Device tree

Bart Tanghe

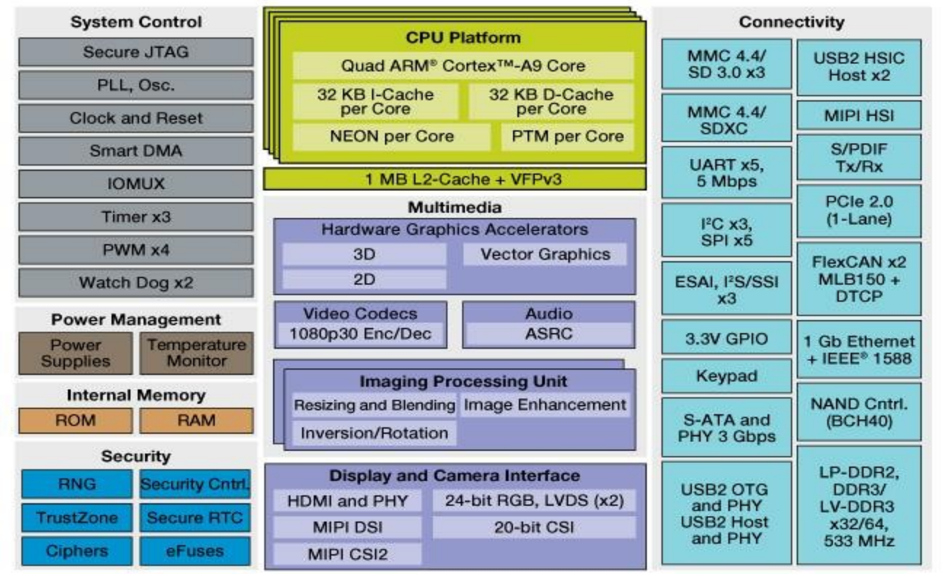
Board files (the past)

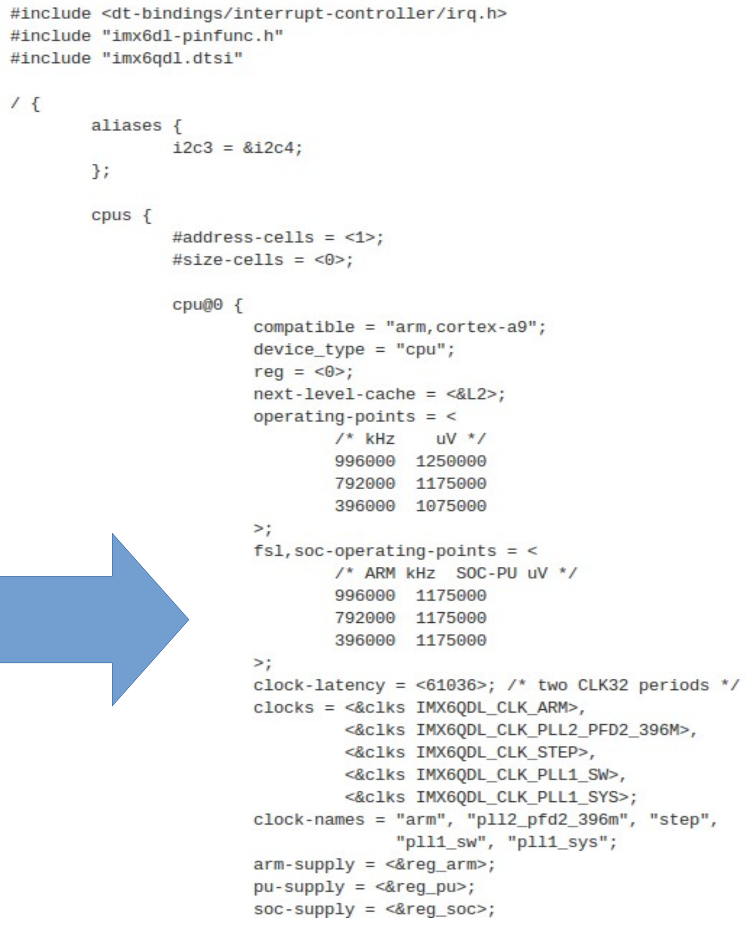
- Description of the hardware
- Example raspberry pi (rpi kernel tree)
  - Arch/arm/mach-bcm2708/bcm2708.c
  - Register devices
    - platform_device_register(pdev);
    - Registers, irq, clk

Device tree

- XML
- Description of the hardware
- http://www.devicetree.org/Main_Page

i.MX 6Quad Applications Processor Block Diagram





```
#include <dt-bindings/interrupt-controller/irq.h>
#include "imx6dl-pinfunc.h"
#include "imx6qdl.dtsi"

/ {
        aliases {
                i2c3 = &i2c4;
        };

        cpus {
                #address-cells = <1>;
                #size-cells = <0>;

                cpu@0 {
                        compatible = "arm,cortex-a9";
                        device_type = "cpu";
                        reg = <0>;
                        next-level-cache = <&L2>;
                        operating-points = <
                                /* kHz    uV */
                                996000  1250000
                                792000  1175000
                                396000  1075000
                        >;
                        fsl,soc-operating-points = <
                                /* ARM kHz  SOC-PU uV */
                                996000  1175000
                                792000  1175000
                                396000  1175000
                        >;
                        clock-latency = <61036>; /* two CLK32 periods */
                        clocks = <&clks IMX6QDL_CLK_ARM>,
                                 <&clks IMX6QDL_CLK_PLL2_PFD2_396M>,
                                 <&clks IMX6QDL_CLK_STEP>,
                                 <&clks IMX6QDL_CLK_PLL1_SW>,
                                 <&clks IMX6QDL_CLK_PLL1_SYS>;
                        clock-names = "arm", "pll2_pfd2_396m", "step",
                                      "pll1_sw", "pll1_sys";
                        arm-supply = <&reg_arm>;
                        pu-supply = <&reg_pu>;
                        soc-supply = <&reg_soc>;
```

Device tree

- Dts directory
  - arch/arm/boot/dts
  - Most common platforms
- Device tree bindings documentation
  - Documentation/devicetree/bindings
  - ex. pwm/mxs-pwm.txt

Device tree

Freescale MXS PWM controller

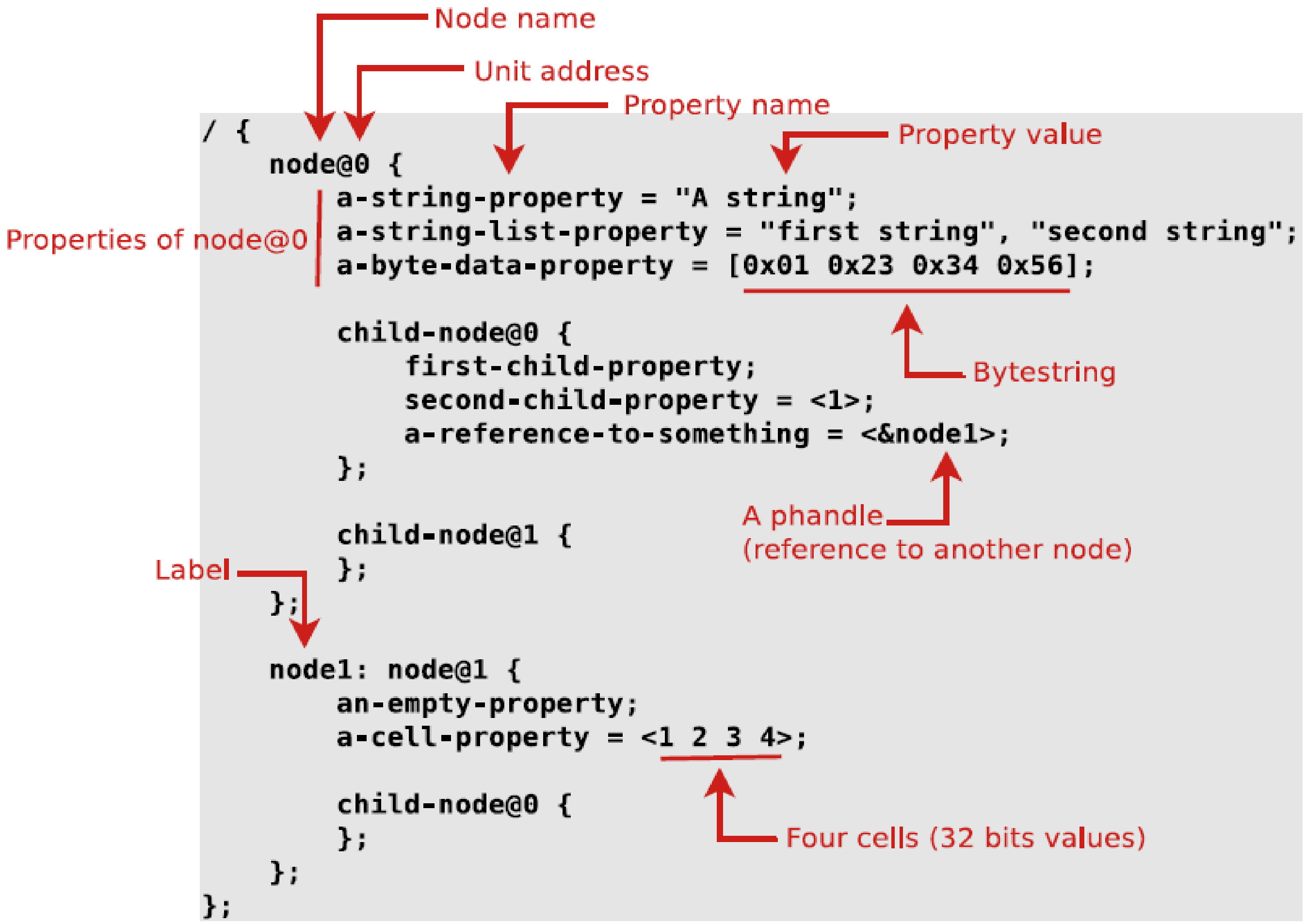
Required properties:

- compatible: should be "fsl,imx23-pwm"
- reg: physical base address and length of the controller's registers
- #pwm-cells: should be 2. See pwm.txt in this directory for a description of the cells format.
- fsl,pwm-number: the number of PWM devices

Example:

```
pwm: pwm@80064000 {
        compatible = "fsl,imx28-pwm", "fsl,imx23-pwm";
        reg = <0x80064000 0x2000>;
        #pwm-cells = <2>;
        fsl,pwm-number = <8>;
};
```

Device tree



```
/ {
    node@0 {
        a-string-property = "A string";
        a-string-list-property = "first string", "second string";
        a-byte-data-property = [0x01 0x23 0x34 0x56];

        child-node@0 {
            first-child-property;
            second-child-property = <1>;
            a-reference-to-something = <&node1>;
        };

        child-node@1 {
        };
    };

    node1: node@1 {
        an-empty-property;
        a-cell-property = <1 2 3 4>;

        child-node@0 {
        };
    };
};
```

Device tree compiler

- Device tree compiler (dtc)
- Device tree blob
- Binary code
- Dts => dtb
- Linux source directory

  -
    ```
    scripts/dtc/dtc -I dts -O dtb -o my-tree.dtb my-tree.dts
    ```

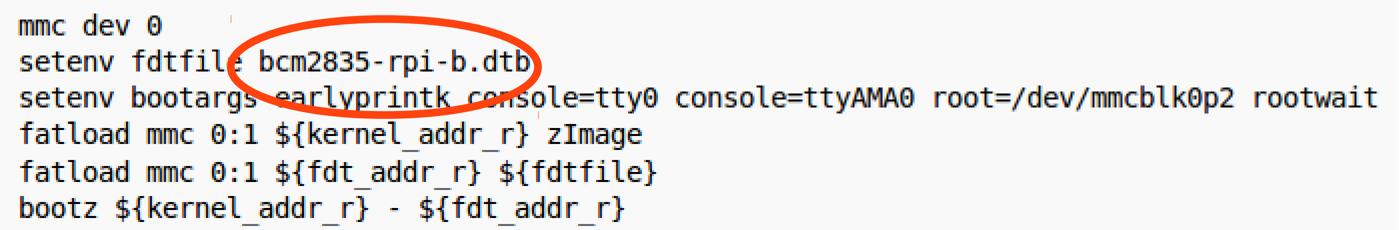
  -
    ```
    scripts/dtc/dtc -I dtb -O dts -o my-tree.dts my-tree.dtb
    ```

Device tree files

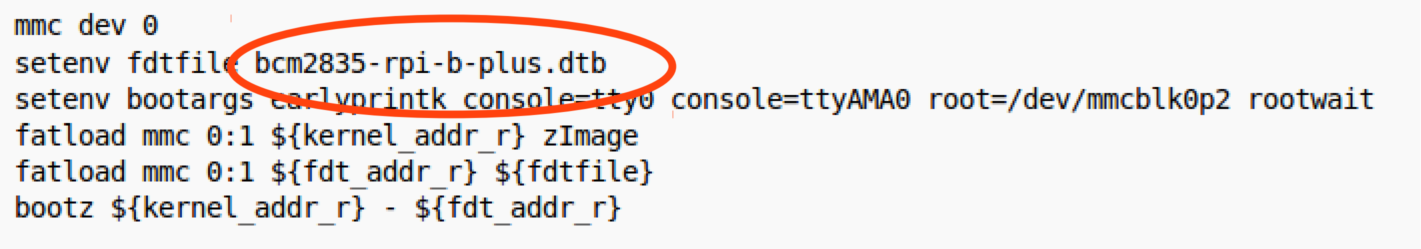
- Predefined dts and dtsi (dts include)
- Arch/arm/boot/dts
  - bcm2835.dtsi – bcm2835-rpi-b.dts
  - imx6qdl.dtsi – imx6qdl-udoo.dtsi – imx6q-udoo.dts
  - zynq-7000.dtsi – zynq-zed.dts

Device tree boot

- make ARCH=arm CROSS_COMPILE=arm-linux-gnueabi- dtbs -j4
- make ARCH=arm CROSS_COMPILE=arm-linux-gnueabi- zImage -j4
- U-boot

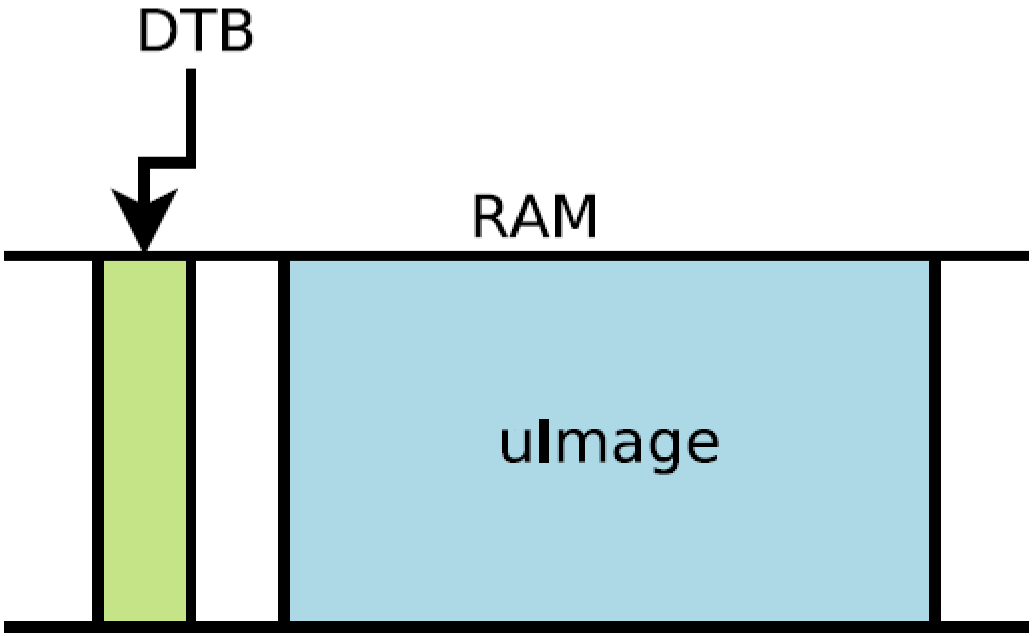


```
mmc dev 0
setenv fdtfile bcm2835-rpi-b.dtb
setenv bootargs earlyprintk console=tty0 console=ttyAMA0 root=/dev/mmcblk0p2 rootwait
fatload mmc 0:1 ${kernel_addr_r} zImage
fatload mmc 0:1 ${fdt_addr_r} ${fdtfile}
bootz ${kernel_addr_r} - ${fdt_addr_r}
```



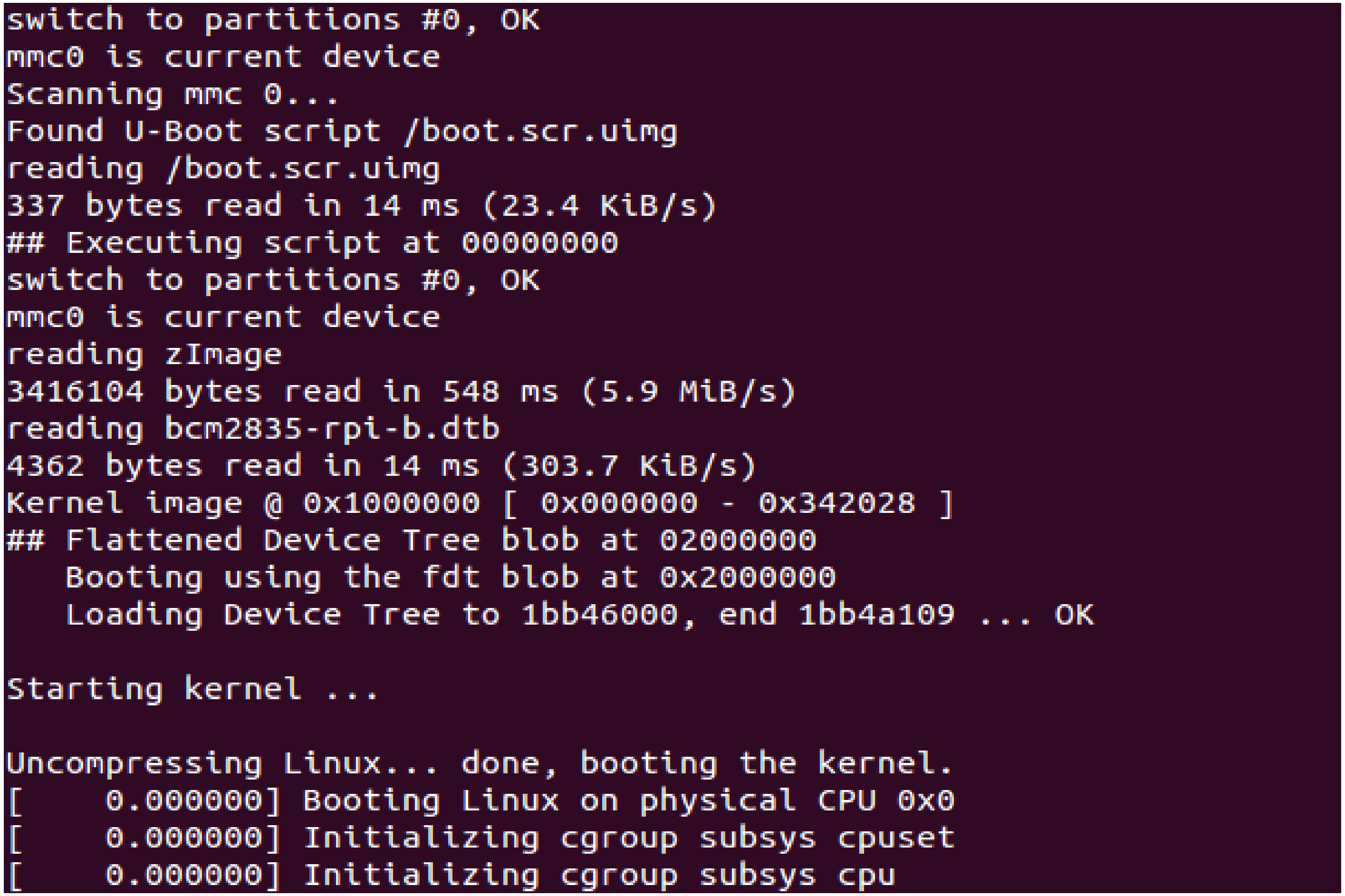
```
mmc dev 0
setenv fdtfile bcm2835-rpi-b-plus.dtb
setenv bootargs earlyprintk console=tty0 console=ttyAMA0 root=/dev/mmcblk0p2 rootwait
fatload mmc 0:1 ${kernel_addr_r} zImage
fatload mmc 0:1 ${fdt_addr_r} ${fdtfile}
bootz ${kernel_addr_r} - ${fdt_addr_r}
```

Device tree boot



```
r1 = don't care
r2 = <pointer to DTB>
```

Device tree boot output



```
switch to partitions #0, OK
mmc0 is current device
Scanning mmc 0...
Found U-Boot script /boot.scr.uimg
reading /boot.scr.uimg
337 bytes read in 14 ms (23.4 KiB/s)
## Executing script at 00000000
switch to partitions #0, OK
mmc0 is current device
reading zImage
3416104 bytes read in 548 ms (5.9 MiB/s)
reading bcm2835-rpi-b.dtb
4362 bytes read in 14 ms (303.7 KiB/s)
Kernel image @ 0x1000000 [ 0x000000 - 0x342028 ]
## Flattened Device Tree blob at 02000000
   Booting using the fdt blob at 0x2000000
   Loading Device Tree to 1bb46000, end 1bb4a109 ... OK

Starting kernel ...

Uncompressing Linux... done, booting the kernel.
[    0.000000] Booting Linux on physical CPU 0x0
[    0.000000] Initializing cgroup subsys cpuset
[    0.000000] Initializing cgroup subsys cpu
```

Bcm2835-rpi-b-plus.dts

```
/dts-v1/;
/include/ "bcm2835-rpi.dtsi"

/ {
    compatible = "raspberrypi,model-b-plus", "brcm,bcm2835";
    model = "Raspberry Pi Model B+";

    leds {
        act {
            gpios = <&gpio 47 0>;
        };

        pwr {
            label = "PWR";
            gpios = <&gpio 35 0>;
            default-state = "keep";
            linux,default-trigger = "default-on";
        };
    };
};

&gpio {
    pinctrl-0 = <&gpioout &alt0 &i2s_alt0 &alt3>;

    /* I2S interface */
    i2s_alt0: i2s_alt0 {
        brcm,pins = <18 19 20 21>;
        brcm,function = <4>; /* alt0 */
    };
};
```

Udoo quad

```
setenv bootargs 'console=ttymxc1,115200 root=/dev/mmcblk0p1  rootwait rw
rootfstype=ext4 consoleblank=0'
setenv fdt_addr 0x12000000
ext2load mmc ${mmcdev}:${mmcpart} ${fdt_addr} /boot/imx6q-udoo.dtb
ext2load mmc ${mmcdev}:${mmcpart} 10800000 /boot/uImage
bootm 10800000 - ${fdt_addr}
```

Udoo dual versus quad

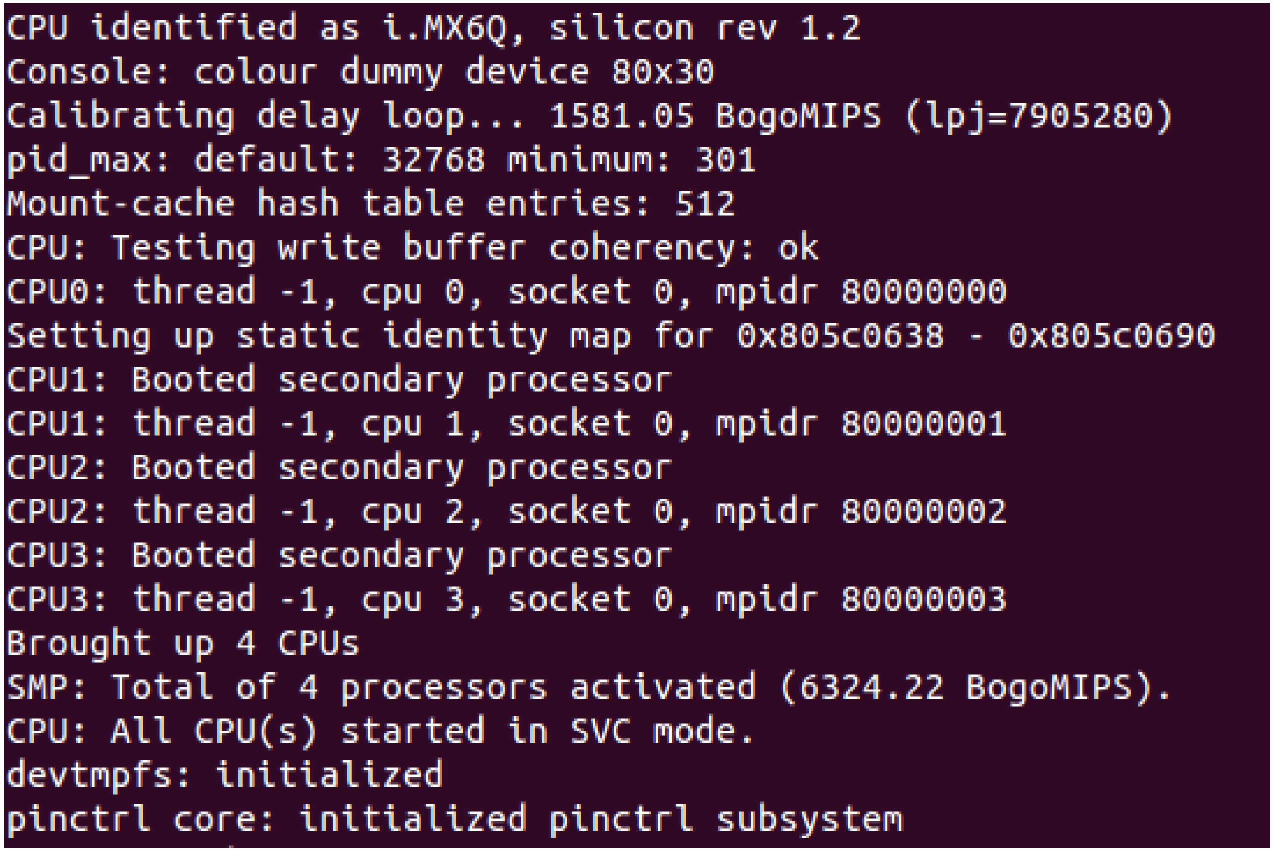
imx6dl.dtsi

```
cpus {
        #address-cells = <1>;
        #size-cells = <0>;

        cpu@0 {
            compatible = "arm,cortex-a9";
            device_type = "cpu";
            reg = <0>;
            next-level-cache = <&L2>;
            operating-points = <
                /* kHz    uV */
                996000  1250000
                792000  1175000
            ….



        };

        cpu@1 {
            compatible = "arm,cortex-a9";
            device_type = "cpu";
            reg = <1>;
            next-level-cache = <&L2>;
        };
    };

    soc {
```

Imx6q.dtsi

```
cpus {
    #address-cells = <1>;
    #size-cells = <0>;

    cpu@0 {
        compatible = "arm,cortex-a9";
        device_type = "cpu";
        reg = <0>;
        …

        soc-supply = <&reg_soc>;
    };

    cpu@1 {
        compatible = "arm,cortex-a9";
        device_type = "cpu";
        reg = <1>;
        next-level-cache = <&L2>;
    };

    cpu@2 {
        compatible = "arm,cortex-a9";
        device_type = "cpu";
        reg = <2>;
        next-level-cache = <&L2>;
    };

    cpu@3 {
        compatible = "arm,cortex-a9";
        device_type = "cpu";
        reg = <3>;
        next-level-cache = <&L2>;
    };
};

soc {
```

Udoo dual versus quad



```
CPU identified as i.MX6Q, silicon rev 1.2
Console: colour dummy device 80x30
Calibrating delay loop... 1581.05 BogoMIPS (lpj=7905280)
pid_max: default: 32768 minimum: 301
Mount-cache hash table entries: 512
CPU: Testing write buffer coherency: ok
CPU0: thread -1, cpu 0, socket 0, mpidr 80000000
Setting up static identity map for 0x805c0638 - 0x805c0690
CPU1: Booted secondary processor
CPU1: thread -1, cpu 1, socket 0, mpidr 80000001
CPU2: Booted secondary processor
CPU2: thread -1, cpu 2, socket 0, mpidr 80000002
CPU3: Booted secondary processor
CPU3: thread -1, cpu 3, socket 0, mpidr 80000003
Brought up 4 CPUs
SMP: Total of 4 processors activated (6324.22 BogoMIPS).
CPU: All CPU(s) started in SVC mode.
devtmpfs: initialized
pinctrl core: initialized pinctrl subsystem
```

Compatible string

DTS

```
pwm: pwm@7e20e000 {
                compatible = "brcm,bcm2835-pwm";
                reg = <0x7e20c000 0x28>;
                clocks = <&clk_pwm>;
                #pwm-cells = <2>;
                status = "disabled";
                //status = “ok”
        };
...

        clk_pwm: clock@3 {
                compatible = "fixed-clock";
                reg = <3>;
                #clock-cells = <0>;
                clock-output-names = "pwm";
                clock-frequency = <9200000>;
        };
```

Driver code

```
static int bcm2835_pwm_probe(struct platform_device *pdev)      //probe
...
        r = platform_get_resource(pdev, IORESOURCE_MEM, 0);
        pwm->base = devm_ioremap_resource(&pdev->dev, r);

…
        clk = devm_clk_get(&pdev->dev, NULL);
…

        static const struct of_device_id bcm2835_pwm_of_match[] = {
                { .compatible = "brcm,bcm2835-pwm", },
                { /* sentinel */ }
        };

        MODULE_DEVICE_TABLE(of, bcm2835_pwm_of_match);
```

Compatible with board files

```
        static struct platform_driver bcm2835_pwm_driver = {
                .driver = {
                        .name = "bcm2835-pwm",
                        .of_match_table = bcm2835_pwm_of_match,
                },
                .probe = bcm2835_pwm_probe,
                .remove = bcm2835_pwm_remove,
        };
        module_platform_driver(bcm2835_pwm_driver);
```

Clocks – fixed clock (root clock)

DTS

```
        pwm: pwm@7e20e000 {
                        compatible = "brcm,bcm2835-pwm";
                        reg = <0x7e20c000 0x28>;
                        clocks = <&clk_pwm>;
                        #pwm-cells = <2>;
                        status = "disabled";
                        //status = “ok”
                };
        ...
                clk_pwm: clock@3 {
                        compatible = "fixed-clock";
                        reg = <3>;
                        #clock-cells = <0>;
                        clock-output-names = "pwm";
                        clock-frequency = <9200000>;
                };
```

Driver code

```
static int bcm2835_pwm_probe(struct platform_device *pdev)       //probe
...
        r = platform_get_resource(pdev, IORESOURCE_MEM, 0);
        pwm->base = devm_ioremap_resource(&pdev->dev, r);

…
        clk = devm_clk_get(&pdev->dev, NULL);
…

        static const struct of_device_id bcm2835_pwm_of_match[] = {
                { .compatible = "brcm,bcm2835-pwm", },
                { /* sentinel */ }
        };

        MODULE_DEVICE_TABLE(of, bcm2835_pwm_of_match);
```

Compatible with board files

```
        static struct platform_driver bcm2835_pwm_driver = {
                .driver = {
                        .name = "bcm2835-pwm",
                        .of_match_table = bcm2835_pwm_of_match,
                },
                .probe = bcm2835_pwm_probe,
                .remove = bcm2835_pwm_remove,
        };
        module_platform_driver(bcm2835_pwm_driver);
```

Registers

DTS

```
    pwm: pwm@7e20e000 {
                compatible = "brcm,bcm2835-pwm";
                reg = <0x7e20c000 0x28>;
                clocks = <&clk_pwm>;
                #pwm-cells = <2>;
                status = "disabled";
                //status = “ok”
        };
    ...

        clk_pwm: clock@3 {
                compatible = "fixed-clock";
                reg = <3>;
                #clock-cells = <0>;
                clock-output-names = "pwm";
                clock-frequency = <9200000>;
        };
```

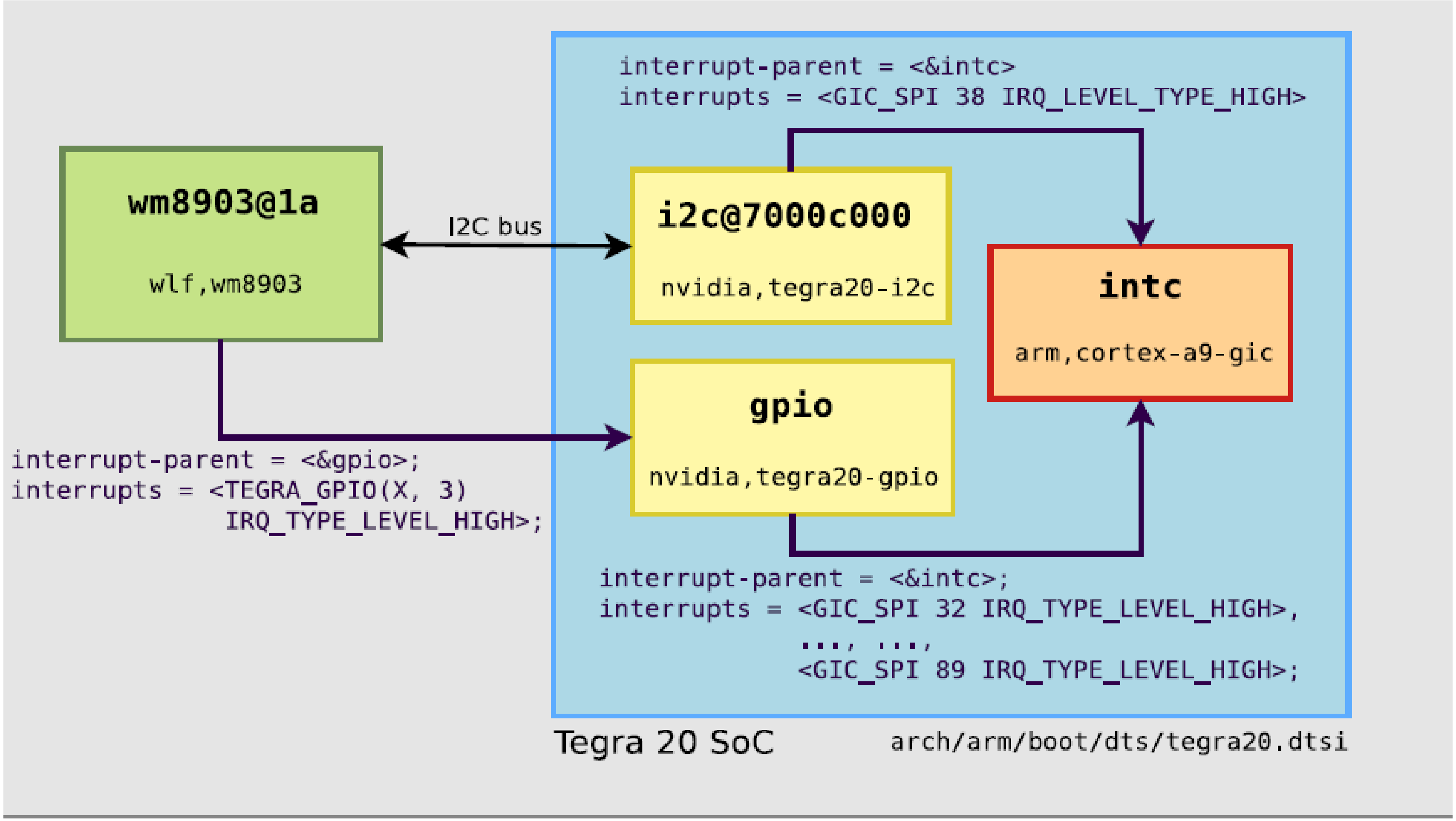
Driver code

```
static int bcm2835_pwm_probe(struct platform_device *pdev)      //probe
...
        r = platform_get_resource(pdev, IORESOURCE_MEM, 0);
        pwm->base = devm_ioremap_resource(&pdev->dev, r);

…
        clk = devm_clk_get(&pdev->dev, NULL);
…

        static const struct of_device_id bcm2835_pwm_of_match[] = {
                { .compatible = "brcm,bcm2835-pwm", },
                { /* sentinel */ }
        };

        MODULE_DEVICE_TABLE(of, bcm2835_pwm_of_match);
```

Compatible with board files

```
        static struct platform_driver bcm2835_pwm_driver = {
                .driver = {
                        .name = "bcm2835-pwm",
                        .of_match_table = bcm2835_pwm_of_match,
                },
                .probe = bcm2835_pwm_probe,
                .remove = bcm2835_pwm_remove,
        };
        module_platform_driver(bcm2835_pwm_driver);
```

Interrupt



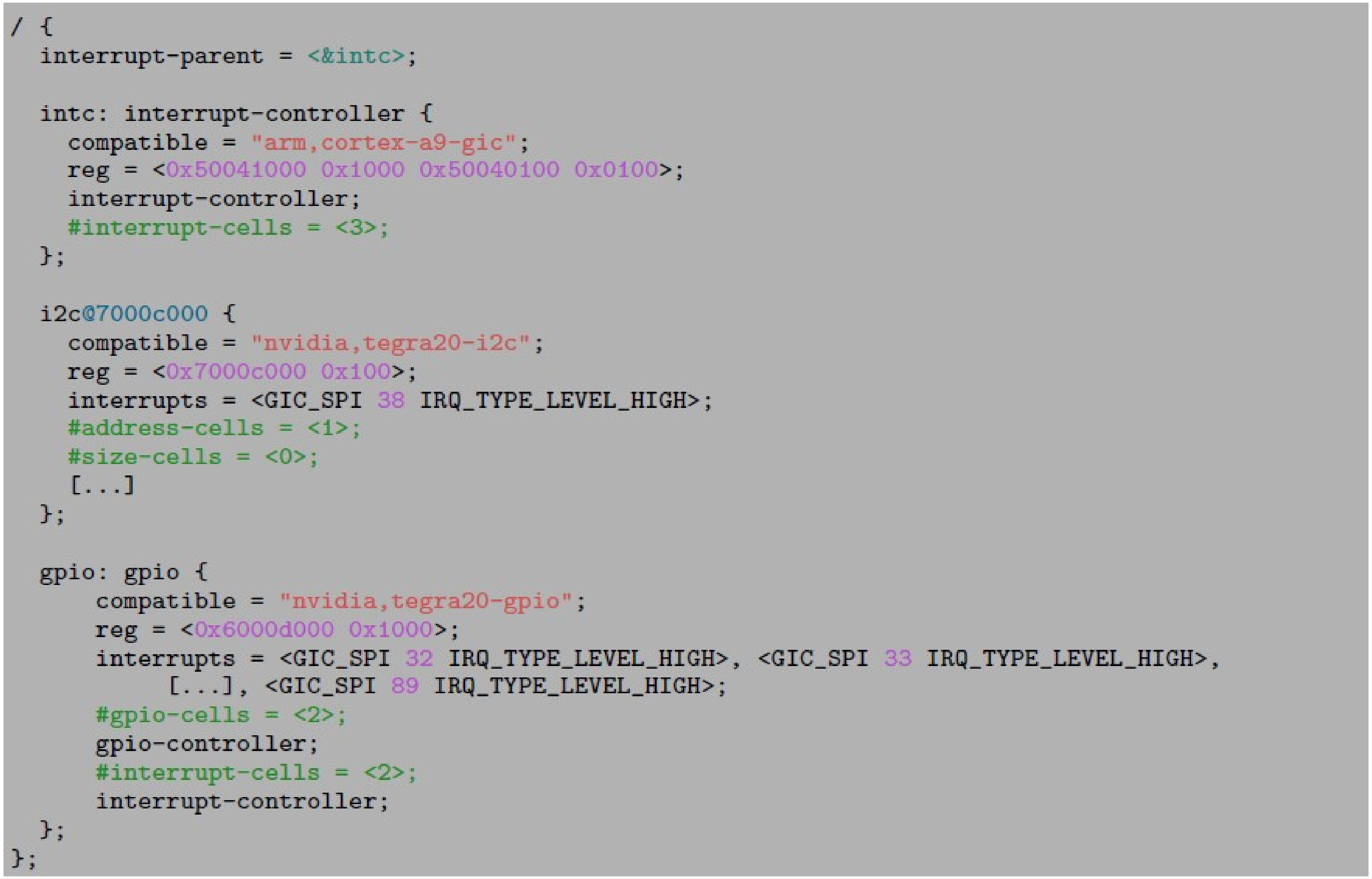
```
interrupt-parent = <&intc>
interrupts = <GIC_SPI 38 IRQ_LEVEL_TYPE_HIGH>
```

```
interrupt-parent = <&gpio>;
interrupts = <TEGRA_GPIO(X, 3)
              IRQ_TYPE_LEVEL_HIGH>;
```

```
interrupt-parent = <&intc>;
interrupts = <GIC_SPI 32 IRQ_TYPE_LEVEL_HIGH>,
             ..., ...,
             <GIC_SPI 89 IRQ_TYPE_LEVEL_HIGH>;
```

arch/arm/boot/dts/tegra20.dtsi

Interrupt



```
/ {
  interrupt-parent = <&intc>;

  intc: interrupt-controller {
    compatible = "arm,cortex-a9-gic";
    reg = <0x50041000 0x1000 0x50040100 0x0100>;
    interrupt-controller;
    #interrupt-cells = <3>;
  };

  i2c@7000c000 {
    compatible = "nvidia,tegra20-i2c";
    reg = <0x7000c000 0x100>;
    interrupts = <GIC_SPI 38 IRQ_TYPE_LEVEL_HIGH>;
    #address-cells = <1>;
    #size-cells = <0>;
    [...]
  };

  gpio: gpio {
      compatible = "nvidia,tegra20-gpio";
      reg = <0x6000d000 0x1000>;
      interrupts = <GIC_SPI 32 IRQ_TYPE_LEVEL_HIGH>, <GIC_SPI 33 IRQ_TYPE_LEVEL_HIGH>,
           [...], <GIC_SPI 89 IRQ_TYPE_LEVEL_HIGH>;
      #gpio-cells = <2>;
      gpio-controller;
      #interrupt-cells = <2>;
      interrupt-controller;
  };
};
```

Device tree kernel functions

drivers/of/base.c

* Procedures for creating, accessing and interpreting the device tree.

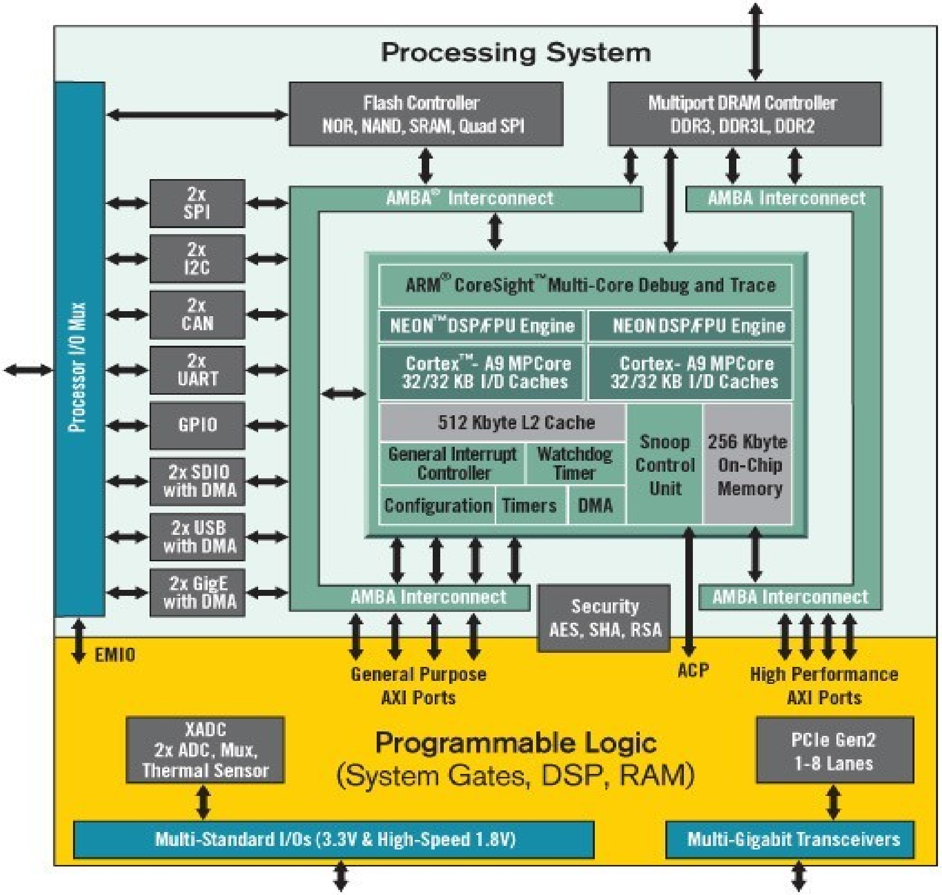
```
/*
 * Find a property with a given name for a given node
 * and return the value.
 */
const void *of_get_property(const struct device_node *np, const char *name,
                            int *lenp)
{
        struct property *pp = of_find_property(np, name, lenp);

        return pp ? pp->value : NULL;
}
EXPORT_SYMBOL(of_get_property);
```

Example drivers/clk/clkdev.c

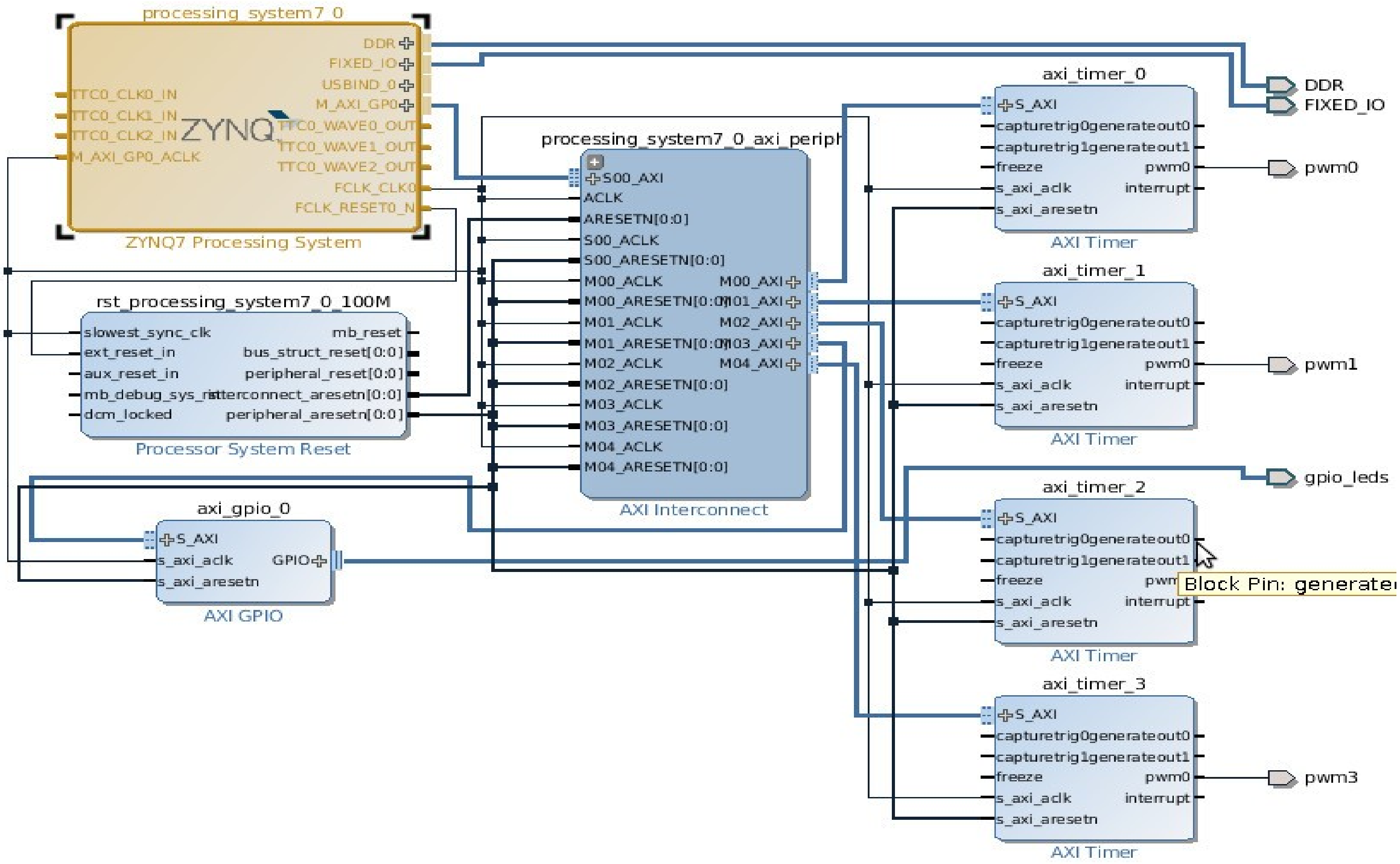
```
                if (np && !of_get_property(np, "clock-ranges", NULL))
```

Zedboard

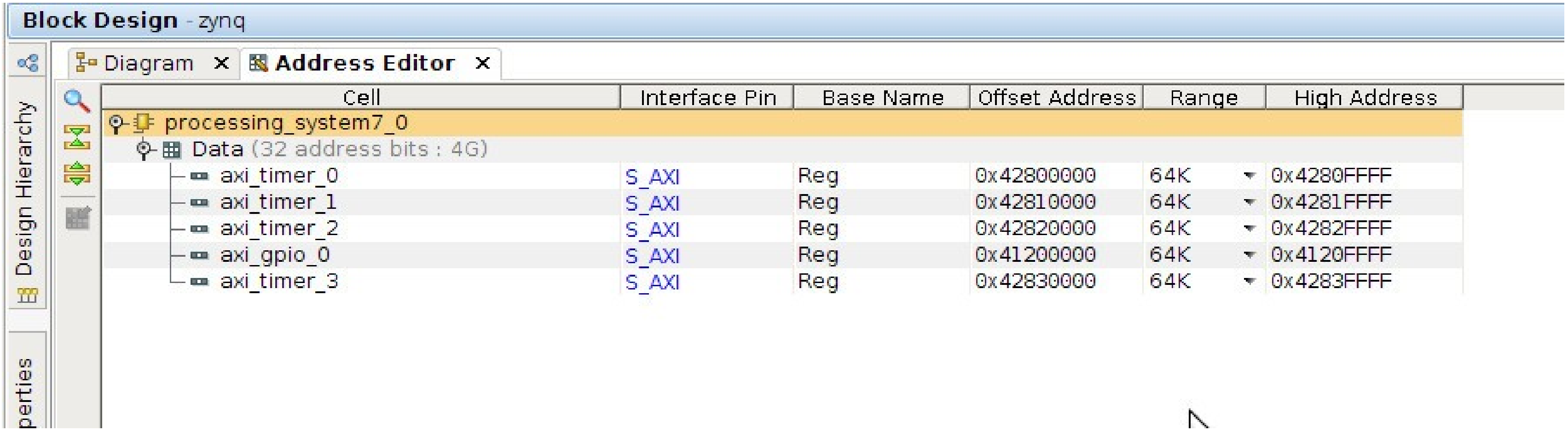
ZedBoard


Processing System
Flash Controller
NOR, NAND, SRAM, Quad SPI
Multiport DRAM Controller
DDR3, DDR3L, DDR2
Processor I/O Mux
2x SPI
2x I2C
2x CAN
2x UART
GPIO
2x SDIO with DMA
2x USB with DMA
2x GigE with DMA
AMBA® Interconnect
AMBA Interconnect
ARM® CoreSight™ Multi-Core Debug and Trace
NEON™ DSP/FPU Engine
NEON DSP/FPU Engine
Cortex™- A9 MPCore
32/32 KB I/D Caches
Cortex- A9 MPCore
32/32 KB I/D Caches
512 Kbyte L2 Cache
General Interrupt Controller
Watchdog Timer
Configuration
Timers
DMA
Snoop Control Unit
256 Kbyte On-Chip Memory
AMBA Interconnect
Security
AES, SHA, RSA
AMBA Interconnect
EMIO
General Purpose
AXI Ports
ACP
High Performance
AXI Ports
XADC
2x ADC, Mux,
Thermal Sensor
Programmable Logic
(System Gates, DSP, RAM)
PCIe Gen2
1-8 Lanes
Multi-Standard I/Os (3.3V & High-Speed 1.8V)
Multi-Gigabit Transceivers

Zedboard – add pwm to design



Zedboard – add pwm to design

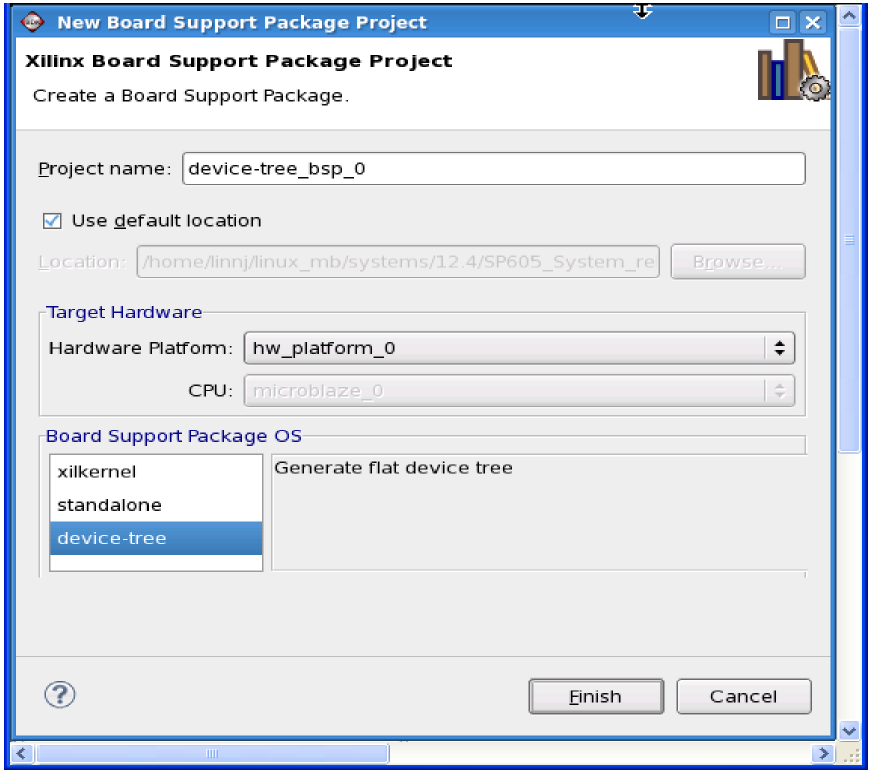


Cell	Interface Pin	Base Name	Offset Address	Range	High Address
processing_system7_0					
Data (32 address bits : 4G)					
axi_timer_0	S_AXI	Reg	0x42800000	64K	0x4280FFFF
axi_timer_1	S_AXI	Reg	0x42810000	64K	0x4281FFFF
axi_timer_2	S_AXI	Reg	0x42820000	64K	0x4282FFFF
axi_gpio_0	S_AXI	Reg	0x41200000	64K	0x4120FFFF
axi_timer_3	S_AXI	Reg	0x42830000	64K	0x4283FFFF

Zedboard – add pwm to design

https://github.com/Xilinx/device-tree-xlnx

Generate the dts file



New Board Support Package Project

Xilinx Board Support Package Project

Create a Board Support Package.

Project name: device-tree_bsp_0

☑ Use default location

Location: /home/linnj/linux_mb/systems/12.4/SP605_System_re

Target Hardware

Hardware Platform: hw_platform_0

CPU: microblaze_0

Board Support Package OS

- xilkernel
- standalone
- device-tree

Generate flat device tree

Zedboard – dts

```
axi_timer_0: timer@42800000 {
                clock-frequency = <100000000>;
                #pwm-cells = <1>;
                clocks = <&clkc 15>;
                compatible = "xlnx,xlnx-pwm";
                reg = <0x42800000 0x10000>;
                xlnx,count-width = <0x20>;
                xlnx,gen0-assert = <0x1>;
                xlnx,gen1-assert = <0x1>;
                xlnx,one-timer-only = <0x0>;
                xlnx,trig0-assert = <0x1>;
                xlnx,trig1-assert = <0x1>;

axi_timer_1: timer@42810000 {
                clock-frequency = <100000000>;
                #pwm-cells = <1>;
                clocks = <&clkc 15>;
                compatible = "xlnx,xlnx-pwm";
                reg = <0x42810000 0x10000>;
```

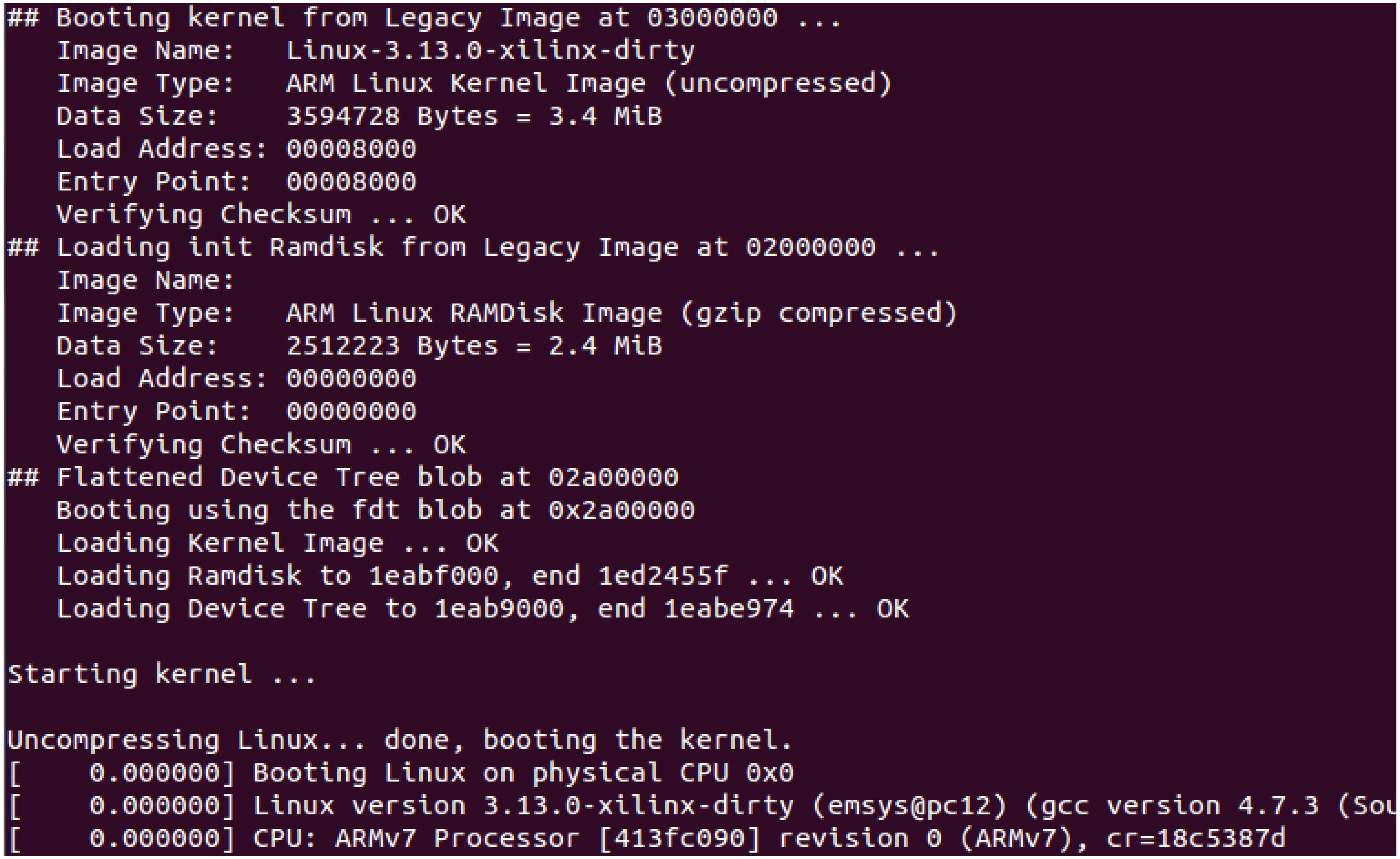
Zedboard

SD-card

- boot.bin
  - (first stage bootloader, hardware configuration, u-boot)
- uImage
- uramdisk

```
u-boot> fatload mmc 0 0x3000000 uImage
u-boot> fatload mmc 0 0x2A00000 devicetree.dtb
u-boot> fatload mmc 0 0x2000000 uramdisk.image.gz
u-boot> bootm 0x3000000 0x2000000 0x2A00000
```

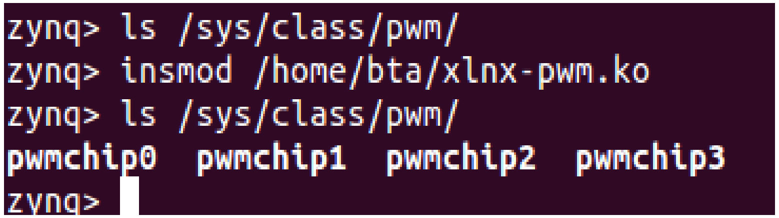
Zedboard



```
## Booting kernel from Legacy Image at 03000000 ...
   Image Name:   Linux-3.13.0-xilinx-dirty
   Image Type:   ARM Linux Kernel Image (uncompressed)
   Data Size:    3594728 Bytes = 3.4 MiB
   Load Address: 00008000
   Entry Point:  00008000
   Verifying Checksum ... OK
## Loading init Ramdisk from Legacy Image at 02000000 ...
   Image Name:
   Image Type:   ARM Linux RAMDisk Image (gzip compressed)
   Data Size:    2512223 Bytes = 2.4 MiB
   Load Address: 00000000
   Entry Point:  00000000
   Verifying Checksum ... OK
## Flattened Device Tree blob at 02a00000
   Booting using the fdt blob at 0x2a00000
   Loading Kernel Image ... OK
   Loading Ramdisk to 1eabf000, end 1ed2455f ... OK
   Loading Device Tree to 1eab9000, end 1eabe974 ... OK

Starting kernel ...

Uncompressing Linux... done, booting the kernel.
[    0.000000] Booting Linux on physical CPU 0x0
[    0.000000] Linux version 3.13.0-xilinx-dirty (emsys@pc12) (gcc version 4.7.3 (Sou
[    0.000000] CPU: ARMv7 Processor [413fc090] revision 0 (ARMv7), cr=18c5387d
```

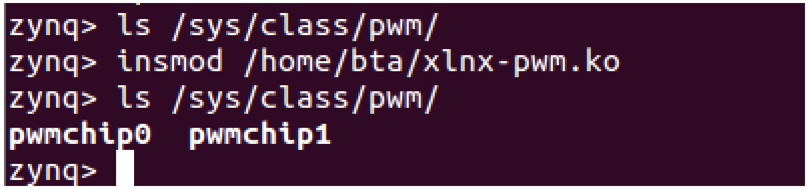
Zedboard

devicetree_4pwm.dtb



```
zynq> ls /sys/class/pwm/
zynq> insmod /home/bta/xlnx-pwm.ko
zynq> ls /sys/class/pwm/
pwmchip0  pwmchip1  pwmchip2  pwmchip3
zynq>
```

devicetree_2pwm.dtb



```
zynq> ls /sys/class/pwm/
zynq> insmod /home/bta/xlnx-pwm.ko
zynq> ls /sys/class/pwm/
pwmchip0  pwmchip1
zynq>
```

Look at the source with

scripts/dtc/dtc -I dtb -O dts -o /devicetree_2pwm.dts system_2pwm.dtb

DTS-example

https://github.com/btanghe/device_tree/

```
example@0 {
        compatible = "general,dts-example";
        reg = <0xdeadbeef 0x40>;
        clocks = <0x2 0xa1>;
        clock-names = "per";
        custom-var = <0x200>;
        interrupts = <0x0 0x1b 0x4>;
        status = "okay";
};
```

Demo's and questions

Resource

Device tree for dummies
http://events.linuxfoundation.org/sites/events/files/slides/petazzoni-device-tree-dummies.pdf

Xillybus zynq device tree
http://www.xillybus.com/tutorials/device-tree-zynq-2

Linux kernel tree