KUnit sorcery and the uncanny nature of FPU in the DRM

Unit tests for the GPU driver

4 Oct 2022

Isabella Basso, Magali Lemes, Maíra Canal, Tales Aparecida

About us

ISABELLA

- Red Hat SWE
- Open source GPU stack Undergrad Researcher
- B.Sc. Molecular Sciences – University of São Paulo/Brazil

MAGALI

- KUnit in the Linux Kernel Undergrad Researcher
- B.Sc. Computer Science - University of São Paulo/Brazil

MAÍRA

- Igalia Coding Experience
- Real-Time Linux Undergrad Researcher
- Computer Engineering - University of São Paulo/Brazil

Thanks

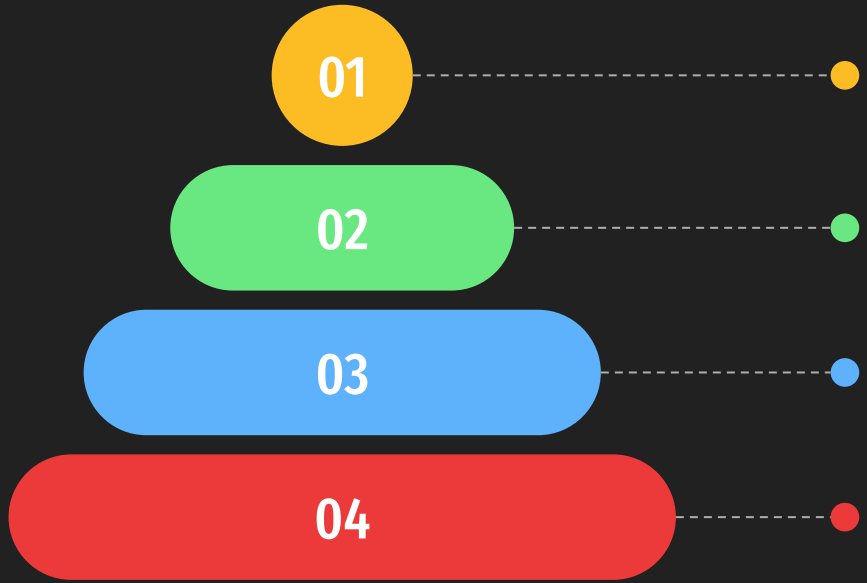
for the GPUs, mentoring, sponsoring, platform, and reviews

- AMD
- Igalia
- KUnit engineers
- DRM community
- X.Org Foundation
- Google Summer of Code
- University of São Paulo

Tests... Tests... Tests...

Are all tests equal? Why introduce unit tests? What is KUnit?

Testing Overview



Manual

Test cases are executed manually by a tester

E2E Tests

Tests the system the same way a real user experiences it - e.g. RenderDoc

Integration Tests

Tests a particular functionality that depends on another functionality - e.g. IGT

Unit Tests

Write tests to a small isolated unit

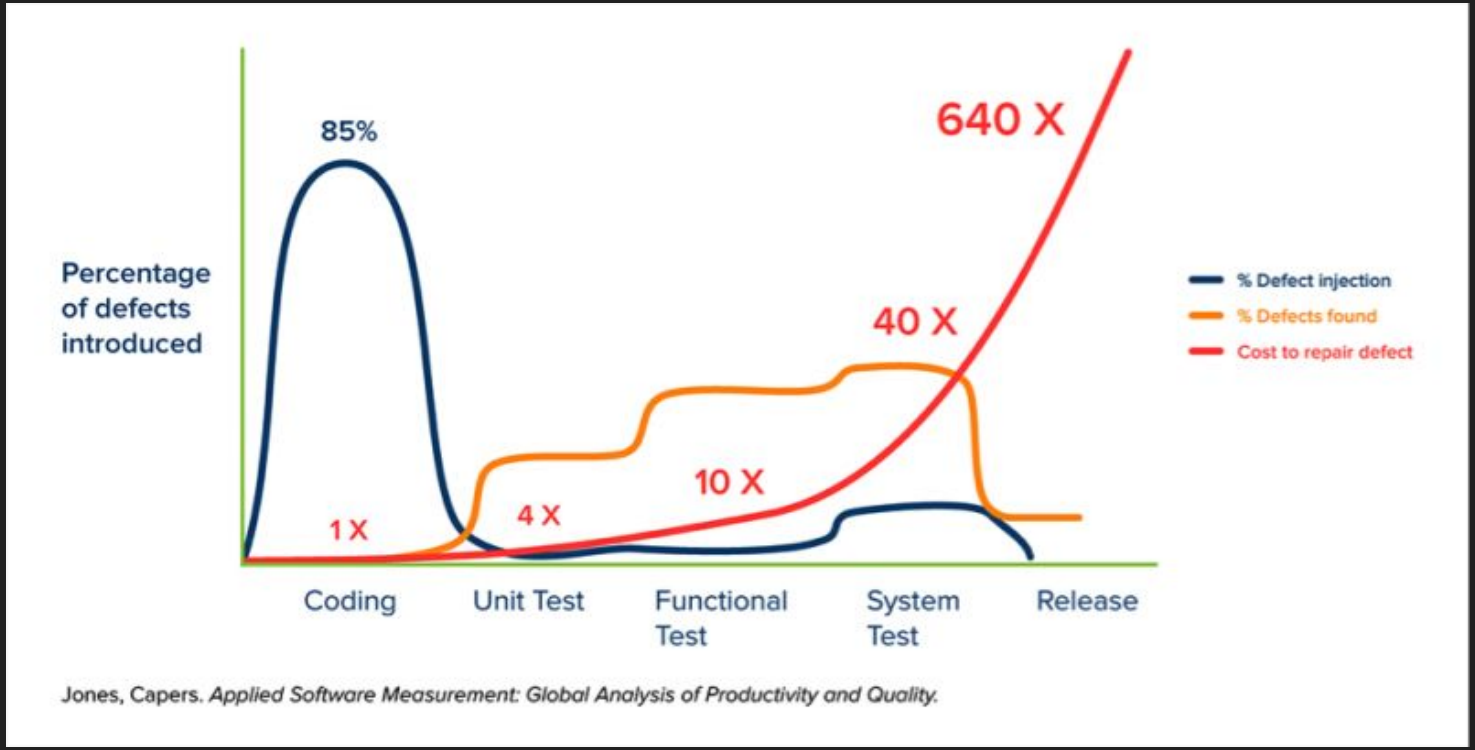
Why introduce unit tests into the Linux kernel?

- Avoid regressions
  - With automation, a good test suite can catch regressions as soon as the programmer pushes changes.
- Expedite janitorial refactoring
  - A developer can manipulate code more freely relying on tests.
  - Encourages modularity, which is easier to test **and maintain.**
- Assert behavior of smaller segments
  - Test a single function instead of a complex behavior.
- There's a framework: **KUnit**

Tradeoffs

- More code to manage
- Good tests demand effort

Business Value



Jones, Capers. *Applied Software Measurement: Global Analysis of Productivity and Quality.*

KUnit - Kernel Unit Testing Framework

More about KUnit

- KUnit in mainline since 2019
  - Created and maintained by Brendan Higgins and David Gow at Google
  - Growing inside various subsystems
- White-box testing
  - Tests can access internal APIs, instead of userspace APIs
- Inspired by user land testing frameworks
  - JUnit, Python’s unittest, and GoogleTest

As a developer, KUnit...

- Allows for testing hardware-agnostic code
  - Hardware is not always accessible and the codebase is not completely dependent on it
  - Run as a userspace process with UML, inside a VM (QEMU) or even bare-metal
- Can be integrated with other platforms
  - Can be integrated into VSCode on Windows WSL
  - Can even be integrated into IGT
- Has useful helper tools
  - Compile and execute tests in a single command: `$ ./kunit.py run`
  - Run just what needs to be tested
- Has little boilerplate
  - Have a test ready in just a few steps.

Minimal code example

Writing a simple KUnit test

Minimal code example

```c
#include "sum_module.h"

int sum (int a, int b)
{
	return a + b;
}
```

Function to be tested

```c
#include <kunit/test.h>
#include "sum_module.h"

static void sum_test (struct kunit *test)
{
	KUNIT_EXPECT_EQ(test, 1, sum(1, 0));
	KUNIT_EXPECT_EQ(test, 2, sum(1, 1));
	KUNIT_EXPECT_EQ(test, 0, sum(-1, 1));
	KUNIT_EXPECT_EQ(test, INT_MAX, sum(0, INT_MAX));
	KUNIT_EXPECT_EQ(test, -1, sum(INT_MAX, INT_MIN));
}

static struct kunit_case sum_test_cases[] = {
	KUNIT_CASE(sum_test),
	{ }
};

static struct kunit_suite sum_test_suite = {
	.name = "sum-example",
	.test_cases = sum_test_cases,
};
kunit_test_suite(sum_test_suite);
```

The test

Minimal code example

```c
#include "sum_module.h"

int sum (int a, int b)
{
	return a + b;
}
```

Function to be tested

```c
#include <kunit/test.h>
#include "sum_module.h"

static void sum_test (struct kunit *test)
{
	KUNIT_EXPECT_EQ(test, 1, sum(1, 0));
	KUNIT_EXPECT_EQ(test, 2, sum(1, 1));
	KUNIT_EXPECT_EQ(test, 0, sum(-1, 1));
	KUNIT_EXPECT_EQ(test, INT_MAX, sum(0, INT_MAX));
	KUNIT_EXPECT_EQ(test, -1, sum(INT_MAX, INT_MIN));
}

static struct kunit_case sum_test_cases[] = {
	KUNIT_CASE(sum_test),
	{ }
};

static struct kunit_suite sum_test_suite = {
	.name = "sum-example",
	.test_cases = sum_test_cases,
};
kunit_test_suite(sum_test_suite);
```

The test

Minimal code example

```c
#include "sum_module.h"

int sum (int a, int b)
{
	return a + b;
}
```

Function to be tested

```c
#include <kunit/test.h>
#include "sum_module.h"

static void sum_test (struct kunit *test)
{
	KUNIT_EXPECT_EQ(test, 1, sum(1, 0));
	KUNIT_EXPECT_EQ(test, 2, sum(1, 1));
	KUNIT_EXPECT_EQ(test, 0, sum(-1, 1));
	KUNIT_EXPECT_EQ(test, INT_MAX, sum(0, INT_MAX));
	KUNIT_EXPECT_EQ(test, -1, sum(INT_MAX, INT_MIN));
}

static struct kunit_case sum_test_cases[] = {
	KUNIT_CASE(sum_test),
	{ }
};

static struct kunit_suite sum_test_suite = {
	.name = "sum-example",
	.test_cases = sum_test_cases,
};
kunit_test_suite(sum_test_suite);
```

The test

Minimal code example

```c
#include "sum_module.h"

int sum (int a, int b)
{
	return a + b;
}
```

Function to be tested

```c
#include <kunit/test.h>
#include "sum_module.h"

static void sum_test (struct kunit *test)
{
	KUNIT_EXPECT_EQ(test, 1, sum(1, 0));
	KUNIT_EXPECT_EQ(test, 2, sum(1, 1));
	KUNIT_EXPECT_EQ(test, 0, sum(-1, 1));
	KUNIT_EXPECT_EQ(test, INT_MAX, sum(0, INT_MAX));
	KUNIT_EXPECT_EQ(test, -1, sum(INT_MAX, INT_MIN));
}

static struct kunit_case sum_test_cases[] = {
	KUNIT_CASE(sum_test),
	{ }
};

static struct kunit_suite sum_test_suite = {
	.name = "sum-example",
	.test_cases = sum_test_cases,
};
kunit_test_suite(sum_test_suite);
```

The test

Minimal code example

```c
#include "sum_module.h"

int sum (int a, int b)
{
	return a + b;
}
```

Function to be tested

```c
#include <kunit/test.h>
#include "sum_module.h"

static void sum_test (struct kunit *test)
{
	KUNIT_EXPECT_EQ(test, 1, sum(1, 0));
	KUNIT_EXPECT_EQ(test, 2, sum(1, 1));
	KUNIT_EXPECT_EQ(test, 0, sum(-1, 1));
	KUNIT_EXPECT_EQ(test, INT_MAX, sum(0, INT_MAX));
	KUNIT_EXPECT_EQ(test, -1, sum(INT_MAX, INT_MIN));
}

static struct kunit_case sum_test_cases[] = {
	KUNIT_CASE(sum_test),
	{ }
};

static struct kunit_suite sum_test_suite = {
	.name = "sum-example",
	.test_cases = sum_test_cases,
};
kunit_test_suite(sum_test_suite);
```

The test

Minimal code example

```c
#include "sum_module.h"

int sum (int a, int b)
{
	return a + b;
}
```

Function to be tested

```c
#include <kunit/test.h>
#include "sum_module.h"

static void sum_test (struct kunit *test)
{
	KUNIT_EXPECT_EQ(test, 1, sum(1, 0));
	KUNIT_EXPECT_EQ(test, 2, sum(1, 1));
	KUNIT_EXPECT_EQ(test, 0, sum(-1, 1));
	KUNIT_EXPECT_EQ(test, INT_MAX, sum(0, INT_MAX));
	KUNIT_EXPECT_EQ(test, -1, sum(INT_MAX, INT_MIN));
}

static struct kunit_case sum_test_cases[] = {
	KUNIT_CASE(sum_test),
	{ }
};

static struct kunit_suite sum_test_suite = {
	.name = "sum-example",
	.test_cases = sum_test_cases,
};
kunit_test_suite(sum_test_suite);
```

The test

Minimal code example

```c
#include "sum_module.h"

int sum (int a, int b)
{
	return a + b;
}
```

Function to be tested

```c
#include <kunit/test.h>
#include "sum_module.h"

static void sum_test (struct kunit *test)
{
	KUNIT_EXPECT_EQ(test, 1, sum(1, 0));
	KUNIT_EXPECT_EQ(test, 2, sum(1, 1));
	KUNIT_EXPECT_EQ(test, 0, sum(-1, 1));
	KUNIT_EXPECT_EQ(test, INT_MAX, sum(0, INT_MAX));
	KUNIT_EXPECT_EQ(test, -1, sum(INT_MAX, INT_MIN));
}

static struct kunit_case sum_test_cases[] = {
	KUNIT_CASE(sum_test),
	{ }
};

static struct kunit_suite sum_test_suite = {
	.name = "sum-example",
	.test_cases = sum_test_cases,
};
kunit_test_suite(sum_test_suite);
```

The test

Case of Study

Practical KUnit Examples on the AMD codebase

Case of study 1
display_rq_dlg_calc_20

- Assure invariant AMD's DML implementation stays unchanged through time.
- The **display_rq_dlg_calc_20** functions have a limited and comprehensible set of inputs and outputs.
- Test suites can avoid accidental changes.

```
static unsigned int get_bytes_per_element (enum
source_format_class source_format, bool is_chroma)
{
	unsigned int ret_val = 0;

	if (source_format == dm_444_16) {
		if (!is_chroma)
			ret_val = 2;
	} else if (source_format == dm_444_32) {
		if (!is_chroma)
			ret_val = 4;
	} else if (source_format == dm_444_64) {
		if (!is_chroma)
			ret_val = 8;
	} else if (source_format == dm_420_8) {
		if (is_chroma)
			ret_val = 2;
		else
			ret_val = 1;
	} else if (source_format == dm_420_10) {
		if (is_chroma)
			ret_val = 4;
		else
			ret_val = 2;
	} else if (source_format == dm_444_8) {
		ret_val = 1;
	}
	return ret_val;
}
```

```c
static void get_bytes_per_element_test(struct kunit *test)
{
	KUNIT_ASSERT_EQ(test, get_bytes_per_element(dm_444_16, false), 2);
	KUNIT_ASSERT_EQ(test, get_bytes_per_element(dm_444_32, false), 4);
	KUNIT_ASSERT_EQ(test, get_bytes_per_element(dm_444_64, false), 8);

	KUNIT_ASSERT_EQ(test, get_bytes_per_element(dm_420_12, false), 0);
	KUNIT_ASSERT_EQ(test, get_bytes_per_element(dm_420_12, true) , 0);

	KUNIT_ASSERT_EQ(test, get_bytes_per_element(dm_444_16, true), 0);
	KUNIT_ASSERT_EQ(test, get_bytes_per_element(dm_444_32, true), 0);
	KUNIT_ASSERT_EQ(test, get_bytes_per_element(dm_444_64, true), 0);

	KUNIT_ASSERT_EQ(test, get_bytes_per_element(dm_420_8, false) , 1);
	KUNIT_ASSERT_EQ(test, get_bytes_per_element(dm_420_10, false), 2);

	KUNIT_ASSERT_EQ(test, get_bytes_per_element(dm_420_8, true) , 2);
	KUNIT_ASSERT_EQ(test, get_bytes_per_element(dm_420_10, true), 4);

	KUNIT_ASSERT_EQ(test, get_bytes_per_element(dm_mono_8, false),
		get_bytes_per_element(dm_444_8, false));
	KUNIT_ASSERT_EQ(test, get_bytes_per_element(dm_mono_16, false),
		get_bytes_per_element(dm_444_16, false));
	KUNIT_ASSERT_EQ(test, get_bytes_per_element(dm_mono_16, true), 0);

	KUNIT_ASSERT_EQ(test, get_bytes_per_element(dm_422_8, false) , 0);
	KUNIT_ASSERT_EQ(test, get_bytes_per_element(dm_422_8, true)  , 0);

	[ ... ]
}
```

[PATCH v2 3/8] drm/amd/display: Introduce KUnit tests to display_rq_dlg_calc_20
https://lore.kernel.org/dri-devel/20220831172239.344446-4-mairacanal@riseup.net/

```c
static unsigned int get_bytes_per_element (enum
source_format_class source_format, bool is_chroma)
{
	unsigned int ret_val = 0;

	if (source_format == dm_444_16) {
		if (!is_chroma)
			ret_val = 2;
	} else if (source_format == dm_444_32) {
		if (!is_chroma)
			ret_val = 4;
	} else if (source_format == dm_444_64) {
		if (!is_chroma)
			ret_val = 8;
	}

	[ ... ]

	return ret_val;
}
```

Function to be tested

```c
[ ... ]

KUNIT_ASSERT_EQ(test, get_bytes_per_element(dm_444_16, false), 2);
KUNIT_ASSERT_EQ(test, get_bytes_per_element(dm_444_32, false), 4);
KUNIT_ASSERT_EQ(test, get_bytes_per_element(dm_444_64, false), 8);

[ ... ]
```

The test

```
enum source_format_class {
        dm_444_16 = 0,

        [ ... ]

        dm_422_8 = 6,

        [ ... ]

        dm_mono_8 = dm_444_8,
        dm_mono_16 = dm_444_16,

        [ ... ]
};
```

```
static void get_bytes_per_element_test(struct kunit *test)
{

        [ ... ]

        KUNIT_ASSERT_EQ(test, get_bytes_per_element(dm_mono_8, false),
                get_bytes_per_element(dm_444_8, false));
        KUNIT_ASSERT_EQ(test, get_bytes_per_element(dm_mono_16, false),
                get_bytes_per_element(dm_444_16, false));
        KUNIT_ASSERT_EQ(test, get_bytes_per_element(dm_mono_16, true), 0);

        KUNIT_ASSERT_EQ(test, get_bytes_per_element(dm_422_8, false) , 0);
        KUNIT_ASSERT_EQ(test, get_bytes_per_element(dm_422_8, true)  , 0);

        [ ... ]
}
```

Case of study 2 - Catching regressions with KUnit

dcn20_fpu

The **`dcn21_update_bw_bounding_box`** case:

How to include unit tests in such development workflow?

01 We first identify a problem

```
void dcn21_update_bw_bounding_box(struct dc *dc, struct clk_bw_params *bw_params)
{
        [ ... ]
                for (i = 0; i < clk_table->num_entries; i++) {
                        /* loop backwards*/
                        for (closest_clk_lvl = 0, j = dcn2_1_soc.num_states - 1; j >= 0; j--) {
                                if ((unsigned int) dcn2_1_soc.clock_limits[j].dcfclk_mhz <= clk_table->entries[i].dcfclk_mhz) {
                                        closest_clk_lvl = j;
                                        break;
                                }
                        }

                        /* clk_table[1] is reserved for min DF PState.  skip here to fill in later. */
                        if (i == 1)
                                k++;

                        dcn2_1_soc.clock_limits[k].state = k;
                        dcn2_1_soc.clock_limits[k].dcfclk_mhz = clk_table->entries[i].dcfclk_mhz;
                        dcn2_1_soc.clock_limits[k].fabricclk_mhz = clk_table->entries[i].fclk_mhz;
                        dcn2_1_soc.clock_limits[k].socclk_mhz = clk_table->entries[i].socclk_mhz;
                        dcn2_1_soc.clock_limits[k].dram_speed_mts = clk_table->entries[i].memclk_mhz * 2;

                        dcn2_1_soc.clock_limits[k].dispclk_mhz = dcn2_1_soc.clock_limits[closest_clk_lvl].dispclk_mhz;
                        dcn2_1_soc.clock_limits[k].dppclk_mhz = dcn2_1_soc.clock_limits[closest_clk_lvl].dppclk_mhz;
                        dcn2_1_soc.clock_limits[k].dram_bw_per_chan_gbps = dcn2_1_soc.clock_limits[closest_clk_lvl].dram_bw_per_chan_gbps;
                        dcn2_1_soc.clock_limits[k].dscclk_mhz = dcn2_1_soc.clock_limits[closest_clk_lvl].dscclk_mhz;
                        dcn2_1_soc.clock_limits[k].dtbclk_mhz = dcn2_1_soc.clock_limits[closest_clk_lvl].dtbclk_mhz;
                        dcn2_1_soc.clock_limits[k].phyclk_d18_mhz = dcn2_1_soc.clock_limits[closest_clk_lvl].phyclk_d18_mhz;
                        dcn2_1_soc.clock_limits[k].phyclk_mhz = dcn2_1_soc.clock_limits[closest_clk_lvl].phyclk_mhz;

                        k++;
                }
        [ ... ]
}
```

[PATCH 5/6] drm/amd/display: Reduce frame size in the bouding box for DCN21

https://lore.kernel.org/amd-gfx/20220603185042.3408844-6-Rodrigo.Siqueira@amd.com/

02 Then we write a test case that we know will fail due to the regression.

```
$ ./tools/testing/kunit/kunit.py run --arch=x86_64 --kunitconfig=drivers/gpu/drm/amd/display/tests 'dml_dcn20_fpu_dcn21_update_bw_bounding_box_test'
[19:02:25] === dml_dcn20_fpu_dcn21_update_bw_bounding_box_test (1 subtest) ===
[19:02:25] ========== dcn21_update_bw_bounding_box_test ==========
[19:02:25] # dcn21_update_bw_bounding_box_test: EXPECTATION FAILED at drivers/gpu/drm/amd/amdgpu/../display/tests/dc/dml/dcn20/dcn20_fpu_test.c:500
[19:02:25] Expected test_param->expected_clock_limits[i].dispclk_mhz == dcn2_1_soc.clock_limits[i].dispclk_mhz, but
[19:02:25] test_param->expected_clock_limits[i].dispclk_mhz == 757
[19:02:25] dcn2_1_soc.clock_limits[i].dispclk_mhz == 600
[19:02:25] # dcn21_update_bw_bounding_box_test: EXPECTATION FAILED at drivers/gpu/drm/amd/amdgpu/../display/tests/dc/dml/dcn20/dcn20_fpu_test.c:502
[19:02:25] Expected test_param->expected_clock_limits[i].dppclk_mhz == dcn2_1_soc.clock_limits[i].dppclk_mhz, but
[19:02:25] test_param->expected_clock_limits[i].dppclk_mhz == 685
[19:02:25] dcn2_1_soc.clock_limits[i].dppclk_mhz == 400
[19:02:25] # dcn21_update_bw_bounding_box_test: EXPECTATION FAILED at drivers/gpu/drm/amd/amdgpu/../display/tests/dc/dml/dcn20/dcn20_fpu_test.c:508
[19:02:25] Expected test_param->expected_clock_limits[i].dscclk_mhz == dcn2_1_soc.clock_limits[i].dscclk_mhz, but
[19:02:25] test_param->expected_clock_limits[i].dscclk_mhz == 287
[19:02:25] dcn2_1_soc.clock_limits[i].dscclk_mhz == 205
[19:02:25] # dcn21_update_bw_bounding_box_test: EXPECTATION FAILED at drivers/gpu/drm/amd/amdgpu/../display/tests/dc/dml/dcn20/dcn20_fpu_test.c:500
[19:02:25] Expected test_param->expected_clock_limits[i].dispclk_mhz == dcn2_1_soc.clock_limits[i].dispclk_mhz, but
[19:02:25] test_param->expected_clock_limits[i].dispclk_mhz == 847
[19:02:25] dcn2_1_soc.clock_limits[i].dispclk_mhz == 600
[19:02:25] # dcn21_update_bw_bounding_box_test: EXPECTATION FAILED at drivers/gpu/drm/amd/amdgpu/../display/tests/dc/dml/dcn20/dcn20_fpu_test.c:502
[19:02:25] Expected test_param->expected_clock_limits[i].dppclk_mhz == dcn2_1_soc.clock_limits[i].dppclk_mhz, but
[19:02:25] test_param->expected_clock_limits[i].dppclk_mhz == 757
[19:02:25] dcn2_1_soc.clock_limits[i].dppclk_mhz == 400
[19:02:25] # dcn21_update_bw_bounding_box_test: EXPECTATION FAILED at drivers/gpu/drm/amd/amdgpu/../display/tests/dc/dml/dcn20/dcn20_fpu_test.c:508
[19:02:25] Expected test_param->expected_clock_limits[i].dscclk_mhz == dcn2_1_soc.clock_limits[i].dscclk_mhz, but
[19:02:25] test_param->expected_clock_limits[i].dscclk_mhz == 318
[19:02:25] dcn2_1_soc.clock_limits[i].dscclk_mhz == 205
[19:02:25] not ok 1 - 5-entry bounding box clocks table
[19:02:25] [FAILED] 5-entry bounding box clocks table
[19:02:25] # Subtest: dcn21_update_bw_bounding_box_test
[19:02:25] not ok 1 - dcn21_update_bw_bounding_box_test
[19:02:25] ========== [FAILED] dcn21_update_bw_bounding_box_test ==========
[19:02:25] # Subtest: dml_dcn20_fpu_dcn21_update_bw_bounding_box_test
[19:02:25] 1..1
[19:02:25] not ok 8 - dml_dcn20_fpu_dcn21_update_bw_bounding_box_test
[19:02:25] = [FAILED] dml_dcn20_fpu_dcn21_update_bw_bounding_box_test =
[19:02:25] ==========================================================
[19:02:25] Testing complete. Passed: 87, Failed: 6, Crashed: 0, Skipped: 0, Errors: 0
```

[PATCH v2 6/8] drm/amd/display: Introduce KUnit tests for dcn20_fpu
https://lore.kernel.org/dri-devel/20220831172239.344446-7-mairacanal@riseup.net/

02 Then we write a test case that we know will fail due to the regression.

```
$ ./tools/testing/kunit/kunit.py run --arch=x86_64 --kunitconfig=drivers/gpu/drm/amd/display/tests 'dml_dcn20_fpu_dcn21_update_bw_bounding_box_test'
[19:02:25] === dml_dcn20_fpu_dcn21_update_bw_bounding_box_test (1 subtest) ===
[19:02:25] ========== dcn21_update_bw_bounding_box_test ==========
[19:02:25] # dcn21_update_bw_bounding_box_test: EXPECTATION FAILED at drivers/gpu/drm/amd/amdgpu/../display/tests/dc/dml/dcn20/dcn20_fpu_test.c:500
[19:02:25] Expected test_param->expected_clock_limits[i].dispclk_mhz == dcn2_1_soc.clock_limits[i].dispclk_mhz, but
[19:02:25] test_param->expected_clock_limits[i].dispclk_mhz == 757
[19:02:25] dcn2_1_soc.clock_limits[i].dispclk_mhz == 600
[19:02:25] # dcn21_update_bw_bounding_box_test: EXPECTATION FAILED at drivers/gpu/drm/amd/amdgpu/../display/tests/dc/dml/dcn20/dcn20_fpu_test.c:502
[19:02:25] Expected test_param->exp
[19:02:25] test_param->expected_clo
[19:02:25] dcn2_1_soc.clock_limits[i
[19:02:25] # dcn21_update_bw_boundi
[19:02:25] Expected test_param->exp
[19:02:25] test_param->expected_clo
[19:02:25] dcn2_1_soc.clock_limits[i].dscclk_mhz == 205
[19:02:25] # dcn21_update_bw_bounding_box_test: EXPECTATION FAILED at drivers/gpu/drm/amd/amdgpu/../display/tests/dc/dml/dcn20/dcn20_fpu_test.c:500
[19:02:25] Expected test_param->expected_clock_limits[i].dispclk_mhz == dcn2_1_soc.clock_limits[i].dispclk_mhz, but
[19:02:25] test_param->expected_clock_limits[i].dispclk_mhz == 847
[19:02:25] dcn2_1_soc.clock_limits[i].dispclk_mhz == 600
[19:02:25] # dcn21_update_bw_bounding_box_test: EXPECTATION FAILED at drivers/gpu/drm/amd/amdgpu/../display/tests/dc/dml/dcn20/dcn20_fpu_test.c:502
[19:02:25] Expected test_param->expected_clock_limits[i].dppclk_mhz == dcn2_1_soc.clock_limits[i].dppclk_mhz, but
[19:02:25] test_param->expected_clock_limits[i].dsccclk_mhz == 318
[19:02:25] dcn2_1_soc.clock_limits[i].dsccclk_mhz == 205
[19:02:25] not ok 1 - 5-entry bounding box clocks table
[19:02:25] [FAILED] 5-entry bounding box clocks table
[19:02:25] # Subtest: dcn21_update_bw_bounding_box_test
[19:02:25] not ok 1 - dcn21_update_bw_bounding_box_test
[19:02:25] ========== [FAILED] dcn21_update_bw_bounding_box_test ==========
[19:02:25] # Subtest: dml_dcn20_fpu_dcn21_update_bw_bounding_box_test
[19:02:25] 1..1
[19:02:25] not ok 8 - dml_dcn20_fpu_dcn21_update_bw_bounding_box_test
[19:02:25] = [FAILED] dml_dcn20_fpu_dcn21_update_bw_bounding_box_test =
[19:02:25] ==========
[19:02:25] Testing complete. Passed: 87, Failed: 6, Crashed: 0, Skipped: 0, Errors: 0
```

```
[19:02:25] test_param->expected_clock_limits[i].dispclk_mhz == 847
[19:02:25] dcn2_1_soc.clock_limits[i].dispclk_mhz == 600
```

```
[19:02:25] Testing complete. Passed: 87, Failed: 6, Crashed: 0, Skipped: 0, Errors: 0
```

[PATCH v2 6/8] drm/amd/display: Introduce KUnit tests for dcn20_fpu
https://lore.kernel.org/dri-devel/20220831172239.344446-7-mairacanal@riseup.net/

03 Finally, we fix the code and check if the test case we wrote is now passing.

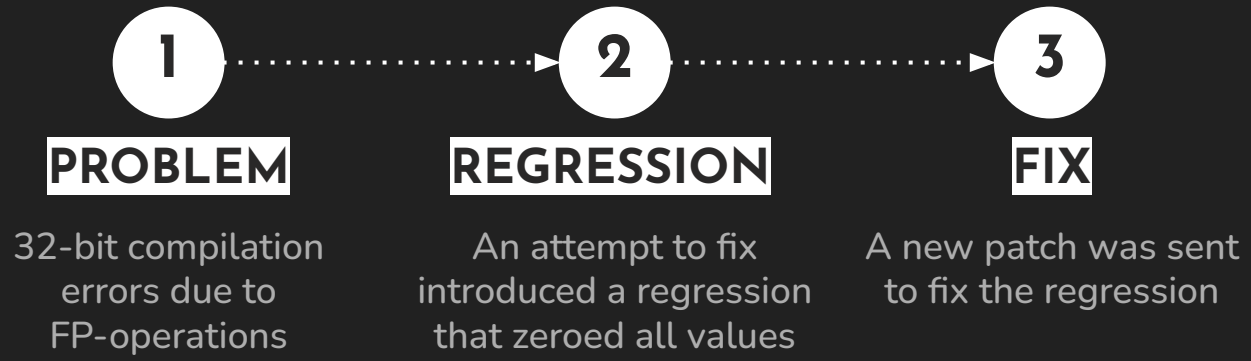
```
$ ./tools/testing/kunit/kunit.py run --arch=x86_64 --kunitconfig=drivers/gpu/drm/amd/display/tests
'dml_dcn20_fpu_dcn21_update_bw_bounding_box_test'
[18:57:57] Configuring KUnit Kernel…
[18:57:57] Building KUnit Kernel…
Populating config with:
$ make ARCH=x86_64 olddefconfig O=.kunit
Building with:
$ make ARCH=x86_64 --jobs=8 O=.kunit
[18:58:17] Starting KUnit Kernel (1/1)...
[18:58:17] ============================================================
Running tests with:
$ qemu-system-x86_64 -nodefaults -m 1024 -kernel .kunit/arch/x86/boot/bzImage -append 'mem=1G console=tty kunit_shutdown=halt
kunit.filter_glob=dml_dcn20_fpu console=ttyS0 kunit_shutdown=reboot' -no-reboot -nographic -serial stdio
[19:17:06] ==== dml_dcn20_fpu_dcn21_update_bw_bounding_box_test (1 subtest) ====
[19:17:06] ============== dcn21_update_bw_bounding_box_test ==============
[19:17:06] [PASSED] 5-entry bounding box clocks table
[19:17:06] ========== [PASSED] dcn21_update_bw_bounding_box_test ==========
[19:17:06] = [PASSED] dml_dcn20_fpu_dcn21_update_bw_bounding_box_test =
[19:17:06] ============================================================
[19:17:06] Testing complete. Passed: 1, Failed: 0, Crashed: 0, Skipped: 0, Errors: 0
[19:17:07] Elapsed time: 9.283s total, 0.003s configuring, 7.811s building, 0.829s running
```

[PATCH] drm/amd/display: Use pre-allocated temp struct for bounding box update
https://lore.kernel.org/amd-gfx/20220608164856.1870594-1-sunpeng.li@amd.com/

Case of study 3 - Catching regressions with KUnit

dc_dmub_srv

The **`populate_subvp_cmd_drr_info()`** case:



How to include unit tests in such development workflow?

01 Initially, we have a problem

[PATCH 02/40] drm/amd/display: Add SubVP required code

https://lore.kernel.org/amd-gfx/20220630191322.9096650-3-Rodrigo.Siqueira@amd.com/

```
static void populate_subvp_cmd_drr_info(struct dc *dc,
                struct pipe_ctx *subvp_pipe,
                struct pipe_ctx *vblank_pipe,
                struct dmub_cmd_fw_assisted_mclk_switch_pipe_data_v2 *pipe_data)
{
        [ ... ]

        pipe_data->pipe_config.vblank_data.drr_info.drr_in_use = true;
        pipe_data->pipe_config.vblank_data.drr_info.use_ramping = false;
        pipe_data->pipe_config.vblank_data.drr_info.drr_window_size_ms = 4;

        drr_frame_us = drr_timing->v_total * drr_timing->h_total /
                        (double)(drr_timing->pix_clk_100hz * 100) * 1000000;

        mall_region_us = phantom_timing->v_addressable * phantom_timing->h_total /
                        (double)(phantom_timing->pix_clk_100hz * 100) * 1000000;
        min_drr_supported_us = drr_frame_us + mall_region_us + SUBVP_DRR_MARGIN_US;
        min_vtotal_supported = drr_timing->pix_clk_100hz * 100 * ((double)min_drr_supported_us / 1000000) /
                        (double)drr_timing->h_total;

        prefetch_us = (phantom_timing->v_total - phantom_timing->v_front_porch) * phantom_timing->h_total /
                        (double)(phantom_timing->pix_clk_100hz * 100) * 1000000 +
                        dc->caps.subvp_prefetch_end_to_mall_start_us;
        subvp_active_us = main_timing->v_addressable * main_timing->h_total /
                        (double)(main_timing->pix_clk_100hz * 100) * 1000000;
        drr_active_us = drr_timing->v_addressable * drr_timing->h_total /
                        (double)(drr_timing->pix_clk_100hz * 100) * 1000000;
        max_drr_vblank_us = (double)(subvp_active_us - prefetch_us - drr_active_us) / 2 + drr_active_us;
        max_drr_mallregion_us = subvp_active_us - prefetch_us - mall_region_us;
        max_drr_supported_us = max_drr_vblank_us > max_drr_mallregion_us ? max_drr_vblank_us : max_drr_mallregion_us;
        max_vtotal_supported = drr_timing->pix_clk_100hz * 100 * ((double)max_drr_supported_us / 1000000) /
                        (double)drr_timing->h_total;

        pipe_data->pipe_config.vblank_data.drr_info.min_vtotal_supported = min_vtotal_supported;
        pipe_data->pipe_config.vblank_data.drr_info.max_vtotal_supported = max_vtotal_supported;
}
```

02 A fix to this problem is proposed

```
static void populate_subvp_cmd_drr_info(struct dc *dc,
                struct pipe_ctx *subvp_pipe,
                struct pipe_ctx *vblank_pipe,
                struct dmub_cmd_fw_assisted_mclk_switch_pipe_data_v2 *pipe_data)
{
        [ ... ]

        pipe_data->pipe_config.vblank_data.drr_info.drr_in_use = true;
        pipe_data->pipe_config.vblank_data.drr_info.use_ramping = false;
        pipe_data->pipe_config.vblank_data.drr_info.drr_window_size_ms = 4;

        drr_frame_us = div64_s64(drr_timing->v_total * drr_timing->h_total,
                                 (int64_t)(drr_timing->pix_clk_100hz * 100) * 1000000);
        mall_region_us = div64_s64(phantom_timing->v_addressable * phantom_timing->h_total,
                                   (int64_t)(phantom_timing->pix_clk_100hz * 100) * 1000000);
        min_drr_supported_us = drr_frame_us + mall_region_us + SUBVP_DRR_MARGIN_US;
        min_vtotal_supported = div64_s64(drr_timing->pix_clk_100hz * 100 *
                                         (div64_s64((int64_t)min_drr_supported_us, 1000000)),
                                         (int64_t)drr_timing->h_total);

        prefetch_us = div64_s64((phantom_timing->v_total - phantom_timing->v_front_porch) * phantom_timing->h_total,
                                (int64_t)(phantom_timing->pix_clk_100hz * 100) * 1000000 +
                                dc->caps.subvp_prefetch_end_to_mall_start_us);
        subvp_active_us = div64_s64(main_timing->v_addressable * main_timing->h_total,
                                    (int64_t)(main_timing->pix_clk_100hz * 100) * 1000000);
        drr_active_us = div64_s64(drr_timing->v_addressable * drr_timing->h_total,
                                  (int64_t)(drr_timing->pix_clk_100hz * 100) * 1000000);
        max_drr_vblank_us = div64_s64((int64_t)(subvp_active_us - prefetch_us - drr_active_us), 2) + drr_active_us;
        max_drr_mallregion_us = subvp_active_us - prefetch_us - mall_region_us;
        max_drr_supported_us = max_drr_vblank_us > max_drr_mallregion_us ? max_drr_vblank_us : max_drr_mallregion_us;
        max_vtotal_supported = div64_s64(drr_timing->pix_clk_100hz * 100 * (div64_s64((int64_t)max_drr_supported_us, 1000000)),
                                         (int64_t)drr_timing->h_total);

        pipe_data->pipe_config.vblank_data.drr_info.min_vtotal_supported = min_vtotal_supported;
        pipe_data->pipe_config.vblank_data.drr_info.max_vtotal_supported = max_vtotal_supported;

}
```

[PATCH] drm/amd/display: fix 32 bit compilation errors in dc_dmub_srv.c

https://lore.kernel.org/amd-gfx/20220708052650.1029150-1-alexander.deucher@amd.com/

03 ...but, the fix introduces a regression

```
static void populate_subvp_cmd_drr_info(struct dc *dc,
                struct pipe_ctx *subvp_pipe,
                struct pipe_ctx *vblank_pipe,
                struct dmub_cmd_fw_assisted_mclk_switch_pipe_data_v2 *pipe_data)
{
        [ ... ]

        pipe_data->pipe_config.vblank_data.drr_info.drr_in_use = true;
        pipe_data->pipe_config.vblank_data.drr_info.use_ramping = false;
        pipe_data->pipe_config.vblank_data.drr_info.drr_window_size_ms = 4;

        drr_frame_us = div64_s64(drr_timing->v_total * drr_timing->h_total,
                                 (int64_t)(drr_timing->pix_clk_100hz * 100) * 1000000);
        mall_region_us = div64_s64(phantom_timing->v_addressable * phantom_timing->h_total,
                                   (int64_t)(phantom_timing->pix_clk_100hz * 100) * 1000000);
        min_drr_supported_us = drr_frame_us + mall_region_us + SUBVP_DRR_MARGIN_US;
        min_vtotal_supported = div64_s64(drr_timing->pix_clk_100hz * 100 *
                                         (div64_s64((int64_t)min_drr_supported_us, 1000000)),
                                         (int64_t)drr_timing->h_total);

        prefetch_us = div64_s64((phantom_timing->v_total - phantom_timing->v_front_porch) * phantom_timing->h_total,
                                (int64_t)(phantom_timing->pix_clk_100hz * 100) * 1000000 +
                                dc->caps.subvp_prefetch_end_to_mall_start_us);
        subvp_active_us = div64_s64(main_timing->v_addressable * main_timing->h_total,
                                    (int64_t)(main_timing->pix_clk_100hz * 100) * 1000000);
        drr_active_us = div64_s64(drr_timing->v_addressable * drr_timing->h_total,
                                  (int64_t)(drr_timing->pix_clk_100hz * 100) * 1000000);
        max_drr_vblank_us = div64_s64((int64_t)(subvp_active_us - prefetch_us - drr_active_us), 2) + drr_active_us;
        max_drr_mallregion_us = subvp_active_us - prefetch_us - mall_region_us;
        max_drr_supported_us = max_drr_vblank_us > max_drr_mallregion_us ? max_drr_vblank_us : max_drr_mallregion_us;
        max_vtotal_supported = div64_s64(drr_timing->pix_clk_100hz * 100 * (div64_s64((int64_t)max_drr_supported_us, 1000000)),
                                         (int64_t)drr_timing->h_total);

        pipe_data->pipe_config.vblank_data.drr_info.min_vtotal_supported = min_vtotal_supported; == 0
        pipe_data->pipe_config.vblank_data.drr_info.max_vtotal_supported = max_vtotal_supported; == 0

}
```

[PATCH] drm/amd/display: fix 32 bit compilation errors in dc_dmub_srv.c

https://lore.kernel.org/amd-gfx/20220708052650.1029150-1-alexander.deucher@amd.com/

04 Then we write a test case based on the behavior pre-regression

```
$ ./tools/testing/kunit/kunit.py run --arch=x86_64 --kunitconfig=drivers/gpu/drm/amd/display/tests 'dc_dmub_srv'
[13:21:45] ================== dc_dmub_srv (1 subtest) ==================
[13:21:45] ============== populate_subvp_cmd_drr_info_test ==============
[13:21:45] # populate_subvp_cmd_drr_info_test: EXPECTATION FAILED at drivers/gpu/drm/amd/amdgpu/../display/dc/../tests/dc/dc_dmub_srv_test.c:269
[13:21:45] Expected test_param->min_vtotal_supported == pipe_data->pipe_config.vblank_data.drr_info.min_vtotal_supported, but
[13:21:45] test_param->min_vtotal_supported == 63709
[13:21:45] pipe_data->pipe_config.vblank_data.drr_info.min_vtotal_supported == 0
[13:21:45] # populate_subvp_cmd_drr_info_test: EXPECTATION FAILED at drivers/gpu/drm/amd/amdgpu/../display/dc/../tests/dc/dc_dmub_srv_test.c:271
[13:21:45] Expected test_param->max_vtotal_supported == pipe_data->pipe_config.vblank_data.drr_info.max_vtotal_supported, but
[13:21:45] test_param->max_vtotal_supported == 363
[13:21:45] pipe_data->pipe_config.vblank_data.drr_info.max_vtotal_supported == 0
[13:21:45] not ok 1 - Same Clock Frequency
[13:21:45] [FAILED] Same Clock Frequency
[13:21:45] # populate_subvp_cmd_drr_info_test: EXPECTATION FAILED at drivers/gpu/drm/amd/amdgpu/../display/dc/../tests/dc/dc_dmub_srv_test.c:269
[13:21:45] Expected test_param->min_vtotal_supported == pipe_data->pipe_config.vblank_data.drr_info.min_vtotal_supported, but
[13:21:45] test_param->min_vtotal_supported == 63709
[13:21:45] pipe_data->pipe_config.vblank_data.drr_info.min_vtotal_supported == 0
[13:21:45] # populate_subvp_cmd_drr_info_test: EXPECTATION FAILED at drivers/gpu/drm/amd/amdgpu/../display/dc/../tests/dc/dc_dmub_srv_test.c:271
[13:21:45] Expected test_param->max_vtotal_supported == pipe_data->pipe_config.vblank_data.drr_info.max_vtotal_supported, but
[13:21:45] test_param->max_vtotal_supported == 346
[13:21:45] pipe_data->pipe_config.vblank_data.drr_info.max_vtotal_supported == 0
[13:21:45] not ok 2 - Same Clock Frequency with Prefetch End to Mall Start
[13:21:45] [FAILED] Same Clock Frequency with Prefetch End to Mall Start
[ ... ]
[13:21:45] # populate_subvp_cmd_drr_info_test: pass:0 fail:5 skip:0 total:5
[13:21:45] not ok 1 - populate_subvp_cmd_drr_info_test
[13:21:45] ======== [FAILED] populate_subvp_cmd_drr_info_test ========
[13:21:45] # Subtest: dc_dmub_srv [13:21:45] 1..1
[13:21:45] # Totals: pass:0 fail:5 skip:0 total:5
[13:21:45] not ok 1 - dc_dmub_srv
[13:21:45] ================== [FAILED] dc_dmub_srv ==================
[13:21:45] ============================================================
[13:21:45] Testing complete. Passed: 0, Failed: 5, Crashed: 0, Skipped: 0, Errors: 0
[13:21:46] Elapsed time: 6.051s total, 0.003s configuring, 4.579s building, 0.843s running
```

[PATCH v2 7/8] drm/amd/display: Introduce KUnit tests to dc_dmub_srv library

https://lore.kernel.org/dri-devel/20220831172239.34446-8-mairacanal@riseup.net/

05 Finally, we revert the regression and check if the test case is now passing.

```
$ ./tools/testing/kunit/kunit.py run --arch=x86_64 --kunitconfig=drivers/gpu/drm/amd/display/tests 'dc_dmub_srv'
[13:36:32] Configuring KUnit Kernel …
[13:36:32] Building KUnit Kernel ...
Populating config with:
$ make ARCH=x86_64 olddefconfig O=.kunit
Building with:
$ make ARCH=x86_64 --jobs=8 O=.kunit
[13:36:40] Starting KUnit Kernel (1/1)...
[13:36:40] ============================================================
Running tests with:
$ qemu-system-x86_64 -nodefaults -m 1024 -kernel .kunit/arch/x86/boot/bzImage -append 'mem=1G console=tty
kunit_shutdown=halt kunit.filter_glob=dc_dmub_srv console=ttyS0 kunit_shutdown=reboot' -no-reboot -nographic -serial
stdio
[13:36:41] ================ dc_dmub_srv (1 subtest) ================
[13:36:41] ============ populate_subvp_cmd_drr_info_test ============
[13:36:41] [PASSED] Same Clock Frequency
[13:36:41] [PASSED] Same Clock Frequency with Prefetch End to Mall Start
[13:36:41] [PASSED] Same Clock Frequency Not Multiple of 2
[13:36:41] [PASSED] Different Clock Frequency for smaller h_total and v_total
[13:36:41] [PASSED] Different Clock Frequency for approximately 1920×1080
[13:36:41] ======== [PASSED] populate_subvp_cmd_drr_info_test ========
[13:36:41] ================== [PASSED] dc_dmub_srv ==================
[13:36:41] ============================================================
[13:36:41] Testing complete. Passed: 5, Failed: 0, Crashed: 0, Skipped: 0, Errors: 0
[13:36:41] Elapsed time: 9.831s total, 0.002s configuring, 8.260s building, 0.917s running
```

How to run KUnit?

How easy is to run the unit tests?

Running KUnit

01 On Userspace

Currently, not supported on AMDGPU

```
$ ./tools/testing/kunit/kunit.py run --kunitconfig=drivers/gpu/drm/amd/display/tests
```

02 On QEMU

Easy way to run tests during development

```
$ ./tools/testing/kunit/kunit.py run --arch=x86_64 --kunitconfig=drivers/gpu/drm/amd/display/tests
```

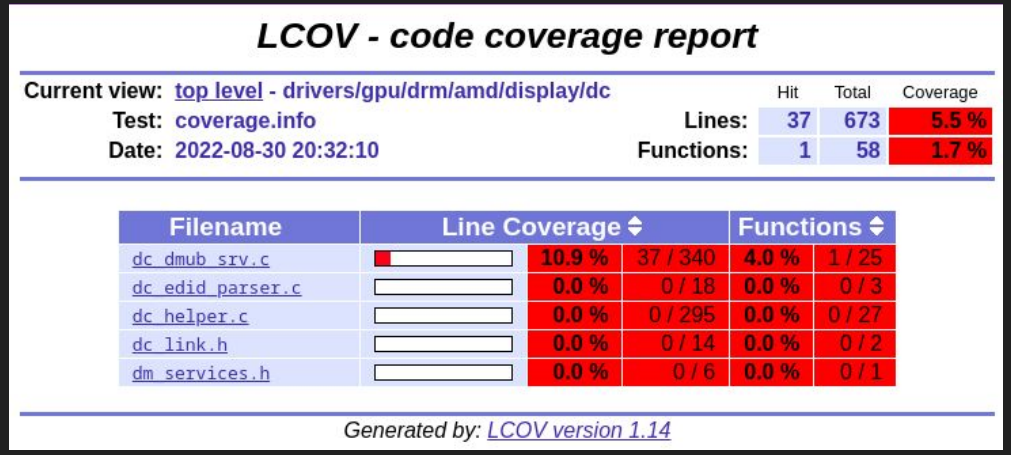
03 On Hardware

Runs on boot if **amdgpu** is built-in or can be loaded as a module

```
$ modprobe amdgpu
```

KUnit + gcov: Test Coverage

- Shows how much source code is run when the tests are executed
- Helps identify areas with missing coverage



LCOV - code coverage report

Current view:	top level - drivers/gpu/drm/amd/display/dc		Hit	Total	Coverage
Test:	coverage.info	Lines:	37	673	5.5 %
Date:	2022-08-30 20:32:10	Functions:	1	58	1.7 %

Filename	Line Coverage			Functions	
dc_dmub_srv.c		10.9 %	37 / 340	4.0 %	1 / 25
dc_edid_parser.c		0.0 %	0 / 18	0.0 %	0 / 3
dc_helper.c		0.0 %	0 / 295	0.0 %	0 / 27
dc_link.h		0.0 %	0 / 14	0.0 %	0 / 2
dm_services.h		0.0 %	0 / 6	0.0 %	0 / 1

Generated by: LCOV version 1.14

Directory	Line Coverage			Functions	
drivers/gpu/drm/amd/display/dc/dml		5.3 %	83 / 1567	4.0 %	6 / 149
drivers/gpu/drm/amd/display/dc/dml/calcs		1.7 %	74 / 4278	15.9 %	7 / 44
drivers/gpu/drm/amd/display/dc/dml/dcn20		2.9 %	219 / 7554	9.3 %	7 / 75

What we know

- It's reasonable to test static functions.
- It's easy to run unit tests without the specific hardware.
- Device mocking is not *really* necessary.

What we don't know

- How to test static functions properly?
- Run inside **IGT** or **standalone**?
  - (+) Use case for a couple of companies
  - (+) Easier to integrate into CI
  - (−) One more piece of code to maintain (IGT ktap parser)
- Currently the tests are inside of the AMDGPU module:

  Should we keep it this way or switch to standalone modules?

Questions?