E2; }; \
165     struct { glm::detail::swizzle<4,T,P,1,2,1,3> E1 ## E2 ## E1 ##
  5 E3; }; \
165     struct { glm::detail::swizzle<4,T,P,1,2,2,0> E1 ## E2 ## E2 ##
  6 E0; }; \
```

```
165     struct { glm::detail::swizzle<4,T,P,1,2,2,1> E1 ## E2 ## E2 ##
  7 E1; }; \
165     struct { glm::detail::swizzle<4,T,P,1,2,2,2> E1 ## E2 ## E2 ##
  8 E2; }; \
165     struct { glm::detail::swizzle<4,T,P,1,2,2,3> E1 ## E2 ## E2 ##
  9 E3; }; \
166     struct { glm::detail::swizzle<4,T,P,1,2,3,0> E1 ## E2 ## E3 ##
  0 E0; }; \
166     struct { glm::detail::swizzle<4,T,P,1,2,3,1> E1 ## E2 ## E3 ##
  1 E1; }; \
166     struct { glm::detail::swizzle<4,T,P,1,2,3,2> E1 ## E2 ## E3 ##
  2 E2; }; \
166     struct { glm::detail::swizzle<4,T,P,1,2,3,3> E1 ## E2 ## E3 ##
  3 E3; }; \
166     struct { glm::detail::swizzle<4,T,P,1,3,0,0> E1 ## E3 ## E0 ##
  4 E0; }; \
166     struct { glm::detail::swizzle<4,T,P,1,3,0,1> E1 ## E3 ## E0 ##
  5 E1; }; \
166     struct { glm::detail::swizzle<4,T,P,1,3,0,2> E1 ## E3 ## E0 ##
  6 E2; }; \
166     struct { glm::detail::swizzle<4,T,P,1,3,0,3> E1 ## E3 ## E0 ##
  7 E3; }; \
166     struct { glm::detail::swizzle<4,T,P,1,3,1,0> E1 ## E3 ## E1 ##
  8 E0; }; \
166     struct { glm::detail::swizzle<4,T,P,1,3,1,1> E1 ## E3 ## E1 ##
  9 E1; }; \
167     struct { glm::detail::swizzle<4,T,P,1,3,1,2> E1 ## E3 ## E1 ##
  0 E2; }; \
167     struct { glm::detail::swizzle<4,T,P,1,3,1,3> E1 ## E3 ## E1 ##
  1 E3; }; \
167     struct { glm::detail::swizzle<4,T,P,1,3,2,0> E1 ## E3 ## E2 ##
  2 E0; }; \
167     struct { glm::detail::swizzle<4,T,P,1,3,2,1> E1 ## E3 ## E2 ##
  3 E1; }; \
167     struct { glm::detail::swizzle<4,T,P,1,3,2,2> E1 ## E3 ## E2 ##
  4 E2; }; \
167     struct { glm::detail::swizzle<4,T,P,1,3,2,3> E1 ## E3 ## E2 ##
  5 E3; }; \
167     struct { glm::detail::swizzle<4,T,P,1,3,3,0> E1 ## E3 ## E3 ##
  6 E0; }; \
```

```
167     struct { glm::detail::swizzle<4,T,P,1,3,3,1> E1 ## E3 ## E3 ##
  7 E1; }; \
167     struct { glm::detail::swizzle<4,T,P,1,3,3,2> E1 ## E3 ## E3 ##
  8 E2; }; \
167     struct { glm::detail::swizzle<4,T,P,1,3,3,3> E1 ## E3 ## E3 ##
  9 E3; }; \
168     struct { glm::detail::swizzle<4,T,P,2,0,0,0> E2 ## E0 ## E0 ##
  0 E0; }; \
168     struct { glm::detail::swizzle<4,T,P,2,0,0,1> E2 ## E0 ## E0 ##
  1 E1; }; \
168     struct { glm::detail::swizzle<4,T,P,2,0,0,2> E2 ## E0 ## E0 ##
  2 E2; }; \
168     struct { glm::detail::swizzle<4,T,P,2,0,0,3> E2 ## E0 ## E0 ##
  3 E3; }; \
168     struct { glm::detail::swizzle<4,T,P,2,0,1,0> E2 ## E0 ## E1 ##
  4 E0; }; \
168     struct { glm::detail::swizzle<4,T,P,2,0,1,1> E2 ## E0 ## E1 ##
  5 E1; }; \
168     struct { glm::detail::swizzle<4,T,P,2,0,1,2> E2 ## E0 ## E1 ##
  6 E2; }; \
168     struct { glm::detail::swizzle<4,T,P,2,0,1,3> E2 ## E0 ## E1 ##
  7 E3; }; \
168     struct { glm::detail::swizzle<4,T,P,2,0,2,0> E2 ## E0 ## E2 ##
  8 E0; }; \
168     struct { glm::detail::swizzle<4,T,P,2,0,2,1> E2 ## E0 ## E2 ##
  9 E1; }; \
169     struct { glm::detail::swizzle<4,T,P,2,0,2,2> E2 ## E0 ## E2 ##
  0 E2; }; \
169     struct { glm::detail::swizzle<4,T,P,2,0,2,3> E2 ## E0 ## E2 ##
  1 E3; }; \
169     struct { glm::detail::swizzle<4,T,P,2,0,3,0> E2 ## E0 ## E3 ##
  2 E0; }; \
169     struct { glm::detail::swizzle<4,T,P,2,0,3,1> E2 ## E0 ## E3 ##
  3 E1; }; \
169     struct { glm::detail::swizzle<4,T,P,2,0,3,2> E2 ## E0 ## E3 ##
  4 E2; }; \
169     struct { glm::detail::swizzle<4,T,P,2,0,3,3> E2 ## E0 ## E3 ##
  5 E3; }; \
169     struct { glm::detail::swizzle<4,T,P,2,1,0,0> E2 ## E1 ## E0 ##
  6 E0; }; \
```

```
169     struct { glm::detail::swizzle<4,T,P,2,1,0,1> E2 ## E1 ## E0 ##
  7 E1; }; \
169     struct { glm::detail::swizzle<4,T,P,2,1,0,2> E2 ## E1 ## E0 ##
  8 E2; }; \
169     struct { glm::detail::swizzle<4,T,P,2,1,0,3> E2 ## E1 ## E0 ##
  9 E3; }; \
170     struct { glm::detail::swizzle<4,T,P,2,1,1,0> E2 ## E1 ## E1 ##
  0 E0; }; \
170     struct { glm::detail::swizzle<4,T,P,2,1,1,1> E2 ## E1 ## E1 ##
  1 E1; }; \
170     struct { glm::detail::swizzle<4,T,P,2,1,1,2> E2 ## E1 ## E1 ##
  2 E2; }; \
170     struct { glm::detail::swizzle<4,T,P,2,1,1,3> E2 ## E1 ## E1 ##
  3 E3; }; \
170     struct { glm::detail::swizzle<4,T,P,2,1,2,0> E2 ## E1 ## E2 ##
  4 E0; }; \
170     struct { glm::detail::swizzle<4,T,P,2,1,2,1> E2 ## E1 ## E2 ##
  5 E1; }; \
170     struct { glm::detail::swizzle<4,T,P,2,1,2,2> E2 ## E1 ## E2 ##
  6 E2; }; \
170     struct { glm::detail::swizzle<4,T,P,2,1,2,3> E2 ## E1 ## E2 ##
  7 E3; }; \
170     struct { glm::detail::swizzle<4,T,P,2,1,3,0> E2 ## E1 ## E3 ##
  8 E0; }; \
170     struct { glm::detail::swizzle<4,T,P,2,1,3,1> E2 ## E1 ## E3 ##
  9 E1; }; \
171     struct { glm::detail::swizzle<4,T,P,2,1,3,2> E2 ## E1 ## E3 ##
  0 E2; }; \
171     struct { glm::detail::swizzle<4,T,P,2,1,3,3> E2 ## E1 ## E3 ##
  1 E3; }; \
171     struct { glm::detail::swizzle<4,T,P,2,2,0,0> E2 ## E2 ## E0 ##
  2 E0; }; \
171     struct { glm::detail::swizzle<4,T,P,2,2,0,1> E2 ## E2 ## E0 ##
  3 E1; }; \
171     struct { glm::detail::swizzle<4,T,P,2,2,0,2> E2 ## E2 ## E0 ##
  4 E2; }; \
171     struct { glm::detail::swizzle<4,T,P,2,2,0,3> E2 ## E2 ## E0 ##
  5 E3; }; \
171     struct { glm::detail::swizzle<4,T,P,2,2,1,0> E2 ## E2 ## E1 ##
  6 E0; }; \
```

```
171     struct { glm::detail::swizzle<4,T,P,2,2,1,1> E2 ## E2 ## E1 ##
  7 E1; }; \
171     struct { glm::detail::swizzle<4,T,P,2,2,1,2> E2 ## E2 ## E1 ##
  8 E2; }; \
171     struct { glm::detail::swizzle<4,T,P,2,2,1,3> E2 ## E2 ## E1 ##
  9 E3; }; \
172     struct { glm::detail::swizzle<4,T,P,2,2,2,0> E2 ## E2 ## E2 ##
  0 E0; }; \
172     struct { glm::detail::swizzle<4,T,P,2,2,2,1> E2 ## E2 ## E2 ##
  1 E1; }; \
172     struct { glm::detail::swizzle<4,T,P,2,2,2,2> E2 ## E2 ## E2 ##
  2 E2; }; \
172     struct { glm::detail::swizzle<4,T,P,2,2,2,3> E2 ## E2 ## E2 ##
  3 E3; }; \
172     struct { glm::detail::swizzle<4,T,P,2,2,3,0> E2 ## E2 ## E3 ##
  4 E0; }; \
172     struct { glm::detail::swizzle<4,T,P,2,2,3,1> E2 ## E2 ## E3 ##
  5 E1; }; \
172     struct { glm::detail::swizzle<4,T,P,2,2,3,2> E2 ## E2 ## E3 ##
  6 E2; }; \
172     struct { glm::detail::swizzle<4,T,P,2,2,3,3> E2 ## E2 ## E3 ##
  7 E3; }; \
172     struct { glm::detail::swizzle<4,T,P,2,3,0,0> E2 ## E3 ## E0 ##
  8 E0; }; \
172     struct { glm::detail::swizzle<4,T,P,2,3,0,1> E2 ## E3 ## E0 ##
  9 E1; }; \
173     struct { glm::detail::swizzle<4,T,P,2,3,0,2> E2 ## E3 ## E0 ##
  0 E2; }; \
173     struct { glm::detail::swizzle<4,T,P,2,3,0,3> E2 ## E3 ## E0 ##
  1 E3; }; \
173     struct { glm::detail::swizzle<4,T,P,2,3,1,0> E2 ## E3 ## E1 ##
  2 E0; }; \
173     struct { glm::detail::swizzle<4,T,P,2,3,1,1> E2 ## E3 ## E1 ##
  3 E1; }; \
173     struct { glm::detail::swizzle<4,T,P,2,3,1,2> E2 ## E3 ## E1 ##
  4 E2; }; \
173     struct { glm::detail::swizzle<4,T,P,2,3,1,3> E2 ## E3 ## E1 ##
  5 E3; }; \
173     struct { glm::detail::swizzle<4,T,P,2,3,2,0> E2 ## E3 ## E2 ##
  6 E0; }; \
```

```
173     struct { glm::detail::swizzle<4,T,P,2,3,2,1> E2 ## E3 ## E2 ##
  7 E1; }; \
173     struct { glm::detail::swizzle<4,T,P,2,3,2,2> E2 ## E3 ## E2 ##
  8 E2; }; \
173     struct { glm::detail::swizzle<4,T,P,2,3,2,3> E2 ## E3 ## E2 ##
  9 E3; }; \
174     struct { glm::detail::swizzle<4,T,P,2,3,3,0> E2 ## E3 ## E3 ##
  0 E0; }; \
174     struct { glm::detail::swizzle<4,T,P,2,3,3,1> E2 ## E3 ## E3 ##
  1 E1; }; \
174     struct { glm::detail::swizzle<4,T,P,2,3,3,2> E2 ## E3 ## E3 ##
  2 E2; }; \
174     struct { glm::detail::swizzle<4,T,P,2,3,3,3> E2 ## E3 ## E3 ##
  3 E3; }; \
174     struct { glm::detail::swizzle<4,T,P,3,0,0,0> E3 ## E0 ## E0 ##
  4 E0; }; \
174     struct { glm::detail::swizzle<4,T,P,3,0,0,1> E3 ## E0 ## E0 ##
  5 E1; }; \
174     struct { glm::detail::swizzle<4,T,P,3,0,0,2> E3 ## E0 ## E0 ##
  6 E2; }; \
174     struct { glm::detail::swizzle<4,T,P,3,0,0,3> E3 ## E0 ## E0 ##
  7 E3; }; \
174     struct { glm::detail::swizzle<4,T,P,3,0,1,0> E3 ## E0 ## E1 ##
  8 E0; }; \
174     struct { glm::detail::swizzle<4,T,P,3,0,1,1> E3 ## E0 ## E1 ##
  9 E1; }; \
175     struct { glm::detail::swizzle<4,T,P,3,0,1,2> E3 ## E0 ## E1 ##
  0 E2; }; \
175     struct { glm::detail::swizzle<4,T,P,3,0,1,3> E3 ## E0 ## E1 ##
  1 E3; }; \
175     struct { glm::detail::swizzle<4,T,P,3,0,2,0> E3 ## E0 ## E2 ##
  2 E0; }; \
175     struct { glm::detail::swizzle<4,T,P,3,0,2,1> E3 ## E0 ## E2 ##
  3 E1; }; \
175     struct { glm::detail::swizzle<4,T,P,3,0,2,2> E3 ## E0 ## E2 ##
  4 E2; }; \
175     struct { glm::detail::swizzle<4,T,P,3,0,2,3> E3 ## E0 ## E2 ##
  5 E3; }; \
175     struct { glm::detail::swizzle<4,T,P,3,0,3,0> E3 ## E0 ## E3 ##
  6 E0; }; \
```

```
175     struct { glm::detail::swizzle<4,T,P,3,0,3,1> E3 ## E0 ## E3 ##
  7 E1; }; \
175     struct { glm::detail::swizzle<4,T,P,3,0,3,2> E3 ## E0 ## E3 ##
  8 E2; }; \
175     struct { glm::detail::swizzle<4,T,P,3,0,3,3> E3 ## E0 ## E3 ##
  9 E3; }; \
176     struct { glm::detail::swizzle<4,T,P,3,1,0,0> E3 ## E1 ## E0 ##
  0 E0; }; \
176     struct { glm::detail::swizzle<4,T,P,3,1,0,1> E3 ## E1 ## E0 ##
  1 E1; }; \
176     struct { glm::detail::swizzle<4,T,P,3,1,0,2> E3 ## E1 ## E0 ##
  2 E2; }; \
176     struct { glm::detail::swizzle<4,T,P,3,1,0,3> E3 ## E1 ## E0 ##
  3 E3; }; \
176     struct { glm::detail::swizzle<4,T,P,3,1,1,0> E3 ## E1 ## E1 ##
  4 E0; }; \
176     struct { glm::detail::swizzle<4,T,P,3,1,1,1> E3 ## E1 ## E1 ##
  5 E1; }; \
176     struct { glm::detail::swizzle<4,T,P,3,1,1,2> E3 ## E1 ## E1 ##
  6 E2; }; \
176     struct { glm::detail::swizzle<4,T,P,3,1,1,3> E3 ## E1 ## E1 ##
  7 E3; }; \
176     struct { glm::detail::swizzle<4,T,P,3,1,2,0> E3 ## E1 ## E2 ##
  8 E0; }; \
176     struct { glm::detail::swizzle<4,T,P,3,1,2,1> E3 ## E1 ## E2 ##
  9 E1; }; \
177     struct { glm::detail::swizzle<4,T,P,3,1,2,2> E3 ## E1 ## E2 ##
  0 E2; }; \
177     struct { glm::detail::swizzle<4,T,P,3,1,2,3> E3 ## E1 ## E2 ##
  1 E3; }; \
177     struct { glm::detail::swizzle<4,T,P,3,1,3,0> E3 ## E1 ## E3 ##
  2 E0; }; \
177     struct { glm::detail::swizzle<4,T,P,3,1,3,1> E3 ## E1 ## E3 ##
  3 E1; }; \
177     struct { glm::detail::swizzle<4,T,P,3,1,3,2> E3 ## E1 ## E3 ##
  4 E2; }; \
177     struct { glm::detail::swizzle<4,T,P,3,1,3,3> E3 ## E1 ## E3 ##
  5 E3; }; \
177     struct { glm::detail::swizzle<4,T,P,3,2,0,0> E3 ## E2 ## E0 ##
  6 E0; }; \
```

```
177     struct { glm::detail::swizzle<4,T,P,3,2,0,1> E3 ## E2 ## E0 ##
  7 E1; }; \
177     struct { glm::detail::swizzle<4,T,P,3,2,0,2> E3 ## E2 ## E0 ##
  8 E2; }; \
177     struct { glm::detail::swizzle<4,T,P,3,2,0,3> E3 ## E2 ## E0 ##
  9 E3; }; \
178     struct { glm::detail::swizzle<4,T,P,3,2,1,0> E3 ## E2 ## E1 ##
  0 E0; }; \
178     struct { glm::detail::swizzle<4,T,P,3,2,1,1> E3 ## E2 ## E1 ##
  1 E1; }; \
178     struct { glm::detail::swizzle<4,T,P,3,2,1,2> E3 ## E2 ## E1 ##
  2 E2; }; \
178     struct { glm::detail::swizzle<4,T,P,3,2,1,3> E3 ## E2 ## E1 ##
  3 E3; }; \
178     struct { glm::detail::swizzle<4,T,P,3,2,2,0> E3 ## E2 ## E2 ##
  4 E0; }; \
178     struct { glm::detail::swizzle<4,T,P,3,2,2,1> E3 ## E2 ## E2 ##
  5 E1; }; \
178     struct { glm::detail::swizzle<4,T,P,3,2,2,2> E3 ## E2 ## E2 ##
  6 E2; }; \
178     struct { glm::detail::swizzle<4,T,P,3,2,2,3> E3 ## E2 ## E2 ##
  7 E3; }; \
178     struct { glm::detail::swizzle<4,T,P,3,2,3,0> E3 ## E2 ## E3 ##
  8 E0; }; \
178     struct { glm::detail::swizzle<4,T,P,3,2,3,1> E3 ## E2 ## E3 ##
  9 E1; }; \
179     struct { glm::detail::swizzle<4,T,P,3,2,3,2> E3 ## E2 ## E3 ##
  0 E2; }; \
179     struct { glm::detail::swizzle<4,T,P,3,2,3,3> E3 ## E2 ## E3 ##
  1 E3; }; \
179     struct { glm::detail::swizzle<4,T,P,3,3,0,0> E3 ## E3 ## E0 ##
  2 E0; }; \
179     struct { glm::detail::swizzle<4,T,P,3,3,0,1> E3 ## E3 ## E0 ##
  3 E1; }; \
179     struct { glm::detail::swizzle<4,T,P,3,3,0,2> E3 ## E3 ## E0 ##
  4 E2; }; \
179     struct { glm::detail::swizzle<4,T,P,3,3,0,3> E3 ## E3 ## E0 ##
  5 E3; }; \
179     struct { glm::detail::swizzle<4,T,P,3,3,1,0> E3 ## E3 ## E1 ##
  6 E0; }; \
```

```
    struct { glm::detail::swizzle<4,T,P,3,3,1,1> E3 ## E3 ## E1 ## E1; }; \
    struct { glm::detail::swizzle<4,T,P,3,3,1,2> E3 ## E3 ## E1 ## E2; }; \
    struct { glm::detail::swizzle<4,T,P,3,3,1,3> E3 ## E3 ## E1 ## E3; }; \
    struct { glm::detail::swizzle<4,T,P,3,3,2,0> E3 ## E3 ## E2 ## E0; }; \
    struct { glm::detail::swizzle<4,T,P,3,3,2,1> E3 ## E3 ## E2 ## E1; }; \
    struct { glm::detail::swizzle<4,T,P,3,3,2,2> E3 ## E3 ## E2 ## E2; }; \
    struct { glm::detail::swizzle<4,T,P,3,3,2,3> E3 ## E3 ## E2 ## E3; }; \
    struct { glm::detail::swizzle<4,T,P,3,3,3,0> E3 ## E3 ## E3 ## E0; }; \
    struct { glm::detail::swizzle<4,T,P,3,3,3,1> E3 ## E3 ## E3 ## E1; }; \
    struct { glm::detail::swizzle<4,T,P,3,3,3,2> E3 ## E3 ## E3 ## E2; }; \
    struct { glm::detail::swizzle<4,T,P,3,3,3,3> E3 ## E3 ## E3 ## E3; };

#endif//glm_core_swizzle

#ifndef glm_core_swizzle_func
#define glm_core_swizzle_func

#define GLM_SWIZZLE_GEN_VEC2_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE, CONST, A, B) \
    SWIZZLED_TYPE<TMPL_TYPE> A ## B() CONST \
    { \
        return SWIZZLED_TYPE<TMPL_TYPE>(this->A, this->B); \
    }

#define GLM_SWIZZLE_GEN_VEC3_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE, CONST, A, B, C) \
    SWIZZLED_TYPE<TMPL_TYPE> A ## B ## C() CONST \
    { \
```

```
182         return SWIZZLED_TYPE<TMPL_TYPE>(this->A, this->B, this->C);
 3 \
182     }
182
182 #define GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE,
 6 SWIZZLED_TYPE, CONST, A, B, C, D) \
182     SWIZZLED_TYPE<TMPL_TYPE> A ## B ## C ## D() CONST \
182     { \
182         return SWIZZLED_TYPE<TMPL_TYPE>(this->A, this->B, this->C,
 9 this->D); \
183     }
183
183 #define GLM_SWIZZLE_GEN_VEC2_ENTRY_DEF(TMPL_TYPE, CLASS_TYPE,
 2 SWIZZLED_TYPE, CONST, A, B) \
183     template <typename TMPL_TYPE> \
183     SWIZZLED_TYPE<TMPL_TYPE> CLASS_TYPE<TMPL_TYPE>::A ## B() CONST \
184     { \
185         return SWIZZLED_TYPE<TMPL_TYPE>(this->A, this->B); \
186     }
187
188 #define GLM_SWIZZLE_GEN_VEC3_ENTRY_DEF(TMPL_TYPE, CLASS_TYPE,
 9 SWIZZLED_TYPE, CONST, A, B, C) \
184     template <typename TMPL_TYPE> \
184     SWIZZLED_TYPE<TMPL_TYPE> CLASS_TYPE<TMPL_TYPE>::A ## B ## C()
 1 CONST \
184     { \
182         return SWIZZLED_TYPE<TMPL_TYPE>(this->A, this->B, this->C);
 3 \
184     }
184
185 #define GLM_SWIZZLE_GEN_VEC4_ENTRY_DEF(TMPL_TYPE, CLASS_TYPE,
 6 SWIZZLED_TYPE, CONST, A, B, C, D) \
184     template <typename TMPL_TYPE> \
187     SWIZZLED_TYPE<TMPL_TYPE> CLASS_TYPE<TMPL_TYPE>::A ## B ## C ##
 8 D() CONST \
184     { \
189         return SWIZZLED_TYPE<TMPL_TYPE>(this->A, this->B, this->C,
 0 this->D); \
185     }
185
```

```
#define GLM_MUTABLE

#define GLM_SWIZZLE_GEN_REF2_FROM_VEC2_SWIZZLE(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE, A, B) \
    GLM_SWIZZLE_GEN_VEC2_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE, GLM_MUTABLE, A, B) \
    GLM_SWIZZLE_GEN_VEC2_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE, GLM_MUTABLE, B, A)

#define GLM_SWIZZLE_GEN_REF_FROM_VEC2(TMPL_TYPE, CLASS_TYPE, SWIZZLED_VEC2_TYPE) \
    GLM_SWIZZLE_GEN_REF2_FROM_VEC2_SWIZZLE(TMPL_TYPE, CLASS_TYPE, SWIZZLED_VEC2_TYPE, x, y) \
    GLM_SWIZZLE_GEN_REF2_FROM_VEC2_SWIZZLE(TMPL_TYPE, CLASS_TYPE, SWIZZLED_VEC2_TYPE, r, g) \
    GLM_SWIZZLE_GEN_REF2_FROM_VEC2_SWIZZLE(TMPL_TYPE, CLASS_TYPE, SWIZZLED_VEC2_TYPE, s, t)

//GLM_SWIZZLE_GEN_REF_FROM_VEC2(valType, detail::vec2, detail::ref2)

#define GLM_SWIZZLE_GEN_REF2_FROM_VEC3_SWIZZLE(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE, A, B, C) \
    GLM_SWIZZLE_GEN_VEC2_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE, GLM_MUTABLE, A, B) \
    GLM_SWIZZLE_GEN_VEC2_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE, GLM_MUTABLE, A, C) \
    GLM_SWIZZLE_GEN_VEC2_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE, GLM_MUTABLE, B, A) \
    GLM_SWIZZLE_GEN_VEC2_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE, GLM_MUTABLE, B, C) \
    GLM_SWIZZLE_GEN_VEC2_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE, GLM_MUTABLE, C, A) \
    GLM_SWIZZLE_GEN_VEC2_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE, GLM_MUTABLE, C, B)

#define GLM_SWIZZLE_GEN_REF3_FROM_VEC3_SWIZZLE(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE, A, B, C) \
    GLM_SWIZZLE_GEN_VEC3_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE, GLM_MUTABLE, A, B, C) \
```

```
187     GLM_SWIZZLE_GEN_VEC3_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  6 GLM_MUTABLE, A, C, B) \
187     GLM_SWIZZLE_GEN_VEC3_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  7 GLM_MUTABLE, B, A, C) \
187     GLM_SWIZZLE_GEN_VEC3_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  8 GLM_MUTABLE, B, C, A) \
187     GLM_SWIZZLE_GEN_VEC3_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  9 GLM_MUTABLE, C, A, B) \
188     GLM_SWIZZLE_GEN_VEC3_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  0 GLM_MUTABLE, C, B, A)
188
188 #define GLM_SWIZZLE_GEN_REF_FROM_VEC3_COMP(TMPL_TYPE, CLASS_TYPE,
  2 SWIZZLED_VEC2_TYPE, SWIZZLED_VEC3_TYPE, A, B, C) \
188     GLM_SWIZZLE_GEN_REF3_FROM_VEC3_SWIZZLE(TMPL_TYPE, CLASS_TYPE,
  3 SWIZZLED_VEC3_TYPE, A, B, C) \
188     GLM_SWIZZLE_GEN_REF2_FROM_VEC3_SWIZZLE(TMPL_TYPE, CLASS_TYPE,
  4 SWIZZLED_VEC2_TYPE, A, B, C)
188
188 #define GLM_SWIZZLE_GEN_REF_FROM_VEC3(TMPL_TYPE, CLASS_TYPE,
  6 SWIZZLED_VEC2_TYPE, SWIZZLED_VEC3_TYPE) \
188     GLM_SWIZZLE_GEN_REF_FROM_VEC3_COMP(TMPL_TYPE, CLASS_TYPE,
  7 SWIZZLED_VEC2_TYPE, SWIZZLED_VEC3_TYPE, x, y, z) \
188     GLM_SWIZZLE_GEN_REF_FROM_VEC3_COMP(TMPL_TYPE, CLASS_TYPE,
  8 SWIZZLED_VEC2_TYPE, SWIZZLED_VEC3_TYPE, r, g, b) \
188     GLM_SWIZZLE_GEN_REF_FROM_VEC3_COMP(TMPL_TYPE, CLASS_TYPE,
  9 SWIZZLED_VEC2_TYPE, SWIZZLED_VEC3_TYPE, s, t, q)
189
189 //GLM_SWIZZLE_GEN_REF_FROM_VEC3(valType, detail::vec3, detail::ref2,
  1 detail::ref3)
189
189 #define GLM_SWIZZLE_GEN_REF2_FROM_VEC4_SWIZZLE(TMPL_TYPE,
  3 CLASS_TYPE, SWIZZLED_TYPE, A, B, C, D) \
189     GLM_SWIZZLE_GEN_VEC2_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  4 GLM_MUTABLE, A, B) \
189     GLM_SWIZZLE_GEN_VEC2_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  5 GLM_MUTABLE, A, C) \
189     GLM_SWIZZLE_GEN_VEC2_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  6 GLM_MUTABLE, A, D) \
189     GLM_SWIZZLE_GEN_VEC2_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  7 GLM_MUTABLE, B, A) \
```

```
189     GLM_SWIZZLE_GEN_VEC2_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  8 GLM_MUTABLE, B, C) \
189     GLM_SWIZZLE_GEN_VEC2_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  9 GLM_MUTABLE, B, D) \
190     GLM_SWIZZLE_GEN_VEC2_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  0 GLM_MUTABLE, C, A) \
190     GLM_SWIZZLE_GEN_VEC2_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  1 GLM_MUTABLE, C, B) \
190     GLM_SWIZZLE_GEN_VEC2_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  2 GLM_MUTABLE, C, D) \
190     GLM_SWIZZLE_GEN_VEC2_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  3 GLM_MUTABLE, D, A) \
190     GLM_SWIZZLE_GEN_VEC2_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  4 GLM_MUTABLE, D, B) \
190     GLM_SWIZZLE_GEN_VEC2_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  5 GLM_MUTABLE, D, C)
190
196 #define GLM_SWIZZLE_GEN_REF3_FROM_VEC4_SWIZZLE(TMPL_TYPE,
  7 CLASS_TYPE, SWIZZLED_TYPE, A, B, C, D) \
190     GLM_SWIZZLE_GEN_VEC3_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  8 , A, B, C) \
190     GLM_SWIZZLE_GEN_VEC3_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  9 , A, B, D) \
191     GLM_SWIZZLE_GEN_VEC3_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  0 , A, C, B) \
191     GLM_SWIZZLE_GEN_VEC3_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  1 , A, C, D) \
191     GLM_SWIZZLE_GEN_VEC3_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  2 , A, D, B) \
191     GLM_SWIZZLE_GEN_VEC3_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  3 , A, D, C) \
191     GLM_SWIZZLE_GEN_VEC3_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  4 , B, A, C) \
191     GLM_SWIZZLE_GEN_VEC3_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  5 , B, A, D) \
191     GLM_SWIZZLE_GEN_VEC3_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  6 , B, C, A) \
191     GLM_SWIZZLE_GEN_VEC3_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  7 , B, C, D) \
```

```
191     GLM_SWIZZLE_GEN_VEC3_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  8 , B, D, A) \
191     GLM_SWIZZLE_GEN_VEC3_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  9 , B, D, C) \
192     GLM_SWIZZLE_GEN_VEC3_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  0 , C, A, B) \
192     GLM_SWIZZLE_GEN_VEC3_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  1 , C, A, D) \
192     GLM_SWIZZLE_GEN_VEC3_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  2 , C, B, A) \
192     GLM_SWIZZLE_GEN_VEC3_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  3 , C, B, D) \
192     GLM_SWIZZLE_GEN_VEC3_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  4 , C, D, A) \
192     GLM_SWIZZLE_GEN_VEC3_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  5 , C, D, B) \
192     GLM_SWIZZLE_GEN_VEC3_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  6 , D, A, B) \
192     GLM_SWIZZLE_GEN_VEC3_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  7 , D, A, C) \
192     GLM_SWIZZLE_GEN_VEC3_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  8 , D, B, A) \
192     GLM_SWIZZLE_GEN_VEC3_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  9 , D, B, C) \
193     GLM_SWIZZLE_GEN_VEC3_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  0 , D, C, A) \
193     GLM_SWIZZLE_GEN_VEC3_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  1 , D, C, B)
193
192 #define GLM_SWIZZLE_GEN_REF4_FROM_VEC4_SWIZZLE(TMPL_TYPE,
  3 CLASS_TYPE, SWIZZLED_TYPE, A, B, C, D) \
193     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  4 , A, C, B, D) \
193     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  5 , A, C, D, B) \
193     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  6 , A, D, B, C) \
193     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  7 , A, D, C, B) \
```

```
193     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  8 , A, B, D, C) \
193     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  9 , A, B, C, D) \
194     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  0 , B, C, A, D) \
194     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  1 , B, C, D, A) \
194     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  2 , B, D, A, C) \
194     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  3 , B, D, C, A) \
194     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  4 , B, A, D, C) \
194     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  5 , B, A, C, D) \
194     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  6 , C, B, A, D) \
194     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  7 , C, B, D, A) \
194     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  8 , C, D, A, B) \
194     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  9 , C, D, B, A) \
195     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  0 , C, A, D, B) \
195     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  1 , C, A, B, D) \
195     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  2 , D, C, B, A) \
195     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  3 , D, C, A, B) \
195     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  4 , D, A, B, C) \
195     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  5 , D, A, C, B) \
195     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  6 , D, B, A, C) \
195     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  7 , D, B, C, A)
```

```
195
198 #define GLM_SWIZZLE_GEN_REF_FROM_VEC4_COMP(TMPL_TYPE, CLASS_TYPE,
  9 SWIZZLED_VEC2_TYPE, SWIZZLED_VEC3_TYPE, SWIZZLED_VEC4_TYPE, A, B, C,
    D) \
196     GLM_SWIZZLE_GEN_REF2_FROM_VEC4_SWIZZLE(TMPL_TYPE, CLASS_TYPE,
  0 SWIZZLED_VEC2_TYPE, A, B, C, D) \
196     GLM_SWIZZLE_GEN_REF3_FROM_VEC4_SWIZZLE(TMPL_TYPE, CLASS_TYPE,
  1 SWIZZLED_VEC3_TYPE, A, B, C, D) \
196     GLM_SWIZZLE_GEN_REF4_FROM_VEC4_SWIZZLE(TMPL_TYPE, CLASS_TYPE,
  2 SWIZZLED_VEC4_TYPE, A, B, C, D)
196
196 #define GLM_SWIZZLE_GEN_REF_FROM_VEC4(TMPL_TYPE, CLASS_TYPE,
  4 SWIZZLED_VEC2_TYPE, SWIZZLED_VEC3_TYPE, SWIZZLED_VEC4_TYPE) \
196     GLM_SWIZZLE_GEN_REF_FROM_VEC4_COMP(TMPL_TYPE, CLASS_TYPE,
  5 SWIZZLED_VEC2_TYPE, SWIZZLED_VEC3_TYPE, SWIZZLED_VEC4_TYPE, x, y, z,
    w) \
196     GLM_SWIZZLE_GEN_REF_FROM_VEC4_COMP(TMPL_TYPE, CLASS_TYPE,
  6 SWIZZLED_VEC2_TYPE, SWIZZLED_VEC3_TYPE, SWIZZLED_VEC4_TYPE, r, g, b,
    a) \
196     GLM_SWIZZLE_GEN_REF_FROM_VEC4_COMP(TMPL_TYPE, CLASS_TYPE,
  7 SWIZZLED_VEC2_TYPE, SWIZZLED_VEC3_TYPE, SWIZZLED_VEC4_TYPE, s, t, q,
    p)
196
196 //GLM_SWIZZLE_GEN_REF_FROM_VEC4(valType, detail::vec4, detail::ref2,
  9 detail::ref3, detail::ref4)
197
197 #define GLM_SWIZZLE_GEN_VEC2_FROM_VEC2_SWIZZLE(TMPL_TYPE,
  1 CLASS_TYPE, SWIZZLED_TYPE, A, B) \
197     GLM_SWIZZLE_GEN_VEC2_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  2 const, A, A) \
197     GLM_SWIZZLE_GEN_VEC2_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  3 const, A, B) \
197     GLM_SWIZZLE_GEN_VEC2_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  4 const, B, A) \
197     GLM_SWIZZLE_GEN_VEC2_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  5 const, B, B)
197
197 #define GLM_SWIZZLE_GEN_VEC3_FROM_VEC2_SWIZZLE(TMPL_TYPE,
  7 CLASS_TYPE, SWIZZLED_TYPE, A, B) \
```

```
197     GLM_SWIZZLE_GEN_VEC3_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  8 const, A, A, A) \
197     GLM_SWIZZLE_GEN_VEC3_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  9 const, A, A, B) \
198     GLM_SWIZZLE_GEN_VEC3_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  0 const, A, B, A) \
198     GLM_SWIZZLE_GEN_VEC3_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  1 const, A, B, B) \
198     GLM_SWIZZLE_GEN_VEC3_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  2 const, B, A, A) \
198     GLM_SWIZZLE_GEN_VEC3_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  3 const, B, A, B) \
198     GLM_SWIZZLE_GEN_VEC3_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  4 const, B, B, A) \
198     GLM_SWIZZLE_GEN_VEC3_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  5 const, B, B, B)
198
196 #define GLM_SWIZZLE_GEN_VEC4_FROM_VEC2_SWIZZLE(TMPL_TYPE,
  7 CLASS_TYPE, SWIZZLED_TYPE, A, B) \
198     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  8 const, A, A, A, A) \
198     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  9 const, A, A, A, B) \
199     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  0 const, A, A, B, A) \
199     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  1 const, A, A, B, B) \
199     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  2 const, A, B, A, A) \
199     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  3 const, A, B, A, B) \
199     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  4 const, A, B, B, A) \
199     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  5 const, A, B, B, B) \
199     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  6 const, B, A, A, A) \
199     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  7 const, B, A, A, B) \
```

```
199     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  8 const, B, A, B, A) \
199     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  9 const, B, A, B, B) \
200     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  0 const, B, B, A, A) \
200     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  1 const, B, B, A, B) \
200     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  2 const, B, B, B, A) \
200     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  3 const, B, B, B, B)
200
204 #define GLM_SWIZZLE_GEN_VEC_FROM_VEC2_COMP(TMPL_TYPE, CLASS_TYPE,
  5 SWIZZLED_VEC2_TYPE, SWIZZLED_VEC3_TYPE, SWIZZLED_VEC4_TYPE, A, B) \
200     GLM_SWIZZLE_GEN_VEC2_FROM_VEC2_SWIZZLE(TMPL_TYPE, CLASS_TYPE,
  6 SWIZZLED_VEC2_TYPE, A, B) \
200     GLM_SWIZZLE_GEN_VEC3_FROM_VEC2_SWIZZLE(TMPL_TYPE, CLASS_TYPE,
  7 SWIZZLED_VEC3_TYPE, A, B) \
200     GLM_SWIZZLE_GEN_VEC4_FROM_VEC2_SWIZZLE(TMPL_TYPE, CLASS_TYPE,
  8 SWIZZLED_VEC4_TYPE, A, B)
200
209 #define GLM_SWIZZLE_GEN_VEC_FROM_VEC2(TMPL_TYPE, CLASS_TYPE,
  0 SWIZZLED_VEC2_TYPE, SWIZZLED_VEC3_TYPE, SWIZZLED_VEC4_TYPE) \
201     GLM_SWIZZLE_GEN_VEC_FROM_VEC2_COMP(TMPL_TYPE, CLASS_TYPE,
  1 SWIZZLED_VEC2_TYPE, SWIZZLED_VEC3_TYPE, SWIZZLED_VEC4_TYPE, x, y) \
201     GLM_SWIZZLE_GEN_VEC_FROM_VEC2_COMP(TMPL_TYPE, CLASS_TYPE,
  2 SWIZZLED_VEC2_TYPE, SWIZZLED_VEC3_TYPE, SWIZZLED_VEC4_TYPE, r, g) \
201     GLM_SWIZZLE_GEN_VEC_FROM_VEC2_COMP(TMPL_TYPE, CLASS_TYPE,
  3 SWIZZLED_VEC2_TYPE, SWIZZLED_VEC3_TYPE, SWIZZLED_VEC4_TYPE, s, t)
201
214 //GLM_SWIZZLE_GEN_VEC_FROM_VEC2(valType, detail::vec2, detail::vec2,
  5 detail::vec3, detail::vec4)
201
216 #define GLM_SWIZZLE_GEN_VEC2_FROM_VEC3_SWIZZLE(TMPL_TYPE,
  7 CLASS_TYPE, SWIZZLED_TYPE, A, B, C) \
201     GLM_SWIZZLE_GEN_VEC2_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  8 const, A, A) \
201     GLM_SWIZZLE_GEN_VEC2_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  9 const, A, B) \
```

```
202     GLM_SWIZZLE_GEN_VEC2_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  0 const, A, C) \
202     GLM_SWIZZLE_GEN_VEC2_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  1 const, B, A) \
202     GLM_SWIZZLE_GEN_VEC2_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  2 const, B, B) \
202     GLM_SWIZZLE_GEN_VEC2_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  3 const, B, C) \
202     GLM_SWIZZLE_GEN_VEC2_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  4 const, C, A) \
202     GLM_SWIZZLE_GEN_VEC2_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  5 const, C, B) \
202     GLM_SWIZZLE_GEN_VEC2_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  6 const, C, C)
202
207 #define GLM_SWIZZLE_GEN_VEC3_FROM_VEC3_SWIZZLE(TMPL_TYPE,
  8 CLASS_TYPE, SWIZZLED_TYPE, A, B, C) \
202     GLM_SWIZZLE_GEN_VEC3_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  9 const, A, A, A) \
203     GLM_SWIZZLE_GEN_VEC3_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  0 const, A, A, B) \
203     GLM_SWIZZLE_GEN_VEC3_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  1 const, A, A, C) \
203     GLM_SWIZZLE_GEN_VEC3_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  2 const, A, B, A) \
203     GLM_SWIZZLE_GEN_VEC3_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  3 const, A, B, B) \
203     GLM_SWIZZLE_GEN_VEC3_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  4 const, A, B, C) \
203     GLM_SWIZZLE_GEN_VEC3_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  5 const, A, C, A) \
203     GLM_SWIZZLE_GEN_VEC3_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  6 const, A, C, B) \
203     GLM_SWIZZLE_GEN_VEC3_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  7 const, A, C, C) \
203     GLM_SWIZZLE_GEN_VEC3_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  8 const, B, A, A) \
203     GLM_SWIZZLE_GEN_VEC3_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  9 const, B, A, B) \
```

```
204     GLM_SWIZZLE_GEN_VEC3_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  0 const, B, A, C) \
204     GLM_SWIZZLE_GEN_VEC3_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  1 const, B, B, A) \
204     GLM_SWIZZLE_GEN_VEC3_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  2 const, B, B, B) \
204     GLM_SWIZZLE_GEN_VEC3_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  3 const, B, B, C) \
204     GLM_SWIZZLE_GEN_VEC3_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  4 const, B, C, A) \
204     GLM_SWIZZLE_GEN_VEC3_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  5 const, B, C, B) \
204     GLM_SWIZZLE_GEN_VEC3_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  6 const, B, C, C) \
204     GLM_SWIZZLE_GEN_VEC3_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  7 const, C, A, A) \
204     GLM_SWIZZLE_GEN_VEC3_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  8 const, C, A, B) \
204     GLM_SWIZZLE_GEN_VEC3_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  9 const, C, A, C) \
205     GLM_SWIZZLE_GEN_VEC3_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  0 const, C, B, A) \
205     GLM_SWIZZLE_GEN_VEC3_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  1 const, C, B, B) \
205     GLM_SWIZZLE_GEN_VEC3_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  2 const, C, B, C) \
205     GLM_SWIZZLE_GEN_VEC3_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  3 const, C, C, A) \
205     GLM_SWIZZLE_GEN_VEC3_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  4 const, C, C, B) \
205     GLM_SWIZZLE_GEN_VEC3_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  5 const, C, C, C)
205
206 #define GLM_SWIZZLE_GEN_VEC4_FROM_VEC3_SWIZZLE(TMPL_TYPE,
  7 CLASS_TYPE, SWIZZLED_TYPE, A, B, C) \
205     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  8 const, A, A, A, A) \
205     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  9 const, A, A, A, B) \
```

```
206     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  0 const, A, A, A, C) \
206     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  1 const, A, A, B, A) \
206     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  2 const, A, A, B, B) \
206     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  3 const, A, A, B, C) \
206     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  4 const, A, A, C, A) \
206     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  5 const, A, A, C, B) \
206     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  6 const, A, A, C, C) \
206     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  7 const, A, B, A, A) \
206     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  8 const, A, B, A, B) \
206     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  9 const, A, B, A, C) \
207     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  0 const, A, B, B, A) \
207     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  1 const, A, B, B, B) \
207     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  2 const, A, B, B, C) \
207     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  3 const, A, B, C, A) \
207     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  4 const, A, B, C, B) \
207     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  5 const, A, B, C, C) \
207     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  6 const, A, C, A, A) \
207     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  7 const, A, C, A, B) \
207     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  8 const, A, C, A, C) \
207     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  9 const, A, C, B, A) \
```

```
208      GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  0  const, A, C, B, B) \
208      GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  1  const, A, C, B, C) \
208      GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  2  const, A, C, C, A) \
208      GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  3  const, A, C, C, B) \
208      GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  4  const, A, C, C, C) \
208      GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  5  const, A, D, A, A) \
208      GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  6  const, A, D, A, B) \
208      GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  7  const, A, D, A, C) \
208      GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  8  const, A, D, B, A) \
208      GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  9  const, A, D, B, B) \
209      GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  0  const, A, D, B, C) \
209      GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  1  const, A, D, C, A) \
209      GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  2  const, A, D, C, B) \
209      GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  3  const, A, D, C, C) \
209      GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  4  const, B, A, A, A) \
209      GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  5  const, B, A, A, B) \
209      GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  6  const, B, A, A, C) \
209      GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  7  const, B, A, B, A) \
209      GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  8  const, B, A, B, B) \
209      GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  9  const, B, A, B, C) \
```

```
210      GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  0  const, B, A, C, A) \
210      GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  1  const, B, A, C, B) \
210      GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  2  const, B, A, C, C) \
210      GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  3  const, B, B, A, A) \
210      GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  4  const, B, B, A, B) \
210      GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  5  const, B, B, A, C) \
210      GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  6  const, B, B, B, A) \
210      GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  7  const, B, B, B, B) \
210      GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  8  const, B, B, B, C) \
210      GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  9  const, B, B, C, A) \
211      GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  0  const, B, B, C, B) \
211      GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  1  const, B, B, C, C) \
211      GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  2  const, B, C, A, A) \
211      GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  3  const, B, C, A, B) \
211      GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  4  const, B, C, A, C) \
211      GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  5  const, B, C, B, A) \
211      GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  6  const, B, C, B, B) \
211      GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  7  const, B, C, B, C) \
211      GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  8  const, B, C, C, A) \
211      GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  9  const, B, C, C, B) \
```

```
212     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  0 const, B, C, C, C) \
212     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  1 const, B, D, A, A) \
212     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  2 const, B, D, A, B) \
212     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  3 const, B, D, A, C) \
212     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  4 const, B, D, B, A) \
212     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  5 const, B, D, B, B) \
212     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  6 const, B, D, B, C) \
212     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  7 const, B, D, C, A) \
212     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  8 const, B, D, C, B) \
212     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  9 const, B, D, C, C) \
213     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  0 const, C, A, A, A) \
213     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  1 const, C, A, A, B) \
213     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  2 const, C, A, A, C) \
213     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  3 const, C, A, B, A) \
213     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  4 const, C, A, B, B) \
213     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  5 const, C, A, B, C) \
213     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  6 const, C, A, C, A) \
213     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  7 const, C, A, C, B) \
213     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  8 const, C, A, C, C) \
213     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  9 const, C, B, A, A) \
```

```
214      GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  0  const, C, B, A, B) \
214      GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  1  const, C, B, A, C) \
214      GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  2  const, C, B, B, A) \
214      GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  3  const, C, B, B, B) \
214      GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  4  const, C, B, B, C) \
214      GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  5  const, C, B, C, A) \
214      GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  6  const, C, B, C, B) \
214      GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  7  const, C, B, C, C) \
214      GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  8  const, C, C, A, A) \
214      GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  9  const, C, C, A, B) \
215      GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  0  const, C, C, A, C) \
215      GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  1  const, C, C, B, A) \
215      GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  2  const, C, C, B, B) \
215      GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  3  const, C, C, B, C) \
215      GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  4  const, C, C, C, A) \
215      GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  5  const, C, C, C, B) \
215      GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  6  const, C, C, C, C) \
215      GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  7  const, C, D, A, A) \
215      GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  8  const, C, D, A, B) \
215      GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  9  const, C, D, A, C) \
```

```
216     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  0 const, C, D, B, A) \
216     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  1 const, C, D, B, B) \
216     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  2 const, C, D, B, C) \
216     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  3 const, C, D, C, A) \
216     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  4 const, C, D, C, B) \
216     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  5 const, C, D, C, C) \
216     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  6 const, D, A, A, A) \
216     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  7 const, D, A, A, B) \
216     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  8 const, D, A, A, C) \
216     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  9 const, D, A, B, A) \
217     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  0 const, D, A, B, B) \
217     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  1 const, D, A, B, C) \
217     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  2 const, D, A, C, A) \
217     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  3 const, D, A, C, B) \
217     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  4 const, D, A, C, C) \
217     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  5 const, D, B, A, A) \
217     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  6 const, D, B, A, B) \
217     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  7 const, D, B, A, C) \
217     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  8 const, D, B, B, A) \
217     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  9 const, D, B, B, B) \
```

```
218      GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  0  const, D, B, B, C) \
218      GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  1  const, D, B, C, A) \
218      GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  2  const, D, B, C, B) \
218      GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  3  const, D, B, C, C) \
218      GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  4  const, D, C, A, A) \
218      GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  5  const, D, C, A, B) \
218      GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  6  const, D, C, A, C) \
218      GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  7  const, D, C, B, A) \
218      GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  8  const, D, C, B, B) \
218      GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  9  const, D, C, B, C) \
219      GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  0  const, D, C, C, A) \
219      GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  1  const, D, C, C, B) \
219      GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  2  const, D, C, C, C) \
219      GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  3  const, D, D, A, A) \
219      GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  4  const, D, D, A, B) \
219      GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  5  const, D, D, A, C) \
219      GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  6  const, D, D, B, A) \
219      GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  7  const, D, D, B, B) \
219      GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  8  const, D, D, B, C) \
219      GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  9  const, D, D, C, A) \
```

```
2200     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  0  const, D, D, C, B) \
2201     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  1  const, D, D, C, C)
2202
2202  #define GLM_SWIZZLE_GEN_VEC_FROM_VEC3_COMP(TMPL_TYPE, CLASS_TYPE,
  3  SWIZZLED_VEC2_TYPE, SWIZZLED_VEC3_TYPE, SWIZZLED_VEC4_TYPE, A, B, C)
     \
2204     GLM_SWIZZLE_GEN_VEC2_FROM_VEC3_SWIZZLE(TMPL_TYPE, CLASS_TYPE,
  4  SWIZZLED_VEC2_TYPE, A, B, C) \
2205     GLM_SWIZZLE_GEN_VEC3_FROM_VEC3_SWIZZLE(TMPL_TYPE, CLASS_TYPE,
  5  SWIZZLED_VEC3_TYPE, A, B, C) \
2206     GLM_SWIZZLE_GEN_VEC4_FROM_VEC3_SWIZZLE(TMPL_TYPE, CLASS_TYPE,
  6  SWIZZLED_VEC4_TYPE, A, B, C)
2207
2207  #define GLM_SWIZZLE_GEN_VEC_FROM_VEC3(TMPL_TYPE, CLASS_TYPE,
  8  SWIZZLED_VEC2_TYPE, SWIZZLED_VEC3_TYPE, SWIZZLED_VEC4_TYPE) \
2209     GLM_SWIZZLE_GEN_VEC_FROM_VEC3_COMP(TMPL_TYPE, CLASS_TYPE,
  9  SWIZZLED_VEC2_TYPE, SWIZZLED_VEC3_TYPE, SWIZZLED_VEC4_TYPE, x, y, z)
     \
2210     GLM_SWIZZLE_GEN_VEC_FROM_VEC3_COMP(TMPL_TYPE, CLASS_TYPE,
  0  SWIZZLED_VEC2_TYPE, SWIZZLED_VEC3_TYPE, SWIZZLED_VEC4_TYPE, r, g, b)
     \
2211     GLM_SWIZZLE_GEN_VEC_FROM_VEC3_COMP(TMPL_TYPE, CLASS_TYPE,
  1  SWIZZLED_VEC2_TYPE, SWIZZLED_VEC3_TYPE, SWIZZLED_VEC4_TYPE, s, t, q)
2212
2212  //GLM_SWIZZLE_GEN_VEC_FROM_VEC3(valType, detail::vec3, detail::vec2,
  3  detail::vec3, detail::vec4)
2214
2214  #define GLM_SWIZZLE_GEN_VEC2_FROM_VEC4_SWIZZLE(TMPL_TYPE,
  5  CLASS_TYPE, SWIZZLED_TYPE, A, B, C, D) \
2216     GLM_SWIZZLE_GEN_VEC2_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  6  const, A, A) \
2217     GLM_SWIZZLE_GEN_VEC2_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  7  const, A, B) \
2218     GLM_SWIZZLE_GEN_VEC2_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  8  const, A, C) \
2219     GLM_SWIZZLE_GEN_VEC2_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  9  const, A, D) \
```

```
222     GLM_SWIZZLE_GEN_VEC2_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  0 const, B, A) \
222     GLM_SWIZZLE_GEN_VEC2_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  1 const, B, B) \
222     GLM_SWIZZLE_GEN_VEC2_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  2 const, B, C) \
222     GLM_SWIZZLE_GEN_VEC2_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  3 const, B, D) \
222     GLM_SWIZZLE_GEN_VEC2_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  4 const, C, A) \
222     GLM_SWIZZLE_GEN_VEC2_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  5 const, C, B) \
222     GLM_SWIZZLE_GEN_VEC2_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  6 const, C, C) \
222     GLM_SWIZZLE_GEN_VEC2_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  7 const, C, D) \
222     GLM_SWIZZLE_GEN_VEC2_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  8 const, D, A) \
222     GLM_SWIZZLE_GEN_VEC2_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  9 const, D, B) \
223     GLM_SWIZZLE_GEN_VEC2_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  0 const, D, C) \
223     GLM_SWIZZLE_GEN_VEC2_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  1 const, D, D)
223
223 #define GLM_SWIZZLE_GEN_VEC3_FROM_VEC4_SWIZZLE(TMPL_TYPE,
  3 CLASS_TYPE, SWIZZLED_TYPE, A, B, C, D) \
223     GLM_SWIZZLE_GEN_VEC3_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  4 const, A, A, A) \
223     GLM_SWIZZLE_GEN_VEC3_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  5 const, A, A, B) \
223     GLM_SWIZZLE_GEN_VEC3_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  6 const, A, A, C) \
223     GLM_SWIZZLE_GEN_VEC3_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  7 const, A, A, D) \
223     GLM_SWIZZLE_GEN_VEC3_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  8 const, A, B, A) \
223     GLM_SWIZZLE_GEN_VEC3_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  9 const, A, B, B) \
```

```
2240     GLM_SWIZZLE_GEN_VEC3_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  0  const, A, B, C) \
2240     GLM_SWIZZLE_GEN_VEC3_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  1  const, A, B, D) \
2240     GLM_SWIZZLE_GEN_VEC3_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  2  const, A, C, A) \
2240     GLM_SWIZZLE_GEN_VEC3_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  3  const, A, C, B) \
2240     GLM_SWIZZLE_GEN_VEC3_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  4  const, A, C, C) \
2240     GLM_SWIZZLE_GEN_VEC3_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  5  const, A, C, D) \
2240     GLM_SWIZZLE_GEN_VEC3_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  6  const, A, D, A) \
2240     GLM_SWIZZLE_GEN_VEC3_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  7  const, A, D, B) \
2240     GLM_SWIZZLE_GEN_VEC3_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  8  const, A, D, C) \
2240     GLM_SWIZZLE_GEN_VEC3_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  9  const, A, D, D) \
2250     GLM_SWIZZLE_GEN_VEC3_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  0  const, B, A, A) \
2250     GLM_SWIZZLE_GEN_VEC3_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  1  const, B, A, B) \
2250     GLM_SWIZZLE_GEN_VEC3_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  2  const, B, A, C) \
2250     GLM_SWIZZLE_GEN_VEC3_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  3  const, B, A, D) \
2250     GLM_SWIZZLE_GEN_VEC3_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  4  const, B, B, A) \
2250     GLM_SWIZZLE_GEN_VEC3_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  5  const, B, B, B) \
2250     GLM_SWIZZLE_GEN_VEC3_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  6  const, B, B, C) \
2250     GLM_SWIZZLE_GEN_VEC3_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  7  const, B, B, D) \
2250     GLM_SWIZZLE_GEN_VEC3_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  8  const, B, C, A) \
2250     GLM_SWIZZLE_GEN_VEC3_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  9  const, B, C, B) \
```

```
226      GLM_SWIZZLE_GEN_VEC3_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  0  const, B, C, C) \
226      GLM_SWIZZLE_GEN_VEC3_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  1  const, B, C, D) \
226      GLM_SWIZZLE_GEN_VEC3_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  2  const, B, D, A) \
226      GLM_SWIZZLE_GEN_VEC3_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  3  const, B, D, B) \
226      GLM_SWIZZLE_GEN_VEC3_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  4  const, B, D, C) \
226      GLM_SWIZZLE_GEN_VEC3_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  5  const, B, D, D) \
226      GLM_SWIZZLE_GEN_VEC3_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  6  const, C, A, A) \
226      GLM_SWIZZLE_GEN_VEC3_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  7  const, C, A, B) \
226      GLM_SWIZZLE_GEN_VEC3_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  8  const, C, A, C) \
226      GLM_SWIZZLE_GEN_VEC3_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  9  const, C, A, D) \
227      GLM_SWIZZLE_GEN_VEC3_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  0  const, C, B, A) \
227      GLM_SWIZZLE_GEN_VEC3_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  1  const, C, B, B) \
227      GLM_SWIZZLE_GEN_VEC3_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  2  const, C, B, C) \
227      GLM_SWIZZLE_GEN_VEC3_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  3  const, C, B, D) \
227      GLM_SWIZZLE_GEN_VEC3_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  4  const, C, C, A) \
227      GLM_SWIZZLE_GEN_VEC3_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  5  const, C, C, B) \
227      GLM_SWIZZLE_GEN_VEC3_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  6  const, C, C, C) \
227      GLM_SWIZZLE_GEN_VEC3_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  7  const, C, C, D) \
227      GLM_SWIZZLE_GEN_VEC3_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  8  const, C, D, A) \
227      GLM_SWIZZLE_GEN_VEC3_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  9  const, C, D, B) \
```

```
228      GLM_SWIZZLE_GEN_VEC3_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  0  const, C, D, C) \
228      GLM_SWIZZLE_GEN_VEC3_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  1  const, C, D, D) \
228      GLM_SWIZZLE_GEN_VEC3_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  2  const, D, A, A) \
228      GLM_SWIZZLE_GEN_VEC3_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  3  const, D, A, B) \
228      GLM_SWIZZLE_GEN_VEC3_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  4  const, D, A, C) \
228      GLM_SWIZZLE_GEN_VEC3_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  5  const, D, A, D) \
228      GLM_SWIZZLE_GEN_VEC3_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  6  const, D, B, A) \
228      GLM_SWIZZLE_GEN_VEC3_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  7  const, D, B, B) \
228      GLM_SWIZZLE_GEN_VEC3_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  8  const, D, B, C) \
228      GLM_SWIZZLE_GEN_VEC3_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  9  const, D, B, D) \
229      GLM_SWIZZLE_GEN_VEC3_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  0  const, D, C, A) \
229      GLM_SWIZZLE_GEN_VEC3_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  1  const, D, C, B) \
229      GLM_SWIZZLE_GEN_VEC3_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  2  const, D, C, C) \
229      GLM_SWIZZLE_GEN_VEC3_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  3  const, D, C, D) \
229      GLM_SWIZZLE_GEN_VEC3_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  4  const, D, D, A) \
229      GLM_SWIZZLE_GEN_VEC3_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  5  const, D, D, B) \
229      GLM_SWIZZLE_GEN_VEC3_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  6  const, D, D, C) \
229      GLM_SWIZZLE_GEN_VEC3_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  7  const, D, D, D)
229
228  #define GLM_SWIZZLE_GEN_VEC4_FROM_VEC4_SWIZZLE(TMPL_TYPE,
  9  CLASS_TYPE, SWIZZLED_TYPE, A, B, C, D) \
```

```
230      GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  0  const, A, A, A, A) \
230      GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  1  const, A, A, A, B) \
230      GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  2  const, A, A, A, C) \
230      GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  3  const, A, A, A, D) \
230      GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  4  const, A, A, B, A) \
230      GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  5  const, A, A, B, B) \
230      GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  6  const, A, A, B, C) \
230      GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  7  const, A, A, B, D) \
230      GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  8  const, A, A, C, A) \
230      GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  9  const, A, A, C, B) \
231      GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  0  const, A, A, C, C) \
231      GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  1  const, A, A, C, D) \
231      GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  2  const, A, A, D, A) \
231      GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  3  const, A, A, D, B) \
231      GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  4  const, A, A, D, C) \
231      GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  5  const, A, A, D, D) \
231      GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  6  const, A, B, A, A) \
231      GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  7  const, A, B, A, B) \
231      GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  8  const, A, B, A, C) \
231      GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  9  const, A, B, A, D) \
```

```
232     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  0 const, A, B, B, A) \
232     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  1 const, A, B, B, B) \
232     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  2 const, A, B, B, C) \
232     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  3 const, A, B, B, D) \
232     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  4 const, A, B, C, A) \
232     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  5 const, A, B, C, B) \
232     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  6 const, A, B, C, C) \
232     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  7 const, A, B, C, D) \
232     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  8 const, A, B, D, A) \
232     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  9 const, A, B, D, B) \
233     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  0 const, A, B, D, C) \
233     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  1 const, A, B, D, D) \
233     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  2 const, A, C, A, A) \
233     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  3 const, A, C, A, B) \
233     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  4 const, A, C, A, C) \
233     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  5 const, A, C, A, D) \
233     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  6 const, A, C, B, A) \
233     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  7 const, A, C, B, B) \
233     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  8 const, A, C, B, C) \
233     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  9 const, A, C, B, D) \
```

```
234      GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  0  const, A, C, C, A) \
234      GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  1  const, A, C, C, B) \
234      GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  2  const, A, C, C, C) \
234      GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  3  const, A, C, C, D) \
234      GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  4  const, A, C, D, A) \
234      GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  5  const, A, C, D, B) \
234      GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  6  const, A, C, D, C) \
234      GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  7  const, A, C, D, D) \
234      GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  8  const, A, D, A, A) \
234      GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  9  const, A, D, A, B) \
235      GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  0  const, A, D, A, C) \
235      GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  1  const, A, D, A, D) \
235      GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  2  const, A, D, B, A) \
235      GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  3  const, A, D, B, B) \
235      GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  4  const, A, D, B, C) \
235      GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  5  const, A, D, B, D) \
235      GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  6  const, A, D, C, A) \
235      GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  7  const, A, D, C, B) \
235      GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  8  const, A, D, C, C) \
235      GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  9  const, A, D, C, D) \
```

```
236     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  0 const, A, D, D, A) \
236     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  1 const, A, D, D, B) \
236     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  2 const, A, D, D, C) \
236     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  3 const, A, D, D, D) \
236     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  4 const, B, A, A, A) \
236     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  5 const, B, A, A, B) \
236     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  6 const, B, A, A, C) \
236     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  7 const, B, A, A, D) \
236     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  8 const, B, A, B, A) \
236     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  9 const, B, A, B, B) \
237     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  0 const, B, A, B, C) \
237     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  1 const, B, A, B, D) \
237     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  2 const, B, A, C, A) \
237     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  3 const, B, A, C, B) \
237     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  4 const, B, A, C, C) \
237     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  5 const, B, A, C, D) \
237     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  6 const, B, A, D, A) \
237     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  7 const, B, A, D, B) \
237     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  8 const, B, A, D, C) \
237     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  9 const, B, A, D, D) \
```

```
238     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  0 const, B, B, A, A) \
238     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  1 const, B, B, A, B) \
238     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  2 const, B, B, A, C) \
238     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  3 const, B, B, A, D) \
238     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  4 const, B, B, B, A) \
238     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  5 const, B, B, B, B) \
238     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  6 const, B, B, B, C) \
238     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  7 const, B, B, B, D) \
238     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  8 const, B, B, C, A) \
238     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  9 const, B, B, C, B) \
239     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  0 const, B, B, C, C) \
239     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  1 const, B, B, C, D) \
239     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  2 const, B, B, D, A) \
239     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  3 const, B, B, D, B) \
239     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  4 const, B, B, D, C) \
239     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  5 const, B, B, D, D) \
239     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  6 const, B, C, A, A) \
239     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  7 const, B, C, A, B) \
239     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  8 const, B, C, A, C) \
239     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  9 const, B, C, A, D) \
```

```
240     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  0 const, B, C, B, A) \
240     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  1 const, B, C, B, B) \
240     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  2 const, B, C, B, C) \
240     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  3 const, B, C, B, D) \
240     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  4 const, B, C, C, A) \
240     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  5 const, B, C, C, B) \
240     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  6 const, B, C, C, C) \
240     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  7 const, B, C, C, D) \
240     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  8 const, B, C, D, A) \
240     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  9 const, B, C, D, B) \
241     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  0 const, B, C, D, C) \
241     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  1 const, B, C, D, D) \
241     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  2 const, B, D, A, A) \
241     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  3 const, B, D, A, B) \
241     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  4 const, B, D, A, C) \
241     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  5 const, B, D, A, D) \
241     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  6 const, B, D, B, A) \
241     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  7 const, B, D, B, B) \
241     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  8 const, B, D, B, C) \
241     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  9 const, B, D, B, D) \
```

```
242     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  0 const, B, D, C, A) \
242     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  1 const, B, D, C, B) \
242     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  2 const, B, D, C, C) \
242     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  3 const, B, D, C, D) \
242     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  4 const, B, D, D, A) \
242     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  5 const, B, D, D, B) \
242     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  6 const, B, D, D, C) \
242     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  7 const, B, D, D, D) \
242     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  8 const, C, A, A, A) \
242     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  9 const, C, A, A, B) \
243     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  0 const, C, A, A, C) \
243     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  1 const, C, A, A, D) \
243     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  2 const, C, A, B, A) \
243     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  3 const, C, A, B, B) \
243     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  4 const, C, A, B, C) \
243     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  5 const, C, A, B, D) \
243     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  6 const, C, A, C, A) \
243     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  7 const, C, A, C, B) \
243     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  8 const, C, A, C, C) \
243     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  9 const, C, A, C, D) \
```

```
2440     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  0  const, C, A, D, A) \
2441     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  1  const, C, A, D, B) \
2442     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  2  const, C, A, D, C) \
2443     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  3  const, C, A, D, D) \
2444     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  4  const, C, B, A, A) \
2445     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  5  const, C, B, A, B) \
2446     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  6  const, C, B, A, C) \
2447     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  7  const, C, B, A, D) \
2448     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  8  const, C, B, B, A) \
2449     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  9  const, C, B, B, B) \
2450     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  0  const, C, B, B, C) \
2451     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  1  const, C, B, B, D) \
2452     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  2  const, C, B, C, A) \
2453     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  3  const, C, B, C, B) \
2454     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  4  const, C, B, C, C) \
2455     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  5  const, C, B, C, D) \
2456     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  6  const, C, B, D, A) \
2457     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  7  const, C, B, D, B) \
2458     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  8  const, C, B, D, C) \
2459     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  9  const, C, B, D, D) \
```

```
246      GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  0  const, C, C, A, A) \
246      GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  1  const, C, C, A, B) \
246      GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  2  const, C, C, A, C) \
246      GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  3  const, C, C, A, D) \
246      GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  4  const, C, C, B, A) \
246      GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  5  const, C, C, B, B) \
246      GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  6  const, C, C, B, C) \
246      GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  7  const, C, C, B, D) \
246      GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  8  const, C, C, C, A) \
246      GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  9  const, C, C, C, B) \
247      GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  0  const, C, C, C, C) \
247      GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  1  const, C, C, C, D) \
247      GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  2  const, C, C, D, A) \
247      GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  3  const, C, C, D, B) \
247      GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  4  const, C, C, D, C) \
247      GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  5  const, C, C, D, D) \
247      GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  6  const, C, D, A, A) \
247      GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  7  const, C, D, A, B) \
247      GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  8  const, C, D, A, C) \
247      GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  9  const, C, D, A, D) \
```

```
248     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  0 const, C, D, B, A) \
248     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  1 const, C, D, B, B) \
248     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  2 const, C, D, B, C) \
248     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  3 const, C, D, B, D) \
248     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  4 const, C, D, C, A) \
248     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  5 const, C, D, C, B) \
248     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  6 const, C, D, C, C) \
248     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  7 const, C, D, C, D) \
248     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  8 const, C, D, D, A) \
248     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  9 const, C, D, D, B) \
249     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  0 const, C, D, D, C) \
249     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  1 const, C, D, D, D) \
249     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  2 const, D, A, A, A) \
249     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  3 const, D, A, A, B) \
249     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  4 const, D, A, A, C) \
249     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  5 const, D, A, A, D) \
249     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  6 const, D, A, B, A) \
249     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  7 const, D, A, B, B) \
249     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  8 const, D, A, B, C) \
249     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  9 const, D, A, B, D) \
```

```
250     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  0  const, D, A, C, A) \
250     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  1  const, D, A, C, B) \
250     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  2  const, D, A, C, C) \
250     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  3  const, D, A, C, D) \
250     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  4  const, D, A, D, A) \
250     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  5  const, D, A, D, B) \
250     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  6  const, D, A, D, C) \
250     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  7  const, D, A, D, D) \
250     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  8  const, D, B, A, A) \
250     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  9  const, D, B, A, B) \
251     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  0  const, D, B, A, C) \
251     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  1  const, D, B, A, D) \
251     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  2  const, D, B, B, A) \
251     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  3  const, D, B, B, B) \
251     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  4  const, D, B, B, C) \
251     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  5  const, D, B, B, D) \
251     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  6  const, D, B, C, A) \
251     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  7  const, D, B, C, B) \
251     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  8  const, D, B, C, C) \
251     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  9  const, D, B, C, D) \
```

```
252     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  0 const, D, B, D, A) \
252     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  1 const, D, B, D, B) \
252     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  2 const, D, B, D, C) \
252     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  3 const, D, B, D, D) \
252     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  4 const, D, C, A, A) \
252     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  5 const, D, C, A, B) \
252     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  6 const, D, C, A, C) \
252     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  7 const, D, C, A, D) \
252     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  8 const, D, C, B, A) \
252     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  9 const, D, C, B, B) \
253     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  0 const, D, C, B, C) \
253     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  1 const, D, C, B, D) \
253     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  2 const, D, C, C, A) \
253     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  3 const, D, C, C, B) \
253     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  4 const, D, C, C, C) \
253     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  5 const, D, C, C, D) \
253     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  6 const, D, C, D, A) \
253     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  7 const, D, C, D, B) \
253     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  8 const, D, C, D, C) \
253     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  9 const, D, C, D, D) \
```

```
254     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  0 const, D, D, A, A) \
254     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  1 const, D, D, A, B) \
254     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  2 const, D, D, A, C) \
254     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  3 const, D, D, A, D) \
254     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  4 const, D, D, B, A) \
254     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  5 const, D, D, B, B) \
254     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  6 const, D, D, B, C) \
254     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  7 const, D, D, B, D) \
254     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  8 const, D, D, C, A) \
254     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  9 const, D, D, C, B) \
255     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  0 const, D, D, C, C) \
255     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  1 const, D, D, C, D) \
255     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  2 const, D, D, D, A) \
255     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  3 const, D, D, D, B) \
255     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  4 const, D, D, D, C) \
255     GLM_SWIZZLE_GEN_VEC4_ENTRY(TMPL_TYPE, CLASS_TYPE, SWIZZLED_TYPE,
  5 const, D, D, D, D)
255
256 #define GLM_SWIZZLE_GEN_VEC_FROM_VEC4_COMP(TMPL_TYPE, CLASS_TYPE,
  7 SWIZZLED_VEC2_TYPE, SWIZZLED_VEC3_TYPE, SWIZZLED_VEC4_TYPE, A, B, C,
    D) \
255     GLM_SWIZZLE_GEN_VEC2_FROM_VEC4_SWIZZLE(TMPL_TYPE, CLASS_TYPE,
  8 SWIZZLED_VEC2_TYPE, A, B, C, D) \
255     GLM_SWIZZLE_GEN_VEC3_FROM_VEC4_SWIZZLE(TMPL_TYPE, CLASS_TYPE,
  9 SWIZZLED_VEC3_TYPE, A, B, C, D) \
```

```
256     GLM_SWIZZLE_GEN_VEC4_FROM_VEC4_SWIZZLE(TMPL_TYPE, CLASS_TYPE,
  0 SWIZZLED_VEC4_TYPE, A, B, C, D)
256
256 #define GLM_SWIZZLE_GEN_VEC_FROM_VEC4(TMPL_TYPE, CLASS_TYPE,
  2 SWIZZLED_VEC2_TYPE, SWIZZLED_VEC3_TYPE, SWIZZLED_VEC4_TYPE) \
256     GLM_SWIZZLE_GEN_VEC_FROM_VEC4_COMP(TMPL_TYPE, CLASS_TYPE,
  3 SWIZZLED_VEC2_TYPE, SWIZZLED_VEC3_TYPE, SWIZZLED_VEC4_TYPE, x, y, z,
    w) \
256     GLM_SWIZZLE_GEN_VEC_FROM_VEC4_COMP(TMPL_TYPE, CLASS_TYPE,
  4 SWIZZLED_VEC2_TYPE, SWIZZLED_VEC3_TYPE, SWIZZLED_VEC4_TYPE, r, g, b,
    a) \
256     GLM_SWIZZLE_GEN_VEC_FROM_VEC4_COMP(TMPL_TYPE, CLASS_TYPE,
  5 SWIZZLED_VEC2_TYPE, SWIZZLED_VEC3_TYPE, SWIZZLED_VEC4_TYPE, s, t, q,
    p)
256
256 //GLM_SWIZZLE_GEN_VEC_FROM_VEC4(valType, detail::vec4, detail::vec2,
  7 detail::vec3, detail::vec4)
256
258 #endif//glm_core_swizzle_func
259
250 #define VECTORIZE2_VEC(func) \
251     template <typename T> \
252     GLM_FUNC_QUALIFIER detail::tvec2<T> func( \
253         detail::tvec2<T> const & v) \
254     { \
255         return detail::tvec2<T>( \
256             func(v.x), \
257             func(v.y)); \
258     }
259
250 #define VECTORIZE3_VEC(func) \
251     template <typename T> \
252     GLM_FUNC_QUALIFIER detail::tvec3<T> func( \
258         detail::tvec3<T> const & v) \
254     { \
258         return detail::tvec3<T>( \
256             func(v.x), \
257             func(v.y), \
258             func(v.z)); \
259     }
```

```
250
251 #define VECTORIZE4_VEC(func) \
252     template <typename T> \
253     GLM_FUNC_QUALIFIER detail::tvec4<T> func( \
254         detail::tvec4<T> const & v) \
255     { \
256         return detail::tvec4<T>( \
257             func(v.x), \
258             func(v.y), \
259             func(v.z), \
260             func(v.w)); \
261     }
262
263 #define VECTORIZE_VEC(func) \
264     VECTORIZE2_VEC(func) \
265     VECTORIZE3_VEC(func) \
266     VECTORIZE4_VEC(func)
267
268 #define VECTORIZE2_VEC_SCA(func) \
269     template <typename T> \
270     GLM_FUNC_QUALIFIER detail::tvec2<T> func \
271     ( \
272         detail::tvec2<T> const & x,  \
273         typename detail::tvec2<T>::value_type const & y \
274     ) \
275     { \
276         return detail::tvec2<T>( \
277             func(x.x, y), \
278             func(x.y, y)); \
279     }
280
281 #define VECTORIZE3_VEC_SCA(func) \
282     template <typename T> \
283     GLM_FUNC_QUALIFIER detail::tvec3<T> func \
284     ( \
285         detail::tvec3<T> const & x,  \
286         typename detail::tvec3<T>::value_type const & y \
287     ) \
288     { \
289         return detail::tvec3<T>( \
```

```
260             func(x.x, y), \
261             func(x.y, y), \
262             func(x.z, y)); \
263     }
264
265 #define VECTORIZE4_VEC_SCA(func) \
266     template <typename T> \
267     GLM_FUNC_QUALIFIER detail::tvec4<T> func \
268     ( \
269         detail::tvec4<T> const & x,  \
270         typename detail::tvec4<T>::value_type const & y \
271     ) \
272     { \
273         return detail::tvec4<T>( \
274             func(x.x, y), \
275             func(x.y, y), \
276             func(x.z, y), \
277             func(x.w, y)); \
278     }
279
280 #define VECTORIZE_VEC_SCA(func) \
281     VECTORIZE2_VEC_SCA(func) \
282     VECTORIZE3_VEC_SCA(func) \
283     VECTORIZE4_VEC_SCA(func)
284
285 #define VECTORIZE2_VEC_VEC(func) \
286     template <typename T> \
287     GLM_FUNC_QUALIFIER detail::tvec2<T> func \
288     ( \
289         detail::tvec2<T> const & x,  \
290         detail::tvec2<T> const & y \
291     ) \
292     { \
293         return detail::tvec2<T>( \
294             func(x.x, y.x), \
295             func(x.y, y.y)); \
296     }
297
298 #define VECTORIZE3_VEC_VEC(func) \
299     template <typename T> \
```

```
260     GLM_FUNC_QUALIFIER detail::tvec3<T> func \
261     ( \
262         detail::tvec3<T> const & x,  \
263         detail::tvec3<T> const & y \
264     ) \
265     { \
266         return detail::tvec3<T>( \
267             func(x.x, y.x), \
268             func(x.y, y.y), \
269             func(x.z, y.z)); \
270     }
271
272 #define VECTORIZE4_VEC_VEC(func) \
273     template <typename T> \
274     GLM_FUNC_QUALIFIER detail::tvec4<T> func \
275     ( \
276         detail::tvec4<T> const & x,  \
277         detail::tvec4<T> const & y \
278     ) \
279     { \
280         return detail::tvec4<T>( \
281             func(x.x, y.x), \
282             func(x.y, y.y), \
283             func(x.z, y.z), \
284             func(x.w, y.w)); \
285     }
286
287 #define VECTORIZE_VEC_VEC(func) \
288     VECTORIZE2_VEC_VEC(func) \
289     VECTORIZE3_VEC_VEC(func) \
290     VECTORIZE4_VEC_VEC(func)
```