Project Deliverables

ÖPENSÖURCE AB 201506

June 19th 2015

1. TMU/CMT development and upstreaming (part 1)
2. Arch timer development and upstreaming (part 1)

1. TMU/CMT development and upstreaming (part 1)

1.1 Task description:

Develop, upstream feature support and integrate the TMU and CMT drivers on new SoCs and boards. The code shall be developed in an incremental fashion to fit latest upstream requirements. The main goal with this task is to prepare for R-Car Gen 3 support which may require remote access to real Gen 3 hardware or use of FPGA based target platforms, however SoCs and boards related to the R-Car Gen 2 family may also be covered on a best effort basis, including r8a7790 Lager, r8a7791 Koelsch, r8a7794 Alt and r8a7793 Gose. DT shall be used whenever possible together with Multiplatform support. Care should be taken to ease back porting to LTSI kernels. Patches shall be posted to community mailing lists for upstream merge.

1.2 Task deliverable #1:

1.2.1 Deliverable name:

description

1.2.2 Deliverable contents:

clocksource: sh_cmt: DT binding rework

[PATCH 01/08] devicetree: bindings: Remove sh7372 CMT binding
[PATCH 02/08] clocksource: sh_cmt: Use 0x3f mask for SH_CMT_48BIT case
[PATCH 03/08] devicetree: bindings: R-Car Gen2 CMT0 and CMT1 bindings
[PATCH 04/08] clocksource: sh_cmt: Support separate R-Car Gen2 CMT0/1
[PATCH 05/08] devicetree: bindings: r8a73a4 and R-Car Gen2 CMT bindings
[PATCH 06/08] ARM: shmobile: Update CMT compat string users in DTS
[PATCH 07/08] devicetree: bindings: Deprecate property, update example
[PATCH 08/08] ARM: shmobile: Remove CMT renesas,channels-mask from DTS

This series reworks the CMT DT bindings to try to deal with the following:
- R-Car Gen2 CMT0 and CMT1 hardware instances are not identical
- The property renesas,channels-mask is not enough to describe the difference
- DT should describe the hardware, not the software implementation
- Not all documented DT bindings are actually used

Without these patches the binding "renesas,cmt-48-gen2" is used for both CMT0 and CMT1 on R-Car Gen2 SoCs. CMT0 and CMT1 are currently seen by the Linux device driver as compatible hardware, and the device-specific property "renesas,channels-mask" is used to point out some of the hardware configuration differences. Since the driver is not feature complete only some differences are described and when diving into the data sheet we can see that:

1) CMT0 is not 48-bit at all, instead it only supports 32-bit counters.
2) Some channels of CMT1 are 48-bit, some 32-bit.
3) A couple of CMT1 channels have even more features.

It turns out that none of the above differences are described in our current DT files. And since we use the same compat string for CMT0 and CMT1 the driver

itself cannot enable features specific only to CMT1 without first updating
the DTS. So this series is ground work for future feature patches.

It seems that we have two choices if we want to support CMT1 features:
A) Keep existing DT bindings, add more properties for CMT1
B) Rework the compatible strings and keep configuration in the driver

Judging by above it seems that DT update is inevitable. In my mind it is
rather painful to update the DT so I'd like to minimize the number of
updates and let the majority of the changes only happen in the driver.
And since we should really describe hardware in DT but driver features tend
to be implemented incrementally then B) seems like a good fit to me.

I wouldn't mind going with A) but to be honest I must say that the existing
compat string "renesas,cmt-48-gen2" is just too confusing with "48"
(when CMT0 is 32bit-only) and also "gen2" (used without product line R-Car).

Because of that I've gone with B) and reworked the bindings to separate
CMT0 from CMT1 and keep channel configuration in the device driver.

While at it I've deprecated "renesas,channels-mask" and the old DT compat
strings. Per-SoC compat strings have also been reworked, but are not left
around documented as deprecated since they were never used by the driver.

Comments are very welcome! If all are in favor then it would be good to
merge patch 1-5 first and patch 6-8 later once 1-5 are in. Old DTBs are
left working but treated as low-feature CMT0. New DTBs including patch 6-8
will not work on old kernels.

Signed-off-by: Magnus Damm <damm+renesas@opensource.se>

```
 Documentation/devicetree/bindings/timer/renesas,cmt.txt |   55 +++++++++------
 arch/arm/boot/dts/r8a73a4.dtsi                          |    5 -
 arch/arm/boot/dts/r8a7740.dtsi                          |    3
 arch/arm/boot/dts/r8a7790.dtsi                          |   10 --
 arch/arm/boot/dts/r8a7791.dtsi                          |   10 --
 arch/arm/boot/dts/r8a7793.dtsi                          |   10 --
 arch/arm/boot/dts/r8a7794.dtsi                          |   10 --
 arch/arm/boot/dts/sh73a0.dtsi                           |    3
 drivers/clocksource/sh_cmt.c                            |   54 ++++++++++----
 9 files changed, 83 insertions(+), 77 deletions(-)
```

1.3 Task deliverable #2:

1.3.1 Deliverable name:

linux-clocksource-sh_cmt-support-separate-rcar-gen2-cmt01.patch

1.3.2 Deliverable contents:

From: Magnus Damm <damm+renesas@opensource.se>

Add support for the new R-Car Gen2 CMT0 and CMT1 bindings. Support
for the old DT binding is still kept around, however devices using
such binding will be treated as a low-feature CMT0 device. If users
want to make use of CMT1-specific features then they need to update
their DTBs. No special CMT1-specific features are however implemented
by his patch, only DT bindings are redone as groundwork for future
feature patches.

Signed-off-by: Magnus Damm <damm+renesas@opensource.se>

```
---

 drivers/clocksource/sh_cmt.c |   40 ++++++++++++++++++++++++++++------------
 1 file changed, 28 insertions(+), 12 deletions(-)

--- 0005/drivers/clocksource/sh_cmt.c
+++ work/drivers/clocksource/sh_cmt.c
@@ -39,16 +39,16 @@ struct sh_cmt_device;
  * SoC but also on the particular instance. The following table lists the main
  * characteristics of those flavours.
  *
- *                   16B     32B     32B-F   48B     48B-2
+ *                   16B     32B     32B-F   48B     R-Car Gen2
  * -----------------------------------------------------------------------------
  * Channels          2       1/4     1       6       2/8
  * Control Width     16      16      16      16      32
  * Counter Width     16      32      32      32/48   32/48
  * Shared Start/Stop Y       Y       Y       Y       N
  *
- * The 48-bit gen2 version has a per-channel start/stop register located in the
- * channel registers block. All other versions have a shared start/stop register
- * located in the global space.
+ * The r8a73a4 / R-Car Gen2 version has a per-channel start/stop register
+ * located in the channel registers block. All other versions have a shared
+ * start/stop register located in the global space.
  *
  * Channels are indexed from 0 to N-1 in the documentation. The channel index
  * infers the start/stop bit position in the control register and the channel
@@ -68,7 +68,8 @@ enum sh_cmt_model {
 	SH_CMT_32BIT,
 	SH_CMT_32BIT_FAST,
 	SH_CMT_48BIT,
-	SH_CMT_48BIT_GEN2,
+	SH_CMT0_RCAR_GEN2,
+	SH_CMT1_RCAR_GEN2,
 };

 struct sh_cmt_info {
@@ -223,8 +224,20 @@ static const struct sh_cmt_info sh_cmt_i
 		.read_count = sh_cmt_read32,
 		.write_count = sh_cmt_write32,
 	},
-	[SH_CMT_48BIT_GEN2] = {
-		.model = SH_CMT_48BIT_GEN2,
+	[SH_CMT0_RCAR_GEN2] = {
+		.model = SH_CMT0_RCAR_GEN2,
+		.channels_mask = 0x60,
+		.width = 32,
+		.overflow_bit = SH_CMT32_CMCSR_CMF,
+		.clear_bits = ~(SH_CMT32_CMCSR_CMF | SH_CMT32_CMCSR_OVF),
+		.read_control = sh_cmt_read32,
+		.write_control = sh_cmt_write32,
+		.read_count = sh_cmt_read32,
+		.write_count = sh_cmt_write32,
+	},
+	[SH_CMT1_RCAR_GEN2] = {
+		.model = SH_CMT1_RCAR_GEN2,
+		.channels_mask = 0xff,
 		.width = 32,
 		.overflow_bit = SH_CMT32_CMCSR_CMF,
 		.clear_bits = ~(SH_CMT32_CMCSR_CMF | SH_CMT32_CMCSR_OVF),
@@ -861,6 +874,7 @@ static int sh_cmt_setup_channel(struct s
 	ch->cmt = cmt;
 	ch->index = index;
 	ch->hwidx = hwidx;
+	ch->timer_bit = hwidx;

 	/*
 	 * Compute the address of the channel control register block. For the
@@ -882,9 +896,11 @@ static int sh_cmt_setup_channel(struct s
 		 */
 		ch->ioctrl = cmt->mapbase + 0x40;
 		break;
-	case SH_CMT_48BIT_GEN2:
+	case SH_CMT0_RCAR_GEN2:
+	case SH_CMT1_RCAR_GEN2:
 		ch->iostart = cmt->mapbase + ch->hwidx * 0x100;
 		ch->ioctrl = ch->iostart + 0x10;
+		ch->timer_bit = 0;
 		break;
 	}

@@ -896,8 +912,6 @@ static int sh_cmt_setup_channel(struct s
 	ch->match_value = ch->max_match_value;
 	raw_spin_lock_init(&ch->lock);

-	ch->timer_bit = cmt->info->model == SH_CMT_48BIT_GEN2 ? 0 : ch->hwidx;
-
 	ret = sh_cmt_register(ch, dev_name(&cmt->pdev->dev),
 			      clockevent, clocksource);
 	if (ret) {
@@ -934,7 +948,7 @@ static const struct platform_device_id s
 	{ "sh-cmt-32", (kernel_ulong_t)&sh_cmt_info[SH_CMT_32BIT] },
 	{ "sh-cmt-32-fast", (kernel_ulong_t)&sh_cmt_info[SH_CMT_32BIT_FAST] },
	{ "sh-cmt-48", (kernel_ulong_t)&sh_cmt_info[SH_CMT_48BIT] },
-	{ "sh-cmt-48-gen2", (kernel_ulong_t)&sh_cmt_info[SH_CMT_48BIT_GEN2] },
+	{ "sh-cmt-48-gen2", (kernel_ulong_t)&sh_cmt_info[SH_CMT0_RCAR_GEN2] },
	{ }
 };
 MODULE_DEVICE_TABLE(platform, sh_cmt_id_table);
@@ -943,7 +957,9 @@ static const struct of_device_id sh_cmt_
	{ .compatible = "renesas,cmt-32", .data = &sh_cmt_info[SH_CMT_32BIT] },
	{ .compatible = "renesas,cmt-32-fast", .data = &sh_cmt_info[SH_CMT_32BIT_FAST] },
	{ .compatible = "renesas,cmt-48", .data = &sh_cmt_info[SH_CMT_48BIT] },
-	{ .compatible = "renesas,cmt-48-gen2", .data = &sh_cmt_info[SH_CMT_48BIT_GEN2] },
+	{ .compatible = "renesas,cmt-48-gen2", .data = &sh_cmt_info[SH_CMT0_RCAR_GEN2] },
+	{ .compatible = "renesas,cmt0-rcar-gen2", .data =
&sh_cmt_info[SH_CMT0_RCAR_GEN2] },
+	{ .compatible = "renesas,cmt1-rcar-gen2", .data =
&sh_cmt_info[SH_CMT1_RCAR_GEN2] },
	{ }
 };
 MODULE_DEVICE_TABLE(of, sh_cmt_of_table);
```

2. Arch timer development and upstreaming (part 1)

2.1 Task description:

Develop, upstream feature support and integrate the ARM arch timer driver on new SoCs and boards. The code shall be developed in an incremental fashion to fit latest upstream requirements. The main goal with this task is to prepare for R-Car Gen 3 support which may require remote access to real Gen 3 hardware or use of FPGA based target platforms, however SoCs and boards related to the R-Car Gen 2 family may also be covered on a best effort basis, including r8a7790 Lager, r8a7791 Koelsch, r8a7794 Alt and r8a7793 Gose. DT shall be used whenever possible together with Multiplatform support. Care should be taken to ease back porting to LTSI kernels. Patches shall be posted to community mailing lists for upstream merge.

2.2 Task deliverable #1:

2.2.1 Deliverable name:

description

2.2.2 Deliverable contents:

ARM: shmobile: Update HAVE_ARM_ARCH_TIMER Kconfig bits

[PATCH 01/02] ARM: shmobile: Select HAVE_ARM_ARCH_TIMER from Kconfig
[PATCH 02/02] ARM: shmobile: Drop HAVE_ARM_ARCH_TIMER from defconfig

Convert ARM mach-shmobile to select HAVE_ARM_ARCH_TIMER from Kconfig
based on if the enabled SoC contains ARM arch timer IP or not.

Recent SoCs like R-Car Gen2 series and r8a73a4 include Cortex-A7 and/or

Cortex-A15 hardware including arch timer, so select HAVE_ARM_ARCH_TIMER
in such case.

In case only older SoCs with Cortex-A9 are enabled then the arch timer
does not have to be enabled.

Signed-off-by: Magnus Damm <damm+renesas@opensource.se>

```
 arch/arm/configs/shmobile_defconfig |    1 -
 arch/arm/mach-shmobile/Kconfig      |    2 ++
 2 files changed, 2 insertions(+), 1 deletion(-)
```

2.3 Task deliverable #2:

2.3.1 Deliverable name:

linux-arm-shmobile-select-have-arm-arch-timer-from-kconfig.patch

2.3.2 Deliverable contents:

From: Magnus Damm <damm+renesas@opensource.se>

Recent SoCs like R-Car Gen2 series and r8a73a4 include Cortex-A7 and/or
Cortex-A15 hardware including arch timer, so select HAVE_ARM_ARCH_TIMER
in such case.

Signed-off-by: Magnus Damm <damm+renesas@opensource.se>

```
 arch/arm/mach-shmobile/Kconfig |    2 ++
 1 file changed, 2 insertions(+)

--- 0001/arch/arm/mach-shmobile/Kconfig
+++ work/arch/arm/mach-shmobile/Kconfig
@@ -21,6 +21,7 @@ config ARCH_RCAR_GEN2
 	select RENESAS_IRQC
 	select SYS_SUPPORTS_SH_CMT
 	select PCI_DOMAINS if PCI
+	select HAVE_ARM_ARCH_TIMER
 
 config ARCH_RMOBILE
 	bool
@@ -56,6 +57,7 @@ config ARCH_R8A73A4
 	bool "R-Mobile APE6 (R8A73A40)"
 	select ARCH_RMOBILE
 	select RENESAS_IRQC
+	select HAVE_ARM_ARCH_TIMER
 
 config ARCH_R8A7740
 	bool "R-Mobile A1 (R8A77400)"
```