author	Charles Keepax <ckeepax@opensource.cirrus.com>	2025-02-28 15:14:56 +0000
committer	Mark Brown <broonie@kernel.org>	2025-03-05 17:25:25 +0000
commit	0eba2a7e858907a746ba69cd002eb9eb4dbd7bf3 (patch)	
tree	90a790819c504ca80712e19f52f89eeffa172619	
parent	927e6bec5cf3624665b0a2e9f64a1d32f3d22cdd (diff)	
download	linux-0eba2a7e858907a746ba69cd002eb9eb4dbd7bf3.tar.gz	

ASoC: ops: Consistently treat platform_max as control value

```
This reverts commit 9bdd10d57a88 ("ASoC: ops: Shift tested values in
snd_soc_put_volsw() by +min"), and makes some additional related
updates.

There are two ways the platform_max could be interpreted; the maximum
register value, or the maximum value the control can be set to. The
patch moved from treating the value as a control value to a register
one. When the patch was applied it was technically correct as
snd_soc_limit_volume() also used the register interpretation. However,
even then most of the other usages treated platform_max as a
control value, and snd_soc_limit_volume() has since been updated to
also do so in commit fb9ad24485087 ("ASoC: ops: add correct range
check for limiting volume"). That patch however, missed updating
snd_soc_put_volsw() back to the control interpretation, and fixing
snd_soc_info_volsw_range(). The control interpretation makes more
sense as limiting is typically done from the machine driver, so it is
appropriate to use the customer facing representation rather than the
internal codec representation. Update all the code to consistently use
this interpretation of platform_max.

Finally, also add some comments to the soc_mixer_control struct to
hopefully avoid further patches switching between the two approaches.

Fixes: fb9ad24485087 ("ASoC: ops: add correct range check for limiting volume")
Signed-off-by: Charles Keepax <ckeepax@opensource.cirrus.com>
Link: https://patch.msgid.link/20250228151456.3703342-1-ckeepax@opensource.cirrus.com
Signed-off-by: Mark Brown <broonie@kernel.org>
```

Diffstat

-rw-r--r-- include/sound/soc.h	5
-rw-r--r-- sound/soc/soc-ops.c	15

2 files changed, 11 insertions, 9 deletions

```
diff --git a/include/sound/soc.h b/include/sound/soc.h
index fcdb5adfcd5ecc..b3e84bc47c6fdd 100644
--- a/include/sound/soc.h
+++ b/include/sound/soc.h
@@ -1261,7 +1261,10 @@ void snd_soc_close_delayed_work(struct snd_soc_pcm_runtime *rtd);

 /* mixer control */
 struct soc_mixer_control {
```

```
-       int min, max, platform_max;
+       /* Minimum and maximum specified as written to the hardware */
+       int min, max;
+       /* Limited maximum value specified as presented through the control */
+       int platform_max;
        int reg, rreg;
        unsigned int shift, rshift;
        unsigned int sign_bit;
```

```
diff --git a/sound/soc/soc-ops.c b/sound/soc/soc-ops.c
index 19928f098d8dcb..b0e4e4168f38d5 100644
--- a/sound/soc/soc-ops.c
+++ b/sound/soc/soc-ops.c
@@ -337,7 +337,7 @@ int snd_soc_put_volsw(struct snd_kcontrol *kcontrol,
        if (ucontrol->value.integer.value[0] < 0)
                return -EINVAL;
        val = ucontrol->value.integer.value[0];
-       if (mc->platform_max && ((int)val + min) > mc->platform_max)
+       if (mc->platform_max && val > mc->platform_max)
                return -EINVAL;
        if (val > max - min)
                return -EINVAL;
@@ -350,7 +350,7 @@ int snd_soc_put_volsw(struct snd_kcontrol *kcontrol,
                if (ucontrol->value.integer.value[1] < 0)
                        return -EINVAL;
                val2 = ucontrol->value.integer.value[1];
-               if (mc->platform_max && ((int)val2 + min) > mc->platform_max)
+               if (mc->platform_max && val2 > mc->platform_max)
                        return -EINVAL;
                if (val2 > max - min)
                        return -EINVAL;
@@ -503,17 +503,16 @@ int snd_soc_info_volsw_range(struct snd_kcontrol *kcontrol,
 {
        struct soc_mixer_control *mc =
                (struct soc_mixer_control *)kcontrol->private_value;
-       int platform_max;
-       int min = mc->min;
+       int max;

-       if (!mc->platform_max)
-               mc->platform_max = mc->max;
-       platform_max = mc->platform_max;
+       max = mc->max - mc->min;
+       if (mc->platform_max && mc->platform_max < max)
+               max = mc->platform_max;

        uinfo->type = SNDRV_CTL_ELEM_TYPE_INTEGER;
        uinfo->count = snd_soc_volsw_is_stereo(mc) ? 2 : 1;
        uinfo->value.integer.min = 0;
-       uinfo->value.integer.max = platform_max - min;
+       uinfo->value.integer.max = max;

        return 0;
 }
```