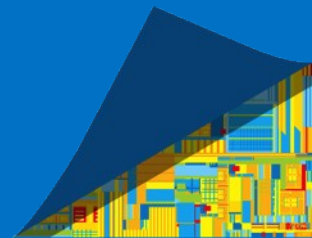




Stochastic Search-Based Testing for Uniform Block Layouts

Ian Romanick <ian.d.romanick@intel.com>

October 8th, 2014



Legal

Copyright © 2014 Intel Corporation. All rights reserved.

*Other names and brands may be claimed as the property of others.

INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH INTEL PRODUCTS. NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT. EXCEPT AS PROVIDED IN INTEL'S TERMS AND CONDITIONS OF SALE FOR SUCH PRODUCTS, INTEL ASSUMES NO LIABILITY WHATSOEVER AND INTEL DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO SALE AND/OR USE OF INTEL PRODUCTS INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT.

A "Mission Critical Application" is any application in which failure of the Intel Product could result, directly or indirectly, in personal injury or death. SHOULD YOU PURCHASE OR USE INTEL'S PRODUCTS FOR ANY SUCH MISSION CRITICAL APPLICATION, YOU SHALL INDEMNIFY AND HOLD INTEL AND ITS SUBSIDIARIES, SUBCONTRACTORS AND AFFILIATES, AND THE DIRECTORS, OFFICERS, AND EMPLOYEES OF EACH, HARMLESS AGAINST ALL CLAIMS COSTS, DAMAGES, AND EXPENSES AND REASONABLE ATTORNEYS' FEES ARISING OUT OF, DIRECTLY OR INDIRECTLY, ANY CLAIM OF PRODUCT LIABILITY, PERSONAL INJURY, OR DEATH ARISING IN ANY WAY OUT OF SUCH MISSION CRITICAL APPLICATION, WHETHER OR NOT INTEL OR ITS SUBCONTRACTOR WAS NEGLIGENT IN THE DESIGN, MANUFACTURE, OR WARNING OF THE INTEL PRODUCT OR ANY OF ITS PARTS.

Intel may make changes to specifications and product descriptions at any time, without notice.

All products, dates, and figures specified are preliminary based on current expectations, and are subject to change without notice.

Intel processors, chipsets, and desktop boards may contain design defects or errors known as errata, which may cause the product to deviate from published specifications. Current characterized errata are available on request.

Any code names featured are used internally within Intel to identify products that are in development and not yet publicly announced for release. Customers, licensees and other third parties are not authorized by Intel to use code names in advertising, promotion or marketing of any product or services and any such use of Intel's internal code names is at the sole risk of the user.

Intel product plans in this presentation do not constitute Intel plan of record product roadmaps. Please contact your Intel representative to obtain Intel's current plan of record product roadmaps.

Performance claims: Software and workloads used in performance tests may have been optimized for performance only on Intel® microprocessors. Performance tests, such as SYSmark and MobileMark, are measured using specific computer systems, components, software, operations and functions. Any change to any of those factors may cause the results to vary. You should consult other information and performance tests to assist you in fully evaluating your contemplated purchases, including the performance of that product when combined with other products. For more information go to <http://www.Intel.com/performance>

Intel, Intel Inside, the Intel logo, Centrino, Intel Core, Intel Atom, Pentium, and Ultrabook are trademarks of Intel Corporation in the United States and other countries.



Introduction

Agenda

- Uniform buffer object background / what's the problem?
- Testing method
- Live demo

UBO std140 packing

Defines an ABI for uniform block layout

- Nine rules for base alignment, structure padding, and array stride

UBO std140 packing

Defines an ABI for uniform block layout

- Nine rules for base alignment, structure padding, and array stride

But implementations are really, *really* bad...

- In July I fixed a lot of bugs found by new Khronos conformance tests
- 12 Mesa commits: `22f7a46d..b48621c3`

UBO std140 packing

“We basically can’t use UBOs.”

- Christophe Riccio (g-truc.net and Unity3D)

“UBO introspection with GL: don’t do it kids. Every driver has its own way of doing things. std140 layout gives no guarantees either.”

- Leonard Ritter (@paniq on Twitter)

UBO std140 packing

Defines an ABI for uniform block layout

- Nine rules for base alignment, structure padding, and array stride

(9) If the member is a structure, the base alignment of the structure is $\langle N \rangle$, where $\langle N \rangle$ is the largest base alignment value of any of its members, and rounded up to the base alignment of a vec4.

```
struct S { float f; int i; };
```

```
uniform U {  
    float f;  
    S s;  
};
```

UBO std140 packing

Defines an ABI for uniform block layout

- Nine rules for base alignment, structure padding, and array stride

(9) If the member is a structure, the base alignment of the structure is $\langle N \rangle$, where $\langle N \rangle$ is the largest base alignment value of any of its members, and **rounded up to the base alignment of a vec4.**

```
struct S { float f; int i; };
```

```
uniform U {  
    float f;  
    S s;      // should be at offset 16  
};
```


Randomized Testing

Generate (semi-)random uniform buffer object

- Each block has a set of required, complex members
- Additional members with basic types fill in around the complex members

Probe all block members

- Every aspect visible to the GL API

Fill UBO with data, probe data in shader

- Not all fields are probed, see [bug #84053](#)
- Exposes a bug in AMD's closed-source driver

Randomized Testing

“Trim” script trims failing test

- Modifications are random, so trimmed test may not be minimal

Results

Found many bugs in open and closed-source drivers

- Bug #83468
- Bug #83506
- Bug #83508
- Bug #83533
- Bug #83639
- Bug #83741
- See white paper for more details

Live Demo

```
idr@mumford-wire:~/devel/graphics/piglit
File Edit View Search Terminal Help
[idr@mumford-wire piglit]$ bash ./random_runs.sh
0x00000014.shader_test
```

```
idr@mumford-wire:~/devel/graphics/piglit
File Edit View Search Terminal Help
[idr@mumford-wire piglit]$ bash ./tria_shader.sh fail/0x00000014.shader_test
S1 field index 9: remove ('vec3', 'fv2')
UB1 field index 1: S4[7] => S4[1]
No progress
UB1 field index 8: remove ('ivec4', 'iv1')
S1 field index 7: remove ('uvec2', 'uv1')
UB1 field index 4: remove ('vec3', 'fv4')
S2 field index 3: remove ('int', 'i1')
UB1 field index 1: S4[7] => S4[1]
No progress
S3 field index 2: remove ('uint', 'u4')
S2 field index 2: remove ('nat3x2', 'n32_1')
S1 field index 1: remove ('uint', 'u2')
S2 field index 1: remove ('vec2', 'fv1')
S2 field index 0: S1[2] => S1[1]
S3 field index 1: remove ('int', 'i3')
UB1 field index 4: remove ('bool', 'b3')
UB1 field index 5: remove ('uvec4', 'uv2')
S4 field index 0: remove ('vec3', 'fv3')
S2 field index 2: remove ('bvec4', 'bv1')
S1 field index 6: remove ('nat4x2', 'n42_2')
S2 field index 3: remove ('nat4x2', 'n42_1')
```

More Information

See the white paper:

<http://www.x.org/wiki/Events/XDC2014/XDC2014RomanickTesting/>

Test scripts:

<http://cgit.freedesktop.org/~idr/piglit/log/?h=ubo-lolz>

