



STM32MP15 clock tree



Contents

1. STM32MP15 clock tree	3
2. Boot chain overview	15
3. Category:STM32MP15 Discovery kits	28
4. Category:STM32MP15 Evaluation boards	42
5. How to change the CPU frequency	55
6. RCC internal peripheral	67
7. Resource manager for coprocessing	80
8. STM32CubeMP1 Package	93
9. STM32CubeMX	106
10. STM32MP15 ROM code overview	119
11. STM32MP15 device tree	132
12. TF-A overview	145
13. U-Boot overview	158



A quality version of this page, accepted on 17 March 2021, was based off this revision.

All the peripherals receive one or several clocks that are generated via **RCC internal peripheral**. RCC relies on several clocks sources (LSI, LSE, HSI, HSE, CSI) and four PLL in order to provide adequate input frequencies to all the peripherals. The clock tree covers all the system clock distribution aspects, from the clock sources to the consumer peripherals (internal and external), except clock gating management that is locally controlled by each peripheral driver.

Contents

1 Overview	4
2 How to build a clock tree?	5
3 ST boards clock tree	6
3.1 STM32MP157x-EV1 case	6
3.1.1 Clock tree	6
3.1.1.1 Device tree	8
3.2 STM32MP157x-DK2 case	10
3.2.1 Clock tree	10
3.2.2 Device tree	12



1 Overview

The clock tree is managed via RCC internal peripheral hardware block and it is configured at different steps from the Cortex-A7:

- When the device is **reset**, all RCC registers take their reset values: the four PLL are disabled and most of the clock source selectors are pointing to the HSI.
- The ROM Code configures the minimum clock tree needed to boot on the selected boot device.
- The **FSBL** (Cf. [Boot chain overview](#)) completely configures the clock tree as expected, all the way up to Linux, thanks to the configuration given in the [device tree](#).
 - U-Boot SPL (basic boot chain) binding is available in [doc/device-tree-bindings/clock/st,stm32mp1.txt](#)
 - TF-A (trusted boot chain) binding follows the same syntax as U-Boot SPL
- **Linux** is responsible for the clock tree at runtime and it may partly modify it thanks to information taken from the [device tree](#):
 - Linux binding is available in [Documentation/devicetree/bindings/clock/clock-bindings.txt](#) (and surrounding files): 'fixed-clock' compatible, 'clocks' and 'assigned-clocks' properties are important concepts to understand the management of clocks providers/consumers.

Notice that [STM32Cube](#) (running on the Cortex-M4) does not control the clock tree setup so the configuration applied by the Cortex-A7 must provide suitable clocks to Cortex-M4, like it is explained in [resource manager](#) description. One exception to this rule is the [engineering boot](#), allowing to directly load (and debug) the Cortex-M4 whereas the Cortex-A7 execution is stalled in the ROM code.



2 How to build a clock tree?

Building a clock tree is quite complex as it needs to take into account the constraints set by each internal and external peripheral, including external clock sources.

We encourage the use of [STM32CubeMX](#) to build the clock tree, and avoid having to know all internal peripherals details: the tool allows to select the peripherals that will be present on the board, fix the clock sources frequencies and automatically find an optimized clock tree. It is then able to generate the device tree that is directly consumed by the boot chain and Linux.



3 ST boards clock tree

 See [How to change the CPU frequency](#) article for more information about the CPU frequency setting (PLL1P), including dynamic voltage and frequency scaling (DVFS) configuration.

3.1 STM32MP157x-EV1 case

This chapter shows the boot time clock tree set by the FSBL on STM32MP157x-EV1 evaluation board. Linux eventual runtime modifications are not covered here.

3.1.1 Clock tree

The following table shows what STM32MP157x-EV1 clock tree looks like, as a result of the boot chain execution with the device tree built with STM32CubeMX.

Component Comment	Parent	Frequency	Used?	
LSI	N.A.	0.032000 MHz	yes	Mandatory for IWDG, DAC
DAC	LSI	0.032000 MHz	yes	
RNG1	LSI	0.032000 MHz	yes	Input frequency should be
as low as possible				
RNG2	LSI	0.032000 MHz	yes	Input frequency should be
as low as possible				
IWDG1	LSI	0.032000 MHz	yes	
IWDG2	LSI	0.032000 MHz	yes	
LSE	N.A.	0.032768 MHz	yes	Mandatory for DTS
RTC	LSE	0.032768 MHz	yes	
TAMP	LSE	0.032768 MHz	yes	
CEC	LSE	0.032768 MHz	yes	
LPTIM4	LSE	0.032768 MHz	yes	
LPTIM5	LSE	0.032768 MHz	yes	
HSI	N.A.	64.000000 MHz	yes	
SPI4	HSI	64.000000 MHz	no	
SPI5	HSI	64.000000 MHz	no	
SPI6	HSI	64.000000 MHz	yes	
I2C4	HSI	64.000000 MHz	yes	PMIC
I2C6	HSI	64.000000 MHz	no	
I2C1	HSI	64.000000 MHz	no	
I2C2	HSI	64.000000 MHz	yes	Rpi and peripherals
I2C3	HSI	64.000000 MHz	no	
I2C5	HSI	64.000000 MHz	yes	Rpi
USART1	HSI	64.000000 MHz	yes	
USART2	HSI	64.000000 MHz	no	
USART3	HSI	64.000000 MHz	yes	Rpi
UART4	HSI	64.000000 MHz	yes	Linux console
UART5	HSI	64.000000 MHz	no	
USART6	HSI	64.000000 MHz	no	
UART7	HSI	64.000000 MHz	no	
UART8	HSI	64.000000 MHz	no	
MC01	HSI	64.000000 MHz	no	Available so can be used
HSE	N.A.	24.000000 MHz	yes	
DSIPLL	HSE	125.000000 MHz	no	DSI DPHY PLL
DSIBL	DSIPLL	125.000000 MHz	no	DSI lanebyte clock
RTCDIV	HSE	1.000000 MHz	yes	Only used when RTC source
is HSE				



ck_per	HSE	24.000000	MHz	yes	
ADC	ck_per	24.000000	MHz	yes	Can use internal divider to
be < 40MHz					
PLL1	HSE	xxx	MHz	yes	
PLL1P	PLL1	xxx	MHz	yes	
MPUDIV	PLL1P	xxx	MHz	yes	
Cortex-A7	PLL1P	xxx	MHz	yes	650 MHz or 800 MHz, see the
information box above					
MC02	Cortex-A7	xxx	MHz	no	Available so can be used
PLL2	HSE	1066.000000	MHz	yes	
PLL2P	PLL2	266.500000	MHz	yes	
AXI	PLL2P	266.500000	MHz	yes	< 266MHz
FMC	AXI	266.500000	MHz	yes	NAND flash
QSPI	AXI	266.500000	MHz	yes	NOR flash
SYSRAM	AXI	266.500000	MHz	yes	
ROM	AXI	266.500000	MHz	yes	
AHB5	AXI	266.500000	MHz	yes	< 266MHz
CRYP1	AHB5	266.500000	MHz	yes	
HASH1	AHB5	266.500000	MHz	yes	
GPI0Z	AHB5	266.500000	MHz	yes	
BKPSRAM	AHB5	266.500000	MHz	yes	
AHB6	AXI	266.500000	MHz	yes	< 266MHz
CRC1	AHB6	266.500000	MHz	yes	
MDMA	AHB6	266.500000	MHz	yes	
USBH	AHB6	266.500000	MHz	yes	USB Host
APB4	AHB6	133.250000	MHz	yes	
APB5	AHB6	66.625000	MHz	yes	
BSEC	APB5	66.625000	MHz	yes	< 67MHz
ETZPC	APB5	66.625000	MHz	yes	
TZC	APB5	66.625000	MHz	yes	
DBGAPB	AXI	133.250000	MHz	yes	JTAG & Coresight
DBGMCU	DBGAPB	66.625000	MHz	yes	
STM	DBGAPB	66.625000	MHz	yes	
PLL2Q	PLL2	533.000000	MHz	yes	
GPU	PLL2Q	533.000000	MHz	yes	< 533MHz
PLL2R	PLL2	533.000000	MHz	yes	
DDR	PLL2R	533.000000	MHz	yes	< 533MHz
PLL3	HSE	417.755859	MHz	yes	
PLL3P	PLL3	208.877930	MHz	yes	
MLAHB	PLL3P	208.877930	MHz	yes	< 209MHz
Cortex-M4	MLAHB	208.877930	MHz	yes	
SRAM1	MLAHB	208.877930	MHz	yes	
SRAM2	MLAHB	208.877930	MHz	yes	
SRAM3	MLAHB	208.877930	MHz	yes	
RETRAM	MLAHB	208.877930	MHz	yes	
AHB1	MLAHB	208.877930	MHz	yes	< 209MHz
AHB2	MLAHB	208.877930	MHz	yes	< 209MHz
DMA1	AHB2	208.877930	MHz	yes	
DMA2	AHB2	208.877930	MHz	yes	
DMAMUX	AHB2	208.877930	MHz	yes	
APB1	MLAHB	104.438965	MHz	yes	
LPTIM1	APB1	104.438965	MHz	yes	
WWDG	APB1	104.438965	MHz	yes	
APB2	MLAHB	104.438965	MHz	yes	
TIM2	MLAHB	208.877930	MHz	yes	TIM Group 1
TIM3	MLAHB	208.877930	MHz	yes	TIM Group 1
TIM4	MLAHB	208.877930	MHz	yes	TIM Group 1
TIM5	MLAHB	208.877930	MHz	yes	TIM Group 1
TIM6	MLAHB	208.877930	MHz	yes	TIM Group 1
TIM7	MLAHB	208.877930	MHz	yes	TIM Group 1
TIM12	MLAHB	208.877930	MHz	yes	TIM Group 1
TIM13	MLAHB	208.877930	MHz	yes	TIM Group 1
TIM14	MLAHB	208.877930	MHz	yes	TIM Group 1
TIM1	MLAHB	208.877930	MHz	yes	TIM Group 2
TIM8	MLAHB	208.877930	MHz	yes	TIM Group 2
TIM15	MLAHB	208.877930	MHz	yes	TIM Group 2
TIM16	MLAHB	208.877930	MHz	yes	TIM Group 2



TIM17	MLAHB	208.877930	MHz	yes	TIM Group 2
APB3	MLAHB	104.438965	MHz	yes	
LPTIM2	APB3	104.438965	MHz	yes	
LPTIM3	APB3	104.438965	MHz	yes	
SYSCFG	APB3	104.438965	MHz	yes	
VREFBUF	APB3	104.438965	MHz	yes	
DTS	APB3	104.438965	MHz	yes	
HDP	APB3	104.438965	MHz	yes	
AHB3	MLAHB	208.877930	MHz	yes	< 209MHz
CRC2	AHB3	208.877930	MHz	yes	
CRYP2	AHB3	208.877930	MHz	yes	
HASH2	AHB3	208.877930	MHz	yes	
DCMI	AHB3	208.877930	MHz	yes	Camera
IPCC	AHB3	208.877930	MHz	yes	Mailbox
AHB4	MLAHB	208.877930	MHz	yes	< 209MHz
PWR	AHB4	208.877930	MHz	yes	
RCC	AHB4	208.877930	MHz	yes	
GPIOA-K	AHB4	208.877930	MHz	yes	
EXTI	AHB4	208.877930	MHz	yes	
PLL3Q	PLL3	24.573874	MHz	yes	
SPI1	PLL3Q	24.573874	MHz	no	
SPI2	PLL3Q	24.573874	MHz	no	
SPI3	PLL3Q	24.573874	MHz	no	
DFSDM	PLL3Q	24.573874	MHz	yes	Digital micro
SAI1	PLL3Q	24.573874	MHz	no	
SAI2	PLL3Q	24.573874	MHz	yes	AudCodec 48kHz (use PLL3R
for 44.1kHz)					
SAI3	PLL3Q	24.573874	MHz	no	
SAI4	PLL3Q	24.573874	MHz	yes	SPDIF TX 48kHz (use PLL3R
for 44.1kHz)					
PLL3R	PLL3	11.290699	MHz	yes	
PLL4	HSE	594.000000	MHz	yes	
PLL4P	PLL4	99.000000	MHz	yes	
SDMMC1	PLL4P	99.000000	MHz	yes	µSD card
SDMMC2	PLL4P	99.000000	MHz	yes	eMMC
SDMMC3	PLL4P	99.000000	MHz	yes	
SPDIF	PLL4P	99.000000	MHz	yes	SPDIF RX
PLL4Q	PLL4	74.250000	MHz	yes	
LCD	PLL4Q	74.250000	MHz	yes	LTDC & DSI display pixel
clock					
PLL4R	PLL4	74.250000	MHz	yes	
FDCAN	PLL4R	74.250000	MHz	yes	Should be as high as
possible and < 100MHz					
STGEN	HSE	24.000000	MHz	yes	
USBPHYC	HSE	24.000000	MHz	yes	USB PHY Ctrl for USB Host
and OTG					
USBPLL	USBPHYC	48.000000	MHz	yes	
USB0	USBPLL	48.000000	MHz	yes	USB OTG
CSI	N.A.	4.000000	MHz	yes	Mandatory for IO
compensation					
ETH	N.A.	0.000000	MHz	no	ETH clocked by RGMII PHY

3.1.1.1 Device tree

Here are the corresponding device tree `rcc_clk` sub node properties consumed by the first stage boot loader (FSBL) to configure the clock tree above:

```
st,clksrc = <
    CLK_MPU_PLL1P
    CLK_AXI_PLL2P
    CLK_MCU_PLL3P
    CLK_PLL12_HSE
```




```

        CLK_PLL3_HSE
        CLK_PLL4_HSE
        CLK_RTC_LSE
        CLK_MC01_DISABLED
        CLK_MC02_DISABLED
    >;

    st,clkdiv = <
        1 /*MPU*/
        0 /*AXI*/
        0 /*MCU*/
        1 /*APB1*/
        1 /*APB2*/
        1 /*APB3*/
        1 /*APB4*/
        2 /*APB5*/
        23 /*RTC*/
        0 /*MC01*/
        0 /*MC02*/
    >;

    st,pkcs = <
        CLK_CKPER_HSE
        CLK_FMC_ACLK
        CLK_QSPI_ACLK
        CLK_ETH_DISABLED
        CLK_SDMMC12_PLL4P
        CLK_DSI_DSIPLL
        CLK_STGEN_HSE
        CLK_USBPHY_HSE
        CLK_SPI2S1_PLL3Q
        CLK_SPI2S23_PLL3Q
        CLK_SPI45_HSI
        CLK_SPI6_HSI
        CLK_I2C46_HSI
        CLK_SDMMC3_PLL4P
        CLK_USB0_USBPHY
        CLK_ADC_CKPER
        CLK_CEC_LSE
        CLK_I2C12_HSI
        CLK_I2C35_HSI
        CLK_UART1_HSI
        CLK_UART24_HSI
        CLK_UART35_HSI
        CLK_UART6_HSI
        CLK_UART78_HSI
        CLK_SPDIF_PLL4P
        CLK_FDCAN_PLL4R
        CLK_SAI1_PLL3Q
        CLK_SAI2_PLL3Q
        CLK_SAI3_PLL3Q
        CLK_SAI4_PLL3Q
        CLK_RNG1_LSI
        CLK_RNG2_LSI
        CLK_LPTIM1_PCLK1
        CLK_LPTIM23_PCLK3
        CLK_LPTIM45_LSE
    >;

    /* VCO = 1066.0 MHz => P = 266 (AXI), Q = 533 (GPU), R = 533 (DDR) */
    pll2: st,pll@1 {
        cfg = < 2 65 1 0 0 PQR(1,1,1) >;
        frac = < 0x1400 >;
        u-boot,dm-pre-reloc;
    };

    /* VCO = 417.8 MHz => P = 209, Q = 25, R = 11 */

```



```

pll3: st,pll@2 {
    cfg = < 1 33 1 16 36 PQR(1,1,1) >;
    frac = < 0x1a04 >;
    u-boot,dm-pre-reloc;
};

/* VCO = 594.0 MHz => P = 99, Q = 74, R = 74 */
pll4: st,pll@3 {
    cfg = < 3 98 5 7 7 PQR(1,1,1) >;
    u-boot,dm-pre-reloc;
};
    
```

3.2 STM32MP157x-DK2 case

This chapter shows the boot time clock tree set by the FSBL on STM32MP157x-DK2 DISCO board. Linux eventual runtime modifications are not covered here.

3.2.1 Clock tree

The following table shows what STM32MP157x-DK2 clock tree looks like, as a result of the boot chain execution with the device tree built with STM32CubeMX.

Component Comment	Parent	Frequency	Used?	
LSI	N.A.	0.032000 MHz	yes	Mandatory for IWDG, DAC
DAC	LSI	0.032000 MHz	no	
RNG1	LSI	0.032000 MHz	yes	Input frequency should be as low as possible
RNG2	LSI	0.032000 MHz	yes	Input frequency should be as low as possible
IWDG1	LSI	0.032000 MHz	yes	
IWDG2	LSI	0.032000 MHz	yes	
LSE	N.A.	0.032768 MHz	yes	Mandatory for DTS
RTC	LSE	0.032768 MHz	yes	
TAMP	LSE	0.032768 MHz	yes	
CEC	LSE	0.032768 MHz	yes	
LPTIM4	LSE	0.032768 MHz	yes	
LPTIM5	LSE	0.032768 MHz	yes	
HSI	N.A.	64.000000 MHz	yes	
SPI4	HSI	64.000000 MHz	yes	Arduino
SPI5	HSI	64.000000 MHz	yes	Rpi
SPI6	HSI	64.000000 MHz	no	
I2C4	HSI	64.000000 MHz	yes	PMIC
I2C6	HSI	64.000000 MHz	no	
I2C1	HSI	64.000000 MHz	yes	Rpi and peripherals
I2C2	HSI	64.000000 MHz	no	
I2C3	HSI	64.000000 MHz	no	
I2C5	HSI	64.000000 MHz	yes	Rpi and Arduino
USART1	HSI	64.000000 MHz	no	
USART2	HSI	64.000000 MHz	yes	Bluetooth
USART3	HSI	64.000000 MHz	yes	Rpi
UART4	HSI	64.000000 MHz	yes	Linux console
UART5	HSI	64.000000 MHz	no	
USART6	HSI	64.000000 MHz	no	
UART7	HSI	64.000000 MHz	yes	Arduino
UART8	HSI	64.000000 MHz	no	
MCO1	HSI	64.000000 MHz	no	Available so can be used
HSE	N.A.	24.000000 MHz	yes	
DSIPLL	HSE	125.000000 MHz	no	DSI DPHY PLL
DSIBL	DSIPLL	125.000000 MHz	no	DSI lanebyte clock



RTCDIV	HSE	1.000000 MHz	yes	Only used when RTC source
is HSE				
ck_per	HSE	24.000000 MHz	yes	
ADC	ck_per	24.000000 MHz	yes	Can use internal divider to
be < 40MHz				
PLL1	HSE	xxx MHz	yes	
PLL1P	PLL1	xxx MHz	yes	
MPUDIV	PLL1P	xxx MHz	yes	
Cortex-A7	PLL1P	xxx MHz	yes	650 MHz or 800 MHz, see the
information box above				
MC02	Cortex-A7	xxx MHz	no	Available so can be used
PLL2	HSE	1066.000000 MHz	yes	
PLL2P	PLL2	266.500000 MHz	yes	
AXI	PLL2P	266.500000 MHz	yes	< 266MHz
FMC	AXI	266.500000 MHz	no	
QSPI	AXI	266.500000 MHz	no	
SYSRAM	AXI	266.500000 MHz	yes	
ROM	AXI	266.500000 MHz	yes	
AHB5	AXI	266.500000 MHz	yes	< 266MHz
CRYP1	AHB5	266.500000 MHz	yes	
HASH1	AHB5	266.500000 MHz	yes	
GPI0Z	AHB5	266.500000 MHz	yes	
BKPSRAM	AHB5	266.500000 MHz	yes	
AHB6	AXI	266.500000 MHz	yes	< 266MHz
CRC1	AHB6	266.500000 MHz	yes	
MDMA	AHB6	266.500000 MHz	yes	
USBH	AHB6	266.500000 MHz	yes	USB Host
APB4	AHB6	133.250000 MHz	yes	
APB5	AHB6	66.625000 MHz	yes	
BSEC	APB5	66.625000 MHz	yes	< 67MHz
ETZPC	APB5	66.625000 MHz	yes	
TZC	APB5	66.625000 MHz	yes	
DBGAPB	AXI	133.250000 MHz	yes	JTAG & Coresight
DBGMCU	DBGAPB	66.625000 MHz	yes	
STM	DBGAPB	66.625000 MHz	no	
PLL2Q	PLL2	533.000000 MHz	yes	
GPU	PLL2Q	533.000000 MHz	yes	< 533MHz
PLL2R	PLL2	533.000000 MHz	yes	
DDR	PLL2R	533.000000 MHz	yes	< 533MHz
PLL3	HSE	417.755859 MHz	yes	
PLL3P	PLL3	208.877930 MHz	yes	
MLAHB	PLL3P	208.877930 MHz	yes	< 209MHz
Cortex-M4	MLAHB	208.877930 MHz	yes	
SRAM1	MLAHB	208.877930 MHz	yes	
SRAM2	MLAHB	208.877930 MHz	yes	
SRAM3	MLAHB	208.877930 MHz	yes	
RETRAM	MLAHB	208.877930 MHz	yes	
AHB1	MLAHB	208.877930 MHz	yes	< 209MHz
AHB2	MLAHB	208.877930 MHz	yes	< 209MHz
DMA1	AHB2	208.877930 MHz	yes	
DMA2	AHB2	208.877930 MHz	yes	
DMAMUX	AHB2	208.877930 MHz	yes	
APB1	MLAHB	104.438965 MHz	yes	
LPTIM1	APB1	104.438965 MHz	yes	
WWDG	APB1	104.438965 MHz	yes	
APB2	MLAHB	104.438965 MHz	yes	
TIM2	MLAHB	208.877930 MHz	yes	TIM Group 1
TIM3	MLAHB	208.877930 MHz	yes	TIM Group 1
TIM4	MLAHB	208.877930 MHz	yes	TIM Group 1
TIM5	MLAHB	208.877930 MHz	yes	TIM Group 1
TIM6	MLAHB	208.877930 MHz	yes	TIM Group 1
TIM7	MLAHB	208.877930 MHz	yes	TIM Group 1
TIM12	MLAHB	208.877930 MHz	yes	TIM Group 1
TIM13	MLAHB	208.877930 MHz	yes	TIM Group 1
TIM14	MLAHB	208.877930 MHz	yes	TIM Group 1
TIM1	MLAHB	208.877930 MHz	yes	TIM Group 2
TIM8	MLAHB	208.877930 MHz	yes	TIM Group 2



TIM15	MLAHB	208.877930	MHz	yes	TIM Group 2
TIM16	MLAHB	208.877930	MHz	yes	TIM Group 2
TIM17	MLAHB	208.877930	MHz	yes	TIM Group 2
APB3	MLAHB	104.438965	MHz	yes	
LPTIM2	APB3	104.438965	MHz	yes	
LPTIM3	APB3	104.438965	MHz	yes	
SYSCFG	APB3	104.438965	MHz	yes	
VREFBUF	APB3	104.438965	MHz	yes	
DTS	APB3	104.438965	MHz	yes	
HDP	APB3	104.438965	MHz	no	
AHB3	MLAHB	208.877930	MHz	yes	< 209MHz
CRC2	AHB3	208.877930	MHz	yes	
CRYP2	AHB3	208.877930	MHz	yes	
HASH2	AHB3	208.877930	MHz	yes	
DCMI	AHB3	208.877930	MHz	no	
IPCC	AHB3	208.877930	MHz	yes	Mailbox
AHB4	MLAHB	208.877930	MHz	yes	< 209MHz
PWR	AHB4	208.877930	MHz	yes	
RCC	AHB4	208.877930	MHz	yes	
GPIOA-K	AHB4	208.877930	MHz	yes	
EXTI	AHB4	208.877930	MHz	yes	
PLL3Q	PLL3	24.573874	MHz	yes	
SPI1	PLL3Q	24.573874	MHz	yes	SPI2S1 for BT PCM
SPI2	PLL3Q	24.573874	MHz	yes	SPI2S2 for HDMI
SPI3	PLL3Q	24.573874	MHz	no	
DFSDM	PLL3Q	24.573874	MHz	no	
SAI1	PLL3Q	24.573874	MHz	no	
SAI2	PLL3Q	24.573874	MHz	yes	AudCodec 48kHz (use PLL3R)
for 44.1kHz)					
SAI3	PLL3Q	24.573874	MHz	no	
SAI4	PLL3Q	24.573874	MHz	no	
PLL3R	PLL3	11.290699	MHz	yes	
PLL4	HSE	594.000000	MHz	yes	
PLL4P	PLL4	99.000000	MHz	yes	
SDMMC1	PLL4P	99.000000	MHz	yes	µSD card
SDMMC2	PLL4P	99.000000	MHz	yes	Wifi
SDMMC3	PLL4P	99.000000	MHz	yes	Rpi
SPDIF	PLL4P	99.000000	MHz	no	
PLL4Q	PLL4	74.250000	MHz	yes	
LCD	PLL4Q	74.250000	MHz	yes	LTDC & DSI display pixel
clock					
PLL4R	PLL4	74.250000	MHz	yes	
FDCAN	PLL4R	74.250000	MHz	no	
STGEN	HSE	24.000000	MHz	yes	
USBPHYC	HSE	24.000000	MHz	yes	USB PHY Ctrl for USB Host
and OTG					
USBPLL	USBPHYC	48.000000	MHz	yes	
USB0	USBPLL	48.000000	MHz	yes	USB OTG
CSI	N.A.	4.000000	MHz	yes	Mandatory for IO
compensation					
ETH	N.A.	0.000000	MHz	no	ETH clocked by RGMII PHY

3.2.2 Device tree

Here are the corresponding device tree `rcc_clk` sub node properties consumed by the first stage boot loader (FSBL) to configure the clock tree above:

```
st,clksrc = <
    CLK_MPU_PLL1P
    CLK_AXI_PLL2P
    CLK_MCU_PLL3P
    CLK_PLL12_HSE
```



```

        CLK_PLL3_HSE
        CLK_PLL4_HSE
        CLK_RTC_LSE
        CLK_MC01_DISABLED
        CLK_MC02_DISABLED
    >;

    st,clkdiv = <
        1 /*MPU*/
        0 /*AXI*/
        0 /*MCU*/
        1 /*APB1*/
        1 /*APB2*/
        1 /*APB3*/
        1 /*APB4*/
        2 /*APB5*/
        23 /*RTC*/
        0 /*MC01*/
        0 /*MC02*/
    >;

    st,pkcs = <
        CLK_CKPER_HSE
        CLK_FMC_ACLK
        CLK_QSPI_ACLK
        CLK_ETH_DISABLED
        CLK_SDMMC12_PLL4P
        CLK_DSI_DSIPLL
        CLK_STGEN_HSE
        CLK_USBPHY_HSE
        CLK_SPI2S1_PLL3Q
        CLK_SPI2S23_PLL3Q
        CLK_SPI45_HSI
        CLK_SPI6_HSI
        CLK_I2C46_HSI
        CLK_SDMMC3_PLL4P
        CLK_USB0_USBPHY
        CLK_ADC_CKPER
        CLK_CEC_LSE
        CLK_I2C12_HSI
        CLK_I2C35_HSI
        CLK_UART1_HSI
        CLK_UART24_HSI
        CLK_UART35_HSI
        CLK_UART6_HSI
        CLK_UART78_HSI
        CLK_SPDIF_PLL4P
        CLK_FDCAN_PLL4R
        CLK_SAI1_PLL3Q
        CLK_SAI2_PLL3Q
        CLK_SAI3_PLL3Q
        CLK_SAI4_PLL3Q
        CLK_RNG1_LSI
        CLK_RNG2_LSI
        CLK_LPTIM1_PCLK1
        CLK_LPTIM23_PCLK3
        CLK_LPTIM45_LSE
    >;

    /* VCO = 1066.0 MHz => P = 266 (AXI), Q = 533 (GPU), R = 533 (DDR) */
    pll2: st,pll@1 {
        cfg = < 2 65 1 0 0 PQR(1,1,1) >;
        frac = < 0x1400 >;
        u-boot,dm-pre-reloc;
    };

    /* VCO = 417.8 MHz => P = 209, Q = 25, R = 11 */

```



```

pll3: st,pll@2 {
    cfg = < 1 33 1 16 36 PQR(1,1,1) >;
    frac = < 0x1a04 >;
    u-boot,dm-pre-reloc;
};

/* VCO = 594.0 MHz => P = 99, Q = 74, R = 74 */
pll4: st,pll@3 {
    cfg = < 3 98 5 7 7 PQR(1,1,1) >;
    u-boot,dm-pre-reloc;
};

```

Reset and Clock Control

Low Speed Internal oscillator (STM32 clock source)

Low Speed External oscillator (STM32 clock source)

High Speed Internal oscillator (STM32 clock source) or High Speed Synchronous Serial Interface (MIPI® Alliance standard)

High Speed External oscillator (STM32 clock source)

Multi Speed Internal oscillator (STM32 clock source)

Cortex®

First Stage Boot Loader

Linux® is a registered trademark of Linus Torvalds.

Das U-Boot -- the Universal Boot Loader (see [U-Boot_overview](#))

Secondary Program Loader, *Also known as **U-Boot SPL***

Read Only Memory

Central processing unit

Dynamic voltage and frequency scaling

Independent Watchdog

Digital-to-analog converter (Electronic circuit that converts a binary number into a continuously varying value.)

Device Tree Source (in software context) or Digital Temperature Sensor (in peripheral context)

Real Time Clock

Tamper

Consumer Electronics Control (HDMI standard)

Power Management Integrated Circuit

Display Serial Interface (MIPI® Alliance standard)

Analog-to-digital converter. The process of converting a sampled analog signal to a digital code that represents the amplitude of the original signal sample.

USB Host (STM32 specific)

Boot and Security and OTP control

Extended TrustZone Protection Controller



TrustZone® address space Controller for DDR

Arm® and TrustZone® are registered trademarks of Arm Limited (or its subsidiaries) in the US and/or elsewhere.

debug and test protocol, named from the Joint Test Action Group that developed it

System Trace Module

Graphics Processing Units

Doubledata rate (memory domain)

System Configuration

voltage reference buffer (STM32 specific)

Hardware Debug Port

Digital Camera Memory Interface

Inter-Processor Communication Controller

External Interrupt

Digital Filter for Sigma-Delta Modulator

Transmit

former spelling for e•MMC ('e' in italic)

Receive

LCD TFT Display Controller (STM32 specific)

System Time Generator

USB On-The-Go (Capability/type of USB port, acting primarily as USB device, to also act as USB host. Also known as USB OTG.)

input/output

Ethernet

Microprocessor Unit

Microcontroller Unit (MCUs have internal flash memory and are intended to operate with a minimum amount of external support ICs. They commonly are a self-contained, system-on-chip (SoC) designs.)

Discovery kit

BlueTooth

High-Definition Multimedia Interface (HDMI standard)

Stable: 12.03.2021 - 11:29 / Revision: 12.03.2021 - 11:15

All the peripherals receive one or several clocks that are generated via [RCC internal peripheral](#). RCC relies on several clocks sources (LSI, LSE, HSI, HSE, CSI) and four PLL in order to provide adequate input frequencies to all the peripherals. The clock tree covers all the system clock distribution aspects, from the clock sources to the consumer peripherals (internal and external), except clock gating management that is locally controlled by each peripheral driver.

Contents

1 Overview	17
2 How to build a clock tree?	18



3 ST boards clock tree	19
3.1 STM32MP157x-EV1 case	19
3.1.1 Clock tree	19
3.1.1.1 Device tree	21
3.2 STM32MP157x-DK2 case	23
3.2.1 Clock tree	23
3.2.2 Device tree	25



1 Overview

The clock tree is managed via RCC internal peripheral hardware block and it is configured at different steps from the Cortex-A7:

- When the device is **reset**, all RCC registers take their reset values: the four PLL are disabled and most of the clock source selectors are pointing to the HSI.
- The ROM Code configures the minimum clock tree needed to boot on the selected boot device.
- The **FSBL** (Cf. [Boot chain overview](#)) completely configures the clock tree as expected, all the way up to Linux, thanks to the configuration given in the [device tree](#).
 - U-Boot SPL (basic boot chain) binding is available in [doc/device-tree-bindings/clock/st,stm32mp1.txt](#)
 - TF-A (trusted boot chain) binding follows the same syntax as U-Boot SPL
- **Linux** is responsible for the clock tree at runtime and it may partly modify it thanks to information taken from the [device tree](#):
 - Linux binding is available in [Documentation/devicetree/bindings/clock/clock-bindings.txt](#) (and surrounding files): 'fixed-clock' compatible, 'clocks' and 'assigned-clocks' properties are important concepts to understand the management of clocks providers/consumers.

Notice that [STM32Cube](#) (running on the Cortex-M4) does not control the clock tree setup so the configuration applied by the Cortex-A7 must provide suitable clocks to Cortex-M4, like it is explained in [resource manager](#) description. One exception to this rule is the [engineering boot](#), allowing to directly load (and debug) the Cortex-M4 whereas the Cortex-A7 execution is stalled in the ROM code.



2 How to build a clock tree?

Building a clock tree is quite complex as it needs to take into account the constraints set by each internal and external peripheral, including external clock sources.

We encourage the use of [STM32CubeMX](#) to build the clock tree, and avoid having to know all internal peripherals details: the tool allows to select the peripherals that will be present on the board, fix the clock sources frequencies and automatically find an optimized clock tree. It is then able to generate the device tree that is directly consumed by the boot chain and Linux.



3 ST boards clock tree

 See [How to change the CPU frequency](#) article for more information about the CPU frequency setting (PLL1P), including dynamic voltage and frequency scaling (DVFS) configuration.

3.1 STM32MP157x-EV1 case

This chapter shows the boot time clock tree set by the FSBL on STM32MP157x-EV1 evaluation board. Linux eventual runtime modifications are not covered here.

3.1.1 Clock tree

The following table shows what STM32MP157x-EV1 clock tree looks like, as a result of the boot chain execution with the device tree built with STM32CubeMX.

Component Comment	Parent	Frequency	Used?	
LSI	N.A.	0.032000 MHz	yes	Mandatory for IWDG, DAC
DAC	LSI	0.032000 MHz	yes	
RNG1	LSI	0.032000 MHz	yes	Input frequency should be
as low as possible				
RNG2	LSI	0.032000 MHz	yes	Input frequency should be
as low as possible				
IWDG1	LSI	0.032000 MHz	yes	
IWDG2	LSI	0.032000 MHz	yes	
LSE	N.A.	0.032768 MHz	yes	Mandatory for DTS
RTC	LSE	0.032768 MHz	yes	
TAMP	LSE	0.032768 MHz	yes	
CEC	LSE	0.032768 MHz	yes	
LPTIM4	LSE	0.032768 MHz	yes	
LPTIM5	LSE	0.032768 MHz	yes	
HSI	N.A.	64.000000 MHz	yes	
SPI4	HSI	64.000000 MHz	no	
SPI5	HSI	64.000000 MHz	no	
SPI6	HSI	64.000000 MHz	yes	
I2C4	HSI	64.000000 MHz	yes	PMIC
I2C6	HSI	64.000000 MHz	no	
I2C1	HSI	64.000000 MHz	no	
I2C2	HSI	64.000000 MHz	yes	Rpi and peripherals
I2C3	HSI	64.000000 MHz	no	
I2C5	HSI	64.000000 MHz	yes	Rpi
USART1	HSI	64.000000 MHz	yes	
USART2	HSI	64.000000 MHz	no	
USART3	HSI	64.000000 MHz	yes	Rpi
UART4	HSI	64.000000 MHz	yes	Linux console
UART5	HSI	64.000000 MHz	no	
UART6	HSI	64.000000 MHz	no	
UART7	HSI	64.000000 MHz	no	
UART8	HSI	64.000000 MHz	no	
MC01	HSI	64.000000 MHz	no	Available so can be used
HSE	N.A.	24.000000 MHz	yes	
DSIPLL	HSE	125.000000 MHz	no	DSI DPHY PLL
DSIBL	DSIPLL	125.000000 MHz	no	DSI lanebyte clock
RTCDIV	HSE	1.000000 MHz	yes	Only used when RTC source
is HSE				



ck_per	HSE	24.000000	MHz	yes	
ADC	ck_per	24.000000	MHz	yes	Can use internal divider to
be < 40MHz					
PLL1	HSE	xxx	MHz	yes	
PLL1P	PLL1	xxx	MHz	yes	
MPUDIV	PLL1P	xxx	MHz	yes	
Cortex-A7	PLL1P	xxx	MHz	yes	650 MHz or 800 MHz, see the
information box above					
MC02	Cortex-A7	xxx	MHz	no	Available so can be used
PLL2	HSE	1066.000000	MHz	yes	
PLL2P	PLL2	266.500000	MHz	yes	
AXI	PLL2P	266.500000	MHz	yes	< 266MHz
FMC	AXI	266.500000	MHz	yes	NAND flash
QSPI	AXI	266.500000	MHz	yes	NOR flash
SYSRAM	AXI	266.500000	MHz	yes	
ROM	AXI	266.500000	MHz	yes	
AHB5	AXI	266.500000	MHz	yes	< 266MHz
CRYP1	AHB5	266.500000	MHz	yes	
HASH1	AHB5	266.500000	MHz	yes	
GPI0Z	AHB5	266.500000	MHz	yes	
BKPSRAM	AHB5	266.500000	MHz	yes	
AHB6	AXI	266.500000	MHz	yes	< 266MHz
CRC1	AHB6	266.500000	MHz	yes	
MDMA	AHB6	266.500000	MHz	yes	
USBH	AHB6	266.500000	MHz	yes	USB Host
APB4	AHB6	133.250000	MHz	yes	
APB5	AHB6	66.625000	MHz	yes	
BSEC	APB5	66.625000	MHz	yes	< 67MHz
ETZPC	APB5	66.625000	MHz	yes	
TZC	APB5	66.625000	MHz	yes	
DBGAPB	AXI	133.250000	MHz	yes	JTAG & Coresight
DBGMCU	DBGAPB	66.625000	MHz	yes	
STM	DBGAPB	66.625000	MHz	yes	
PLL2Q	PLL2	533.000000	MHz	yes	
GPU	PLL2Q	533.000000	MHz	yes	< 533MHz
PLL2R	PLL2	533.000000	MHz	yes	
DDR	PLL2R	533.000000	MHz	yes	< 533MHz
PLL3	HSE	417.755859	MHz	yes	
PLL3P	PLL3	208.877930	MHz	yes	
MLAHB	PLL3P	208.877930	MHz	yes	< 209MHz
Cortex-M4	MLAHB	208.877930	MHz	yes	
SRAM1	MLAHB	208.877930	MHz	yes	
SRAM2	MLAHB	208.877930	MHz	yes	
SRAM3	MLAHB	208.877930	MHz	yes	
RETRAM	MLAHB	208.877930	MHz	yes	
AHB1	MLAHB	208.877930	MHz	yes	< 209MHz
AHB2	MLAHB	208.877930	MHz	yes	< 209MHz
DMA1	AHB2	208.877930	MHz	yes	
DMA2	AHB2	208.877930	MHz	yes	
DMAMUX	AHB2	208.877930	MHz	yes	
APB1	MLAHB	104.438965	MHz	yes	
LPTIM1	APB1	104.438965	MHz	yes	
WWDG	APB1	104.438965	MHz	yes	
APB2	MLAHB	104.438965	MHz	yes	
TIM2	MLAHB	208.877930	MHz	yes	TIM Group 1
TIM3	MLAHB	208.877930	MHz	yes	TIM Group 1
TIM4	MLAHB	208.877930	MHz	yes	TIM Group 1
TIM5	MLAHB	208.877930	MHz	yes	TIM Group 1
TIM6	MLAHB	208.877930	MHz	yes	TIM Group 1
TIM7	MLAHB	208.877930	MHz	yes	TIM Group 1
TIM12	MLAHB	208.877930	MHz	yes	TIM Group 1
TIM13	MLAHB	208.877930	MHz	yes	TIM Group 1
TIM14	MLAHB	208.877930	MHz	yes	TIM Group 1
TIM1	MLAHB	208.877930	MHz	yes	TIM Group 2
TIM8	MLAHB	208.877930	MHz	yes	TIM Group 2
TIM15	MLAHB	208.877930	MHz	yes	TIM Group 2
TIM16	MLAHB	208.877930	MHz	yes	TIM Group 2



TIM17	MLAHB	208.877930	MHz	yes	TIM Group 2
APB3	MLAHB	104.438965	MHz	yes	
LPTIM2	APB3	104.438965	MHz	yes	
LPTIM3	APB3	104.438965	MHz	yes	
SYSCFG	APB3	104.438965	MHz	yes	
VREFBUF	APB3	104.438965	MHz	yes	
DTS	APB3	104.438965	MHz	yes	
HDP	APB3	104.438965	MHz	yes	
AHB3	MLAHB	208.877930	MHz	yes	< 209MHz
CRC2	AHB3	208.877930	MHz	yes	
CRYP2	AHB3	208.877930	MHz	yes	
HASH2	AHB3	208.877930	MHz	yes	
DCMI	AHB3	208.877930	MHz	yes	Camera
IPCC	AHB3	208.877930	MHz	yes	Mailbox
AHB4	MLAHB	208.877930	MHz	yes	< 209MHz
PWR	AHB4	208.877930	MHz	yes	
RCC	AHB4	208.877930	MHz	yes	
GPIOA-K	AHB4	208.877930	MHz	yes	
EXTI	AHB4	208.877930	MHz	yes	
PLL3Q	PLL3	24.573874	MHz	yes	
SPI1	PLL3Q	24.573874	MHz	no	
SPI2	PLL3Q	24.573874	MHz	no	
SPI3	PLL3Q	24.573874	MHz	no	
DFSDM	PLL3Q	24.573874	MHz	yes	Digital micro
SAI1	PLL3Q	24.573874	MHz	no	
SAI2	PLL3Q	24.573874	MHz	yes	AudCodec 48kHz (use PLL3R
for 44.1kHz)					
SAI3	PLL3Q	24.573874	MHz	no	
SAI4	PLL3Q	24.573874	MHz	yes	SPDIF TX 48kHz (use PLL3R
for 44.1kHz)					
PLL3R	PLL3	11.290699	MHz	yes	
PLL4	HSE	594.000000	MHz	yes	
PLL4P	PLL4	99.000000	MHz	yes	
SDMMC1	PLL4P	99.000000	MHz	yes	µSD card
SDMMC2	PLL4P	99.000000	MHz	yes	eMMC
SDMMC3	PLL4P	99.000000	MHz	yes	
SPDIF	PLL4P	99.000000	MHz	yes	SPDIF RX
PLL4Q	PLL4	74.250000	MHz	yes	
LCD	PLL4Q	74.250000	MHz	yes	LTDC & DSI display pixel
clock					
PLL4R	PLL4	74.250000	MHz	yes	
FDCAN	PLL4R	74.250000	MHz	yes	Should be as high as
possible and < 100MHz					
STGEN	HSE	24.000000	MHz	yes	
USBPHYC	HSE	24.000000	MHz	yes	USB PHY Ctrl for USB Host
and OTG					
USBPLL	USBPHYC	48.000000	MHz	yes	
USB0	USBPLL	48.000000	MHz	yes	USB OTG
CSI	N.A.	4.000000	MHz	yes	Mandatory for IO
compensation					
ETH	N.A.	0.000000	MHz	no	ETH clocked by RGMII PHY

3.1.1.1 Device tree

Here are the corresponding device tree `rcc_clk` sub node properties consumed by the first stage boot loader (FSBL) to configure the clock tree above:

```

st,clksrc = <
    CLK_MPU_PLL1P
    CLK_AXI_PLL2P
    CLK_MCU_PLL3P
    CLK_PLL12_HSE
    
```



```

        CLK_PLL3_HSE
        CLK_PLL4_HSE
        CLK_RTC_LSE
        CLK_MC01_DISABLED
        CLK_MC02_DISABLED
    >;

    st,clkdiv = <
        1 /*MPU*/
        0 /*AXI*/
        0 /*MCU*/
        1 /*APB1*/
        1 /*APB2*/
        1 /*APB3*/
        1 /*APB4*/
        2 /*APB5*/
        23 /*RTC*/
        0 /*MC01*/
        0 /*MC02*/
    >;

    st,pkcs = <
        CLK_CKPER_HSE
        CLK_FMC_ACLK
        CLK_QSPI_ACLK
        CLK_ETH_DISABLED
        CLK_SDMMC12_PLL4P
        CLK_DSI_DSIPLL
        CLK_STGEN_HSE
        CLK_USBPHY_HSE
        CLK_SPI2S1_PLL3Q
        CLK_SPI2S23_PLL3Q
        CLK_SPI45_HSI
        CLK_SPI6_HSI
        CLK_I2C46_HSI
        CLK_SDMMC3_PLL4P
        CLK_USB0_USBPHY
        CLK_ADC_CKPER
        CLK_CEC_LSE
        CLK_I2C12_HSI
        CLK_I2C35_HSI
        CLK_UART1_HSI
        CLK_UART24_HSI
        CLK_UART35_HSI
        CLK_UART6_HSI
        CLK_UART78_HSI
        CLK_SPDIF_PLL4P
        CLK_FDCAN_PLL4R
        CLK_SAI1_PLL3Q
        CLK_SAI2_PLL3Q
        CLK_SAI3_PLL3Q
        CLK_SAI4_PLL3Q
        CLK_RNG1_LSI
        CLK_RNG2_LSI
        CLK_LPTIM1_PCLK1
        CLK_LPTIM23_PCLK3
        CLK_LPTIM45_LSE
    >;

    /* VCO = 1066.0 MHz => P = 266 (AXI), Q = 533 (GPU), R = 533 (DDR) */
    pll2: st,pll@1 {
        cfg = < 2 65 1 0 0 PQR(1,1,1) >;
        frac = < 0x1400 >;
        u-boot,dm-pre-reloc;
    };

    /* VCO = 417.8 MHz => P = 209, Q = 25, R = 11 */

```



```

pll3: st,pll@2 {
    cfg = < 1 33 1 16 36 PQR(1,1,1) >;
    frac = < 0x1a04 >;
    u-boot,dm-pre-reloc;
};

/* VCO = 594.0 MHz => P = 99, Q = 74, R = 74 */
pll4: st,pll@3 {
    cfg = < 3 98 5 7 7 PQR(1,1,1) >;
    u-boot,dm-pre-reloc;
};
    
```

3.2 STM32MP157x-DK2 case

This chapter shows the boot time clock tree set by the FSBL on STM32MP157x-DK2 DISCO board. Linux eventual runtime modifications are not covered here.

3.2.1 Clock tree

The following table shows what STM32MP157x-DK2 clock tree looks like, as a result of the boot chain execution with the device tree built with STM32CubeMX.

Component Comment	Parent	Frequency	Used?	
LSI	N.A.	0.032000 MHz	yes	Mandatory for IWDG, DAC
DAC	LSI	0.032000 MHz	no	
RNG1	LSI	0.032000 MHz	yes	Input frequency should be as low as possible
RNG2	LSI	0.032000 MHz	yes	Input frequency should be as low as possible
IWDG1	LSI	0.032000 MHz	yes	
IWDG2	LSI	0.032000 MHz	yes	
LSE	N.A.	0.032768 MHz	yes	Mandatory for DTS
RTC	LSE	0.032768 MHz	yes	
TAMP	LSE	0.032768 MHz	yes	
CEC	LSE	0.032768 MHz	yes	
LPTIM4	LSE	0.032768 MHz	yes	
LPTIM5	LSE	0.032768 MHz	yes	
HSI	N.A.	64.000000 MHz	yes	
SPI4	HSI	64.000000 MHz	yes	Arduino
SPI5	HSI	64.000000 MHz	yes	Rpi
SPI6	HSI	64.000000 MHz	no	
I2C4	HSI	64.000000 MHz	yes	PMIC
I2C6	HSI	64.000000 MHz	no	
I2C1	HSI	64.000000 MHz	yes	Rpi and peripherals
I2C2	HSI	64.000000 MHz	no	
I2C3	HSI	64.000000 MHz	no	
I2C5	HSI	64.000000 MHz	yes	Rpi and Arduino
USART1	HSI	64.000000 MHz	no	
USART2	HSI	64.000000 MHz	yes	Bluetooth
USART3	HSI	64.000000 MHz	yes	Rpi
UART4	HSI	64.000000 MHz	yes	Linux console
UART5	HSI	64.000000 MHz	no	
USART6	HSI	64.000000 MHz	no	
UART7	HSI	64.000000 MHz	yes	Arduino
UART8	HSI	64.000000 MHz	no	
MCO1	HSI	64.000000 MHz	no	Available so can be used
HSE	N.A.	24.000000 MHz	yes	
DSIPLL	HSE	125.000000 MHz	no	DSI DPHY PLL
DSIBL	DSIPLL	125.000000 MHz	no	DSI lanebyte clock



RTCDIV	HSE	1.000000 MHz	yes	Only used when RTC source
is HSE				
ck_per	HSE	24.000000 MHz	yes	
ADC	ck_per	24.000000 MHz	yes	Can use internal divider to
be < 40MHz				
PLL1	HSE	xxx MHz	yes	
PLL1P	PLL1	xxx MHz	yes	
MPUDIV	PLL1P	xxx MHz	yes	
Cortex-A7	PLL1P	xxx MHz	yes	650 MHz or 800 MHz, see the
information box above				
MC02	Cortex-A7	xxx MHz	no	Available so can be used
PLL2	HSE	1066.000000 MHz	yes	
PLL2P	PLL2	266.500000 MHz	yes	
AXI	PLL2P	266.500000 MHz	yes	< 266MHz
FMC	AXI	266.500000 MHz	no	
QSPI	AXI	266.500000 MHz	no	
SYSRAM	AXI	266.500000 MHz	yes	
ROM	AXI	266.500000 MHz	yes	
AHB5	AXI	266.500000 MHz	yes	< 266MHz
Cryp1	AHB5	266.500000 MHz	yes	
HASH1	AHB5	266.500000 MHz	yes	
GPIOZ	AHB5	266.500000 MHz	yes	
BKPSRAM	AHB5	266.500000 MHz	yes	
AHB6	AXI	266.500000 MHz	yes	< 266MHz
CRC1	AHB6	266.500000 MHz	yes	
MDMA	AHB6	266.500000 MHz	yes	
USBH	AHB6	266.500000 MHz	yes	USB Host
APB4	AHB6	133.250000 MHz	yes	
APB5	AHB6	66.625000 MHz	yes	
BSEC	APB5	66.625000 MHz	yes	< 67MHz
ETZPC	APB5	66.625000 MHz	yes	
TZC	APB5	66.625000 MHz	yes	
DBGAPB	AXI	133.250000 MHz	yes	JTAG & Coresight
DBGMCU	DBGAPB	66.625000 MHz	yes	
STM	DBGAPB	66.625000 MHz	no	
PLL2Q	PLL2	533.000000 MHz	yes	
GPU	PLL2Q	533.000000 MHz	yes	< 533MHz
PLL2R	PLL2	533.000000 MHz	yes	
DDR	PLL2R	533.000000 MHz	yes	< 533MHz
PLL3	HSE	417.755859 MHz	yes	
PLL3P	PLL3	208.877930 MHz	yes	
MLAHB	PLL3P	208.877930 MHz	yes	< 209MHz
Cortex-M4	MLAHB	208.877930 MHz	yes	
SRAM1	MLAHB	208.877930 MHz	yes	
SRAM2	MLAHB	208.877930 MHz	yes	
SRAM3	MLAHB	208.877930 MHz	yes	
RETRAM	MLAHB	208.877930 MHz	yes	
AHB1	MLAHB	208.877930 MHz	yes	< 209MHz
AHB2	MLAHB	208.877930 MHz	yes	< 209MHz
DMA1	AHB2	208.877930 MHz	yes	
DMA2	AHB2	208.877930 MHz	yes	
DMAMUX	AHB2	208.877930 MHz	yes	
APB1	MLAHB	104.438965 MHz	yes	
LPTIM1	APB1	104.438965 MHz	yes	
WWDG	APB1	104.438965 MHz	yes	
APB2	MLAHB	104.438965 MHz	yes	
TIM2	MLAHB	208.877930 MHz	yes	TIM Group 1
TIM3	MLAHB	208.877930 MHz	yes	TIM Group 1
TIM4	MLAHB	208.877930 MHz	yes	TIM Group 1
TIM5	MLAHB	208.877930 MHz	yes	TIM Group 1
TIM6	MLAHB	208.877930 MHz	yes	TIM Group 1
TIM7	MLAHB	208.877930 MHz	yes	TIM Group 1
TIM12	MLAHB	208.877930 MHz	yes	TIM Group 1
TIM13	MLAHB	208.877930 MHz	yes	TIM Group 1
TIM14	MLAHB	208.877930 MHz	yes	TIM Group 1
TIM1	MLAHB	208.877930 MHz	yes	TIM Group 2
TIM8	MLAHB	208.877930 MHz	yes	TIM Group 2



TIM15	MLAHB	208.877930	MHz	yes	TIM Group 2
TIM16	MLAHB	208.877930	MHz	yes	TIM Group 2
TIM17	MLAHB	208.877930	MHz	yes	TIM Group 2
APB3	MLAHB	104.438965	MHz	yes	
LPTIM2	APB3	104.438965	MHz	yes	
LPTIM3	APB3	104.438965	MHz	yes	
SYSCFG	APB3	104.438965	MHz	yes	
VREFBUF	APB3	104.438965	MHz	yes	
DTS	APB3	104.438965	MHz	yes	
HDP	APB3	104.438965	MHz	no	
AHB3	MLAHB	208.877930	MHz	yes	< 209MHz
CRC2	AHB3	208.877930	MHz	yes	
CRYP2	AHB3	208.877930	MHz	yes	
HASH2	AHB3	208.877930	MHz	yes	
DCMI	AHB3	208.877930	MHz	no	
IPCC	AHB3	208.877930	MHz	yes	Mailbox
AHB4	MLAHB	208.877930	MHz	yes	< 209MHz
PWR	AHB4	208.877930	MHz	yes	
RCC	AHB4	208.877930	MHz	yes	
GPIOA-K	AHB4	208.877930	MHz	yes	
EXTI	AHB4	208.877930	MHz	yes	
PLL3Q	PLL3	24.573874	MHz	yes	
SPI1	PLL3Q	24.573874	MHz	yes	SPI2S1 for BT PCM
SPI2	PLL3Q	24.573874	MHz	yes	SPI2S2 for HDMI
SPI3	PLL3Q	24.573874	MHz	no	
DFSDM	PLL3Q	24.573874	MHz	no	
SAI1	PLL3Q	24.573874	MHz	no	
SAI2	PLL3Q	24.573874	MHz	yes	AudCodec 48kHz (use PLL3R)
for 44.1kHz)					
SAI3	PLL3Q	24.573874	MHz	no	
SAI4	PLL3Q	24.573874	MHz	no	
PLL3R	PLL3	11.290699	MHz	yes	
PLL4	HSE	594.000000	MHz	yes	
PLL4P	PLL4	99.000000	MHz	yes	
SDMMC1	PLL4P	99.000000	MHz	yes	µSD card
SDMMC2	PLL4P	99.000000	MHz	yes	Wifi
SDMMC3	PLL4P	99.000000	MHz	yes	Rpi
SPDIF	PLL4P	99.000000	MHz	no	
PLL4Q	PLL4	74.250000	MHz	yes	
LCD	PLL4Q	74.250000	MHz	yes	LTDC & DSI display pixel
clock					
PLL4R	PLL4	74.250000	MHz	yes	
FDCAN	PLL4R	74.250000	MHz	no	
STGEN	HSE	24.000000	MHz	yes	
USBPHYC	HSE	24.000000	MHz	yes	USB PHY Ctrl for USB Host
and OTG					
USBPLL	USBPHYC	48.000000	MHz	yes	
USB0	USBPLL	48.000000	MHz	yes	USB OTG
CSI	N.A.	4.000000	MHz	yes	Mandatory for IO
compensation					
ETH	N.A.	0.000000	MHz	no	ETH clocked by RGMII PHY

3.2.2 Device tree

Here are the corresponding device tree `rcc_clk` sub node properties consumed by the first stage boot loader (FSBL) to configure the clock tree above:

```
st,clksrc = <
    CLK_MPU_PLL1P
    CLK_AXI_PLL2P
    CLK_MCU_PLL3P
    CLK_PLL12_HSE
```



```

    CLK_PLL3_HSE
    CLK_PLL4_HSE
    CLK_RTC_LSE
    CLK_MC01_DISABLED
    CLK_MC02_DISABLED
>;

st,clkdiv = <
    1 /*MPU*/
    0 /*AXI*/
    0 /*MCU*/
    1 /*APB1*/
    1 /*APB2*/
    1 /*APB3*/
    1 /*APB4*/
    2 /*APB5*/
    23 /*RTC*/
    0 /*MC01*/
    0 /*MC02*/
>;

st,pkcs = <
    CLK_CKPER_HSE
    CLK_FMC_ACLK
    CLK_QSPI_ACLK
    CLK_ETH_DISABLED
    CLK_SDMMC12_PLL4P
    CLK_DSI_DSIPLL
    CLK_STGEN_HSE
    CLK_USBPHY_HSE
    CLK_SPI2S1_PLL3Q
    CLK_SPI2S23_PLL3Q
    CLK_SPI45_HSI
    CLK_SPI6_HSI
    CLK_I2C46_HSI
    CLK_SDMMC3_PLL4P
    CLK_USB0_USBPHY
    CLK_ADC_CKPER
    CLK_CEC_LSE
    CLK_I2C12_HSI
    CLK_I2C35_HSI
    CLK_UART1_HSI
    CLK_UART24_HSI
    CLK_UART35_HSI
    CLK_UART6_HSI
    CLK_UART78_HSI
    CLK_SPDIF_PLL4P
    CLK_FDCAN_PLL4R
    CLK_SAI1_PLL3Q
    CLK_SAI2_PLL3Q
    CLK_SAI3_PLL3Q
    CLK_SAI4_PLL3Q
    CLK_RNG1_LSI
    CLK_RNG2_LSI
    CLK_LPTIM1_PCLK1
    CLK_LPTIM23_PCLK3
    CLK_LPTIM45_LSE
>;

/* VCO = 1066.0 MHz => P = 266 (AXI), Q = 533 (GPU), R = 533 (DDR) */
pll2: st,pll@1 {
    cfg = < 2 65 1 0 0 PQR(1,1,1) >;
    frac = < 0x1400 >;
    u-boot,dm-pre-reloc;
};

/* VCO = 417.8 MHz => P = 209, Q = 25, R = 11 */

```



```

pll3: st,pll@2 {
    cfg = < 1 33 1 16 36 PQR(1,1,1) >;
    frac = < 0x1a04 >;
    u-boot,dm-pre-reloc;
};

/* VCO = 594.0 MHz => P = 99, Q = 74, R = 74 */
pll4: st,pll@3 {
    cfg = < 3 98 5 7 7 PQR(1,1,1) >;
    u-boot,dm-pre-reloc;
};

```

Reset and Clock Control

Low Speed Internal oscillator (STM32 clock source)

Low Speed External oscillator (STM32 clock source)

High Speed Internal oscillator (STM32 clock source) or High Speed Synchronous Serial Interface (MIPI® Alliance standard)

High Speed External oscillator (STM32 clock source)

Multi Speed Internal oscillator (STM32 clock source)

Cortex®

First Stage Boot Loader

Linux® is a registered trademark of Linus Torvalds.

Das U-Boot -- the Universal Boot Loader (see [U-Boot_overview](#))

Secondary Program Loader, *Also known as **U-Boot SPL***

Read Only Memory

Central processing unit

Dynamic voltage and frequency scaling

Independent Watchdog

Digital-to-analog converter (Electronic circuit that converts a binary number into a continuously varying value.)

Device Tree Source (in software context) or Digital Temperature Sensor (in peripheral context)

Real Time Clock

Tamper

Consumer Electronics Control (HDMI standard)

Power Management Integrated Circuit

Display Serial Interface (MIPI® Alliance standard)

Analog-to-digital converter. The process of converting a sampled analog signal to a digital code that represents the amplitude of the original signal sample.

USB Host (STM32 specific)

Boot and Security and OTP control

Extended TrustZone Protection Controller



TrustZone® address space Controller for DDR

Arm® and TrustZone® are registered trademarks of Arm Limited (or its subsidiaries) in the US and/or elsewhere.

debug and test protocol, named from the Joint Test Action Group that developed it

System Trace Module

Graphics Processing Units

Doubledata rate (memory domain)

System Configuration

voltage reference buffer (STM32 specific)

Hardware Debug Port

Digital Camera Memory Interface

Inter-Processor Communication Controller

External Interrupt

Digital Filter for Sigma-Delta Modulator

Transmit

former spelling for e•MMC ('e' in italic)

Receive

LCD TFT Display Controller (STM32 specific)

System Time Generator

USB On-The-Go (Capability/type of USB port, acting primarily as USB device, to also act as USB host. Also known as USB OTG.)

input/output

Ethernet

Microprocessor Unit

Microcontroller Unit (MCUs have internal flash memory and are intended to operate with a minimum amount of external support ICs. They commonly are a self-contained, system-on-chip (SoC) designs.)

Discovery kit

BlueTooth

High-Definition Multimedia Interface (HDMI standard)

Stable: 17.06.2020 - 15:27 / Revision: 16.01.2020 - 13:12

All the peripherals receive one or several clocks that are generated via [RCC internal peripheral](#). RCC relies on several clock sources (LSI, LSE, HSI, HSE, CSI) and four PLL in order to provide adequate input frequencies to all the peripherals. The clock tree covers all the system clock distribution aspects, from the clock sources to the consumer peripherals (internal and external), except clock gating management that is locally controlled by each peripheral driver.

Contents

1 Overview	30
2 How to build a clock tree?	31



3 ST boards clock tree	32
3.1 STM32MP157x-EV1 case	32
3.1.1 Clock tree	32
3.1.1.1 Device tree	34
3.2 STM32MP157x-DK2 case	36
3.2.1 Clock tree	36
3.2.2 Device tree	38



1 Overview

The clock tree is managed via RCC internal peripheral hardware block and it is configured at different steps from the Cortex-A7:

- When the device is **reset**, all RCC registers take their reset values: the four PLL are disabled and most of the clock source selectors are pointing to the HSI.
- The ROM Code configures the minimum clock tree needed to boot on the selected boot device.
- The **FSBL** (Cf. [Boot chain overview](#)) completely configures the clock tree as expected, all the way up to Linux, thanks to the configuration given in the [device tree](#).
 - U-Boot SPL (basic boot chain) binding is available in [doc/device-tree-bindings/clock/st,stm32mp1.txt](#)
 - TF-A (trusted boot chain) binding follows the same syntax as U-Boot SPL
- **Linux** is responsible for the clock tree at runtime and it may partly modify it thanks to information taken from the [device tree](#):
 - Linux binding is available in [Documentation/devicetree/bindings/clock/clock-bindings.txt](#) (and surrounding files): 'fixed-clock' compatible, 'clocks' and 'assigned-clocks' properties are important concepts to understand the management of clocks providers/consumers.

Notice that [STM32Cube](#) (running on the Cortex-M4) does not control the clock tree setup so the configuration applied by the Cortex-A7 must provide suitable clocks to Cortex-M4, like it is explained in [resource manager](#) description. One exception to this rule is the [engineering boot](#), allowing to directly load (and debug) the Cortex-M4 whereas the Cortex-A7 execution is stalled in the ROM code.



2 How to build a clock tree?

Building a clock tree is quite complex as it needs to take into account the constraints set by each internal and external peripheral, including external clock sources.

We encourage the use of [STM32CubeMX](#) to build the clock tree, and avoid having to know all internal peripherals details: the tool allows to select the peripherals that will be present on the board, fix the clock sources frequencies and automatically find an optimized clock tree. It is then able to generate the device tree that is directly consumed by the boot chain and Linux.



3 ST boards clock tree

 See [How to change the CPU frequency](#) article for more information about the CPU frequency setting (PLL1P), including dynamic voltage and frequency scaling (DVFS) configuration.

3.1 STM32MP157x-EV1 case

This chapter shows the boot time clock tree set by the FSBL on STM32MP157x-EV1 evaluation board. Linux eventual runtime modifications are not covered here.

3.1.1 Clock tree

The following table shows what STM32MP157x-EV1 clock tree looks like, as a result of the boot chain execution with the device tree built with STM32CubeMX.

Component Comment	Parent	Frequency	Used?	
LSI	N.A.	0.032000 MHz	yes	Mandatory for IWDG, DAC
DAC	LSI	0.032000 MHz	yes	
RNG1	LSI	0.032000 MHz	yes	Input frequency should be
as low as possible				
RNG2	LSI	0.032000 MHz	yes	Input frequency should be
as low as possible				
IWDG1	LSI	0.032000 MHz	yes	
IWDG2	LSI	0.032000 MHz	yes	
LSE	N.A.	0.032768 MHz	yes	Mandatory for DTS
RTC	LSE	0.032768 MHz	yes	
TAMP	LSE	0.032768 MHz	yes	
CEC	LSE	0.032768 MHz	yes	
LPTIM4	LSE	0.032768 MHz	yes	
LPTIM5	LSE	0.032768 MHz	yes	
HSI	N.A.	64.000000 MHz	yes	
SPI4	HSI	64.000000 MHz	no	
SPI5	HSI	64.000000 MHz	no	
SPI6	HSI	64.000000 MHz	yes	
I2C4	HSI	64.000000 MHz	yes	PMIC
I2C6	HSI	64.000000 MHz	no	
I2C1	HSI	64.000000 MHz	no	
I2C2	HSI	64.000000 MHz	yes	Rpi and peripherals
I2C3	HSI	64.000000 MHz	no	
I2C5	HSI	64.000000 MHz	yes	Rpi
USART1	HSI	64.000000 MHz	yes	
USART2	HSI	64.000000 MHz	no	
USART3	HSI	64.000000 MHz	yes	Rpi
UART4	HSI	64.000000 MHz	yes	Linux console
UART5	HSI	64.000000 MHz	no	
USART6	HSI	64.000000 MHz	no	
UART7	HSI	64.000000 MHz	no	
UART8	HSI	64.000000 MHz	no	
MC01	HSI	64.000000 MHz	no	Available so can be used
HSE	N.A.	24.000000 MHz	yes	
DSIPLL	HSE	125.000000 MHz	no	DSI DPHY PLL
DSIBL	DSIPLL	125.000000 MHz	no	DSI lanebyte clock
RTCDIV	HSE	1.000000 MHz	yes	Only used when RTC source
is HSE				



ck_per	HSE	24.000000	MHz	yes	
ADC	ck_per	24.000000	MHz	yes	Can use internal divider to
be < 40MHz					
PLL1	HSE	xxx	MHz	yes	
PLL1P	PLL1	xxx	MHz	yes	
MPUDIV	PLL1P	xxx	MHz	yes	
Cortex-A7	PLL1P	xxx	MHz	yes	650 MHz or 800 MHz, see the
information box above					
MC02	Cortex-A7	xxx	MHz	no	Available so can be used
PLL2	HSE	1066.000000	MHz	yes	
PLL2P	PLL2	266.500000	MHz	yes	
AXI	PLL2P	266.500000	MHz	yes	< 266MHz
FMC	AXI	266.500000	MHz	yes	NAND flash
QSPI	AXI	266.500000	MHz	yes	NOR flash
SYSRAM	AXI	266.500000	MHz	yes	
ROM	AXI	266.500000	MHz	yes	
AHB5	AXI	266.500000	MHz	yes	< 266MHz
CRYP1	AHB5	266.500000	MHz	yes	
HASH1	AHB5	266.500000	MHz	yes	
GPIOZ	AHB5	266.500000	MHz	yes	
BKPSRAM	AHB5	266.500000	MHz	yes	
AHB6	AXI	266.500000	MHz	yes	< 266MHz
CRC1	AHB6	266.500000	MHz	yes	
MDMA	AHB6	266.500000	MHz	yes	
USBH	AHB6	266.500000	MHz	yes	USB Host
APB4	AHB6	133.250000	MHz	yes	
APB5	AHB6	66.625000	MHz	yes	
BSEC	APB5	66.625000	MHz	yes	< 67MHz
ETZPC	APB5	66.625000	MHz	yes	
TZC	APB5	66.625000	MHz	yes	
DBGAPB	AXI	133.250000	MHz	yes	JTAG & Coresight
DBGMCU	DBGAPB	66.625000	MHz	yes	
STM	DBGAPB	66.625000	MHz	yes	
PLL2Q	PLL2	533.000000	MHz	yes	
GPU	PLL2Q	533.000000	MHz	yes	< 533MHz
PLL2R	PLL2	533.000000	MHz	yes	
DDR	PLL2R	533.000000	MHz	yes	< 533MHz
PLL3	HSE	417.755859	MHz	yes	
PLL3P	PLL3	208.877930	MHz	yes	
MLAHB	PLL3P	208.877930	MHz	yes	< 209MHz
Cortex-M4	MLAHB	208.877930	MHz	yes	
SRAM1	MLAHB	208.877930	MHz	yes	
SRAM2	MLAHB	208.877930	MHz	yes	
SRAM3	MLAHB	208.877930	MHz	yes	
RETRAM	MLAHB	208.877930	MHz	yes	
AHB1	MLAHB	208.877930	MHz	yes	< 209MHz
AHB2	MLAHB	208.877930	MHz	yes	< 209MHz
DMA1	AHB2	208.877930	MHz	yes	
DMA2	AHB2	208.877930	MHz	yes	
DMAMUX	AHB2	208.877930	MHz	yes	
APB1	MLAHB	104.438965	MHz	yes	
LPTIM1	APB1	104.438965	MHz	yes	
WWDG	APB1	104.438965	MHz	yes	
APB2	MLAHB	104.438965	MHz	yes	
TIM2	MLAHB	208.877930	MHz	yes	TIM Group 1
TIM3	MLAHB	208.877930	MHz	yes	TIM Group 1
TIM4	MLAHB	208.877930	MHz	yes	TIM Group 1
TIM5	MLAHB	208.877930	MHz	yes	TIM Group 1
TIM6	MLAHB	208.877930	MHz	yes	TIM Group 1
TIM7	MLAHB	208.877930	MHz	yes	TIM Group 1
TIM12	MLAHB	208.877930	MHz	yes	TIM Group 1
TIM13	MLAHB	208.877930	MHz	yes	TIM Group 1
TIM14	MLAHB	208.877930	MHz	yes	TIM Group 1
TIM1	MLAHB	208.877930	MHz	yes	TIM Group 2
TIM8	MLAHB	208.877930	MHz	yes	TIM Group 2
TIM15	MLAHB	208.877930	MHz	yes	TIM Group 2
TIM16	MLAHB	208.877930	MHz	yes	TIM Group 2



TIM17	MLAHB	208.877930	MHz	yes	TIM Group 2
APB3	MLAHB	104.438965	MHz	yes	
LPTIM2	APB3	104.438965	MHz	yes	
LPTIM3	APB3	104.438965	MHz	yes	
SYSCFG	APB3	104.438965	MHz	yes	
VREFBUF	APB3	104.438965	MHz	yes	
DTS	APB3	104.438965	MHz	yes	
HDP	APB3	104.438965	MHz	yes	
AHB3	MLAHB	208.877930	MHz	yes	< 209MHz
CRC2	AHB3	208.877930	MHz	yes	
CRYP2	AHB3	208.877930	MHz	yes	
HASH2	AHB3	208.877930	MHz	yes	
DCMI	AHB3	208.877930	MHz	yes	Camera
IPCC	AHB3	208.877930	MHz	yes	Mailbox
AHB4	MLAHB	208.877930	MHz	yes	< 209MHz
PWR	AHB4	208.877930	MHz	yes	
RCC	AHB4	208.877930	MHz	yes	
GPIOA-K	AHB4	208.877930	MHz	yes	
EXTI	AHB4	208.877930	MHz	yes	
PLL3Q	PLL3	24.573874	MHz	yes	
SPI1	PLL3Q	24.573874	MHz	no	
SPI2	PLL3Q	24.573874	MHz	no	
SPI3	PLL3Q	24.573874	MHz	no	
DFSDM	PLL3Q	24.573874	MHz	yes	Digital micro
SAI1	PLL3Q	24.573874	MHz	no	
SAI2	PLL3Q	24.573874	MHz	yes	AudCodec 48kHz (use PLL3R
for 44.1kHz)					
SAI3	PLL3Q	24.573874	MHz	no	
SAI4	PLL3Q	24.573874	MHz	yes	SPDIF TX 48kHz (use PLL3R
for 44.1kHz)					
PLL3R	PLL3	11.290699	MHz	yes	
PLL4	HSE	594.000000	MHz	yes	
PLL4P	PLL4	99.000000	MHz	yes	
SDMMC1	PLL4P	99.000000	MHz	yes	µSD card
SDMMC2	PLL4P	99.000000	MHz	yes	eMMC
SDMMC3	PLL4P	99.000000	MHz	yes	
SPDIF	PLL4P	99.000000	MHz	yes	SPDIF RX
PLL4Q	PLL4	74.250000	MHz	yes	
LCD	PLL4Q	74.250000	MHz	yes	LTDC & DSI display pixel
clock					
PLL4R	PLL4	74.250000	MHz	yes	
FDCAN	PLL4R	74.250000	MHz	yes	Should be as high as
possible and < 100MHz					
STGEN	HSE	24.000000	MHz	yes	
USBPHYC	HSE	24.000000	MHz	yes	USB PHY Ctrl for USB Host
and OTG					
USBPLL	USBPHYC	48.000000	MHz	yes	
USB0	USBPLL	48.000000	MHz	yes	USB OTG
CSI	N.A.	4.000000	MHz	yes	Mandatory for IO
compensation					
ETH	N.A.	0.000000	MHz	no	ETH clocked by RGMII PHY

3.1.1.1 Device tree

Here are the corresponding device tree `rcc_clk` sub node properties consumed by the first stage boot loader (FSBL) to configure the clock tree above:

```
st,clksrc = <
    CLK_MPU_PLL1P
    CLK_AXI_PLL2P
    CLK_MCU_PLL3P
    CLK_PLL12_HSE
```



```

        CLK_PLL3_HSE
        CLK_PLL4_HSE
        CLK_RTC_LSE
        CLK_MC01_DISABLED
        CLK_MC02_DISABLED
    >;

    st,clkdiv = <
        1 /*MPU*/
        0 /*AXI*/
        0 /*MCU*/
        1 /*APB1*/
        1 /*APB2*/
        1 /*APB3*/
        1 /*APB4*/
        2 /*APB5*/
        23 /*RTC*/
        0 /*MC01*/
        0 /*MC02*/
    >;

    st,pkcs = <
        CLK_CKPER_HSE
        CLK_FMC_ACLK
        CLK_QSPI_ACLK
        CLK_ETH_DISABLED
        CLK_SDMMC12_PLL4P
        CLK_DSI_DSIPLL
        CLK_STGEN_HSE
        CLK_USBPHY_HSE
        CLK_SPI2S1_PLL3Q
        CLK_SPI2S23_PLL3Q
        CLK_SPI45_HSI
        CLK_SPI6_HSI
        CLK_I2C46_HSI
        CLK_SDMMC3_PLL4P
        CLK_USB0_USBPHY
        CLK_ADC_CKPER
        CLK_CEC_LSE
        CLK_I2C12_HSI
        CLK_I2C35_HSI
        CLK_UART1_HSI
        CLK_UART24_HSI
        CLK_UART35_HSI
        CLK_UART6_HSI
        CLK_UART78_HSI
        CLK_SPDIF_PLL4P
        CLK_FDCAN_PLL4R
        CLK_SAI1_PLL3Q
        CLK_SAI2_PLL3Q
        CLK_SAI3_PLL3Q
        CLK_SAI4_PLL3Q
        CLK_RNG1_LSI
        CLK_RNG2_LSI
        CLK_LPTIM1_PCLK1
        CLK_LPTIM23_PCLK3
        CLK_LPTIM45_LSE
    >;

    /* VCO = 1066.0 MHz => P = 266 (AXI), Q = 533 (GPU), R = 533 (DDR) */
    pll2: st,pll@1 {
        cfg = < 2 65 1 0 0 PQR(1,1,1) >;
        frac = < 0x1400 >;
        u-boot,dm-pre-reloc;
    };

    /* VCO = 417.8 MHz => P = 209, Q = 25, R = 11 */

```



```

pll3: st,pll@2 {
    cfg = < 1 33 1 16 36 PQR(1,1,1) >;
    frac = < 0x1a04 >;
    u-boot,dm-pre-reloc;
};

/* VCO = 594.0 MHz => P = 99, Q = 74, R = 74 */
pll4: st,pll@3 {
    cfg = < 3 98 5 7 7 PQR(1,1,1) >;
    u-boot,dm-pre-reloc;
};
    
```

3.2 STM32MP157x-DK2 case

This chapter shows the boot time clock tree set by the FSBL on STM32MP157x-DK2 DISCO board. Linux eventual runtime modifications are not covered here.

3.2.1 Clock tree

The following table shows what STM32MP157x-DK2 clock tree looks like, as a result of the boot chain execution with the device tree built with STM32CubeMX.

Component Comment	Parent	Frequency	Used?	
LSI	N.A.	0.032000 MHz	yes	Mandatory for IWDG, DAC
DAC	LSI	0.032000 MHz	no	
RNG1	LSI	0.032000 MHz	yes	Input frequency should be as low as possible
RNG2	LSI	0.032000 MHz	yes	Input frequency should be as low as possible
IWDG1	LSI	0.032000 MHz	yes	
IWDG2	LSI	0.032000 MHz	yes	
LSE	N.A.	0.032768 MHz	yes	Mandatory for DTS
RTC	LSE	0.032768 MHz	yes	
TAMP	LSE	0.032768 MHz	yes	
CEC	LSE	0.032768 MHz	yes	
LPTIM4	LSE	0.032768 MHz	yes	
LPTIM5	LSE	0.032768 MHz	yes	
HSI	N.A.	64.000000 MHz	yes	
SPI4	HSI	64.000000 MHz	yes	Arduino
SPI5	HSI	64.000000 MHz	yes	Rpi
SPI6	HSI	64.000000 MHz	no	
I2C4	HSI	64.000000 MHz	yes	PMIC
I2C6	HSI	64.000000 MHz	no	
I2C1	HSI	64.000000 MHz	yes	Rpi and peripherals
I2C2	HSI	64.000000 MHz	no	
I2C3	HSI	64.000000 MHz	no	
I2C5	HSI	64.000000 MHz	yes	Rpi and Arduino
USART1	HSI	64.000000 MHz	no	
USART2	HSI	64.000000 MHz	yes	Bluetooth
USART3	HSI	64.000000 MHz	yes	Rpi
UART4	HSI	64.000000 MHz	yes	Linux console
UART5	HSI	64.000000 MHz	no	
USART6	HSI	64.000000 MHz	no	
UART7	HSI	64.000000 MHz	yes	Arduino
UART8	HSI	64.000000 MHz	no	
MCO1	HSI	64.000000 MHz	no	Available so can be used
HSE	N.A.	24.000000 MHz	yes	
DSIPLL	HSE	125.000000 MHz	no	DSI DPHY PLL
DSIBL	DSIPLL	125.000000 MHz	no	DSI lanebyte clock



RTCDIV	HSE	1.000000 MHz	yes	Only used when RTC source
is HSE				
ck_per	HSE	24.000000 MHz	yes	
ADC	ck_per	24.000000 MHz	yes	Can use internal divider to
be < 40MHz				
PLL1	HSE	xxx MHz	yes	
PLL1P	PLL1	xxx MHz	yes	
MPUDIV	PLL1P	xxx MHz	yes	
Cortex-A7	PLL1P	xxx MHz	yes	650 MHz or 800 MHz, see the
information box above				
MC02	Cortex-A7	xxx MHz	no	Available so can be used
PLL2	HSE	1066.000000 MHz	yes	
PLL2P	PLL2	266.500000 MHz	yes	
AXI	PLL2P	266.500000 MHz	yes	< 266MHz
FMC	AXI	266.500000 MHz	no	
QSPI	AXI	266.500000 MHz	no	
SYSRAM	AXI	266.500000 MHz	yes	
ROM	AXI	266.500000 MHz	yes	
AHB5	AXI	266.500000 MHz	yes	< 266MHz
CRYP1	AHB5	266.500000 MHz	yes	
HASH1	AHB5	266.500000 MHz	yes	
GPIOZ	AHB5	266.500000 MHz	yes	
BKPSRAM	AHB5	266.500000 MHz	yes	
AHB6	AXI	266.500000 MHz	yes	< 266MHz
CRC1	AHB6	266.500000 MHz	yes	
MDMA	AHB6	266.500000 MHz	yes	
USBH	AHB6	266.500000 MHz	yes	USB Host
APB4	AHB6	133.250000 MHz	yes	
APB5	AHB6	66.625000 MHz	yes	
BSEC	APB5	66.625000 MHz	yes	< 67MHz
ETZPC	APB5	66.625000 MHz	yes	
TZC	APB5	66.625000 MHz	yes	
DBGAPB	AXI	133.250000 MHz	yes	JTAG & Coresight
DBGMCU	DBGAPB	66.625000 MHz	yes	
STM	DBGAPB	66.625000 MHz	no	
PLL2Q	PLL2	533.000000 MHz	yes	
GPU	PLL2Q	533.000000 MHz	yes	< 533MHz
PLL2R	PLL2	533.000000 MHz	yes	
DDR	PLL2R	533.000000 MHz	yes	< 533MHz
PLL3	HSE	417.755859 MHz	yes	
PLL3P	PLL3	208.877930 MHz	yes	
MLAHB	PLL3P	208.877930 MHz	yes	< 209MHz
Cortex-M4	MLAHB	208.877930 MHz	yes	
SRAM1	MLAHB	208.877930 MHz	yes	
SRAM2	MLAHB	208.877930 MHz	yes	
SRAM3	MLAHB	208.877930 MHz	yes	
RETRAM	MLAHB	208.877930 MHz	yes	
AHB1	MLAHB	208.877930 MHz	yes	< 209MHz
AHB2	MLAHB	208.877930 MHz	yes	< 209MHz
DMA1	AHB2	208.877930 MHz	yes	
DMA2	AHB2	208.877930 MHz	yes	
DMAMUX	AHB2	208.877930 MHz	yes	
APB1	MLAHB	104.438965 MHz	yes	
LPTIM1	APB1	104.438965 MHz	yes	
WWDG	APB1	104.438965 MHz	yes	
APB2	MLAHB	104.438965 MHz	yes	
TIM2	MLAHB	208.877930 MHz	yes	TIM Group 1
TIM3	MLAHB	208.877930 MHz	yes	TIM Group 1
TIM4	MLAHB	208.877930 MHz	yes	TIM Group 1
TIM5	MLAHB	208.877930 MHz	yes	TIM Group 1
TIM6	MLAHB	208.877930 MHz	yes	TIM Group 1
TIM7	MLAHB	208.877930 MHz	yes	TIM Group 1
TIM12	MLAHB	208.877930 MHz	yes	TIM Group 1
TIM13	MLAHB	208.877930 MHz	yes	TIM Group 1
TIM14	MLAHB	208.877930 MHz	yes	TIM Group 1
TIM1	MLAHB	208.877930 MHz	yes	TIM Group 2
TIM8	MLAHB	208.877930 MHz	yes	TIM Group 2



TIM15	MLAHB	208.877930	MHz	yes	TIM Group 2
TIM16	MLAHB	208.877930	MHz	yes	TIM Group 2
TIM17	MLAHB	208.877930	MHz	yes	TIM Group 2
APB3	MLAHB	104.438965	MHz	yes	
LPTIM2	APB3	104.438965	MHz	yes	
LPTIM3	APB3	104.438965	MHz	yes	
SYSCFG	APB3	104.438965	MHz	yes	
VREFBUF	APB3	104.438965	MHz	yes	
DTS	APB3	104.438965	MHz	yes	
HDP	APB3	104.438965	MHz	no	
AHB3	MLAHB	208.877930	MHz	yes	< 209MHz
CRC2	AHB3	208.877930	MHz	yes	
CRYP2	AHB3	208.877930	MHz	yes	
HASH2	AHB3	208.877930	MHz	yes	
DCMI	AHB3	208.877930	MHz	no	
IPCC	AHB3	208.877930	MHz	yes	Mailbox
AHB4	MLAHB	208.877930	MHz	yes	< 209MHz
PWR	AHB4	208.877930	MHz	yes	
RCC	AHB4	208.877930	MHz	yes	
GPIOA-K	AHB4	208.877930	MHz	yes	
EXTI	AHB4	208.877930	MHz	yes	
PLL3Q	PLL3	24.573874	MHz	yes	
SPI1	PLL3Q	24.573874	MHz	yes	SPI2S1 for BT PCM
SPI2	PLL3Q	24.573874	MHz	yes	SPI2S2 for HDMI
SPI3	PLL3Q	24.573874	MHz	no	
DFSDM	PLL3Q	24.573874	MHz	no	
SAI1	PLL3Q	24.573874	MHz	no	
SAI2	PLL3Q	24.573874	MHz	yes	AudCodec 48kHz (use PLL3R)
for 44.1kHz)					
SAI3	PLL3Q	24.573874	MHz	no	
SAI4	PLL3Q	24.573874	MHz	no	
PLL3R	PLL3	11.290699	MHz	yes	
PLL4	HSE	594.000000	MHz	yes	
PLL4P	PLL4	99.000000	MHz	yes	
SDMMC1	PLL4P	99.000000	MHz	yes	µSD card
SDMMC2	PLL4P	99.000000	MHz	yes	Wifi
SDMMC3	PLL4P	99.000000	MHz	yes	Rpi
SPDIF	PLL4P	99.000000	MHz	no	
PLL4Q	PLL4	74.250000	MHz	yes	
LCD	PLL4Q	74.250000	MHz	yes	LTDC & DSI display pixel
clock					
PLL4R	PLL4	74.250000	MHz	yes	
FDCAN	PLL4R	74.250000	MHz	no	
STGEN	HSE	24.000000	MHz	yes	
USBPHYC	HSE	24.000000	MHz	yes	USB PHY Ctrl for USB Host
and OTG					
USBPLL	USBPHYC	48.000000	MHz	yes	
USB0	USBPLL	48.000000	MHz	yes	USB OTG
CSI	N.A.	4.000000	MHz	yes	Mandatory for IO
compensation					
ETH	N.A.	0.000000	MHz	no	ETH clocked by RGMII PHY

3.2.2 Device tree

Here are the corresponding device tree `rcc_clk` sub node properties consumed by the first stage boot loader (FSBL) to configure the clock tree above:

```
st,clksrc = <
    CLK_MPU_PLL1P
    CLK_AXI_PLL2P
    CLK_MCU_PLL3P
    CLK_PLL12_HSE
```



```

        CLK_PLL3_HSE
        CLK_PLL4_HSE
        CLK_RTC_LSE
        CLK_MC01_DISABLED
        CLK_MC02_DISABLED
    >;

    st,clkdiv = <
        1 /*MPU*/
        0 /*AXI*/
        0 /*MCU*/
        1 /*APB1*/
        1 /*APB2*/
        1 /*APB3*/
        1 /*APB4*/
        2 /*APB5*/
        23 /*RTC*/
        0 /*MC01*/
        0 /*MC02*/
    >;

    st,pkcs = <
        CLK_CKPER_HSE
        CLK_FMC_ACLK
        CLK_QSPI_ACLK
        CLK_ETH_DISABLED
        CLK_SDMMC12_PLL4P
        CLK_DSI_DSIPLL
        CLK_STGEN_HSE
        CLK_USBPHY_HSE
        CLK_SPI2S1_PLL3Q
        CLK_SPI2S23_PLL3Q
        CLK_SPI45_HSI
        CLK_SPI6_HSI
        CLK_I2C46_HSI
        CLK_SDMMC3_PLL4P
        CLK_USB0_USBPHY
        CLK_ADC_CKPER
        CLK_CEC_LSE
        CLK_I2C12_HSI
        CLK_I2C35_HSI
        CLK_UART1_HSI
        CLK_UART24_HSI
        CLK_UART35_HSI
        CLK_UART6_HSI
        CLK_UART78_HSI
        CLK_SPDIF_PLL4P
        CLK_FDCAN_PLL4R
        CLK_SAI1_PLL3Q
        CLK_SAI2_PLL3Q
        CLK_SAI3_PLL3Q
        CLK_SAI4_PLL3Q
        CLK_RNG1_LSI
        CLK_RNG2_LSI
        CLK_LPTIM1_PCLK1
        CLK_LPTIM23_PCLK3
        CLK_LPTIM45_LSE
    >;

    /* VCO = 1066.0 MHz => P = 266 (AXI), Q = 533 (GPU), R = 533 (DDR) */
    pll2: st,pll@1 {
        cfg = < 2 65 1 0 0 PQR(1,1,1) >;
        frac = < 0x1400 >;
        u-boot,dm-pre-reloc;
    };

    /* VCO = 417.8 MHz => P = 209, Q = 25, R = 11 */

```



```

pll3: st,pll@2 {
    cfg = < 1 33 1 16 36 PQR(1,1,1) >;
    frac = < 0x1a04 >;
    u-boot,dm-pre-reloc;
};

/* VCO = 594.0 MHz => P = 99, Q = 74, R = 74 */
pll4: st,pll@3 {
    cfg = < 3 98 5 7 7 PQR(1,1,1) >;
    u-boot,dm-pre-reloc;
};

```

Reset and Clock Control

Low Speed Internal oscillator (STM32 clock source)

Low Speed External oscillator (STM32 clock source)

High Speed Internal oscillator (STM32 clock source) or High Speed Synchronous Serial Interface (MIPI® Alliance standard)

High Speed External oscillator (STM32 clock source)

Multi Speed Internal oscillator (STM32 clock source)

Cortex®

First Stage Boot Loader

Linux® is a registered trademark of Linus Torvalds.

Das U-Boot -- the Universal Boot Loader (see [U-Boot_overview](#))

Secondary Program Loader, *Also known as **U-Boot SPL***

Read Only Memory

Central processing unit

Dynamic voltage and frequency scaling

Independent Watchdog

Digital-to-analog converter (Electronic circuit that converts a binary number into a continuously varying value.)

Device Tree Source (in software context) or Digital Temperature Sensor (in peripheral context)

Real Time Clock

Tamper

Consumer Electronics Control (HDMI standard)

Power Management Integrated Circuit

Display Serial Interface (MIPI® Alliance standard)

Analog-to-digital converter. The process of converting a sampled analog signal to a digital code that represents the amplitude of the original signal sample.

USB Host (STM32 specific)

Boot and Security and OTP control

Extended TrustZone Protection Controller



TrustZone® address space Controller for DDR

Arm® and TrustZone® are registered trademarks of Arm Limited (or its subsidiaries) in the US and/or elsewhere.

debug and test protocol, named from the Joint Test Action Group that developed it

System Trace Module

Graphics Processing Units

Doubledata rate (memory domain)

System Configuration

voltage reference buffer (STM32 specific)

Hardware Debug Port

Digital Camera Memory Interface

Inter-Processor Communication Controller

External Interrupt

Digital Filter for Sigma-Delta Modulator

Transmit

former spelling for e•MMC ('e' in italic)

Receive

LCD TFT Display Controller (STM32 specific)

System Time Generator

USB On-The-Go (Capability/type of USB port, acting primarily as USB device, to also act as USB host. Also known as USB OTG.)

input/output

Ethernet

Microprocessor Unit

Microcontroller Unit (MCUs have internal flash memory and are intended to operate with a minimum amount of external support ICs. They commonly are a self-contained, system-on-chip (SoC) designs.)

Discovery kit

BlueTooth

High-Definition Multimedia Interface (HDMI standard)



Pages in category "STM32MP15 Discovery kits"

The following 6 pages are in this category, out of 6 total.

0

- STM32MP15 Discovery kits - getting started
- STM32MP157x-DKx - hardware description

L

- LEDs and buttons on STM32 MPU boards

M

- MB1272
- MB1407

0

- [STM32MP15 Discovery kits - Starter Package](#)

Stable: 17.06.2020 - 15:27 / Revision: 16.01.2020 - 09:31

All the peripherals receive one or several clocks that are generated via **RCC internal peripheral**. RCC relies on several clocks sources (LSI, LSE, HSI, HSE, CSI) and four PLL in order to provide adequate input frequencies to all the peripherals. The clock tree covers all the system clock distribution aspects, from the clock sources to the consumer peripherals (internal and external), except clock gating management that is locally controlled by each peripheral driver.

Contents

1 Overview	43
2 How to build a clock tree?	44
3 ST boards clock tree	45
3.1 STM32MP157x-EV1 case	45
3.1.1 Clock tree	45
3.1.1.1 Device tree	47
3.2 STM32MP157x-DK2 case	49
3.2.1 Clock tree	49
3.2.2 Device tree	51



1 Overview

The clock tree is managed via RCC internal peripheral hardware block and it is configured at different steps from the Cortex-A7:

- When the device is **reset**, all RCC registers take their reset values: the four PLL are disabled and most of the clock source selectors are pointing to the HSI.
- The ROM Code configures the minimum clock tree needed to boot on the selected boot device.
- The **FSBL** (Cf. [Boot chain overview](#)) completely configures the clock tree as expected, all the way up to Linux, thanks to the configuration given in the [device tree](#).
 - U-Boot SPL (basic boot chain) binding is available in [doc/device-tree-bindings/clock/st,stm32mp1.txt](#)
 - TF-A (trusted boot chain) binding follows the same syntax as U-Boot SPL
- **Linux** is responsible for the clock tree at runtime and it may partly modify it thanks to information taken from the [device tree](#):
 - Linux binding is available in [Documentation/devicetree/bindings/clock/clock-bindings.txt](#) (and surrounding files): 'fixed-clock' compatible, 'clocks' and 'assigned-clocks' properties are important concepts to understand the management of clocks providers/consumers.

Notice that [STM32Cube](#) (running on the Cortex-M4) does not control the clock tree setup so the configuration applied by the Cortex-A7 must provide suitable clocks to Cortex-M4, like it is explained in [resource manager](#) description. One exception to this rule is the [engineering boot](#), allowing to directly load (and debug) the Cortex-M4 whereas the Cortex-A7 execution is stalled in the ROM code.



2 How to build a clock tree?

Building a clock tree is quite complex as it needs to take into account the constraints set by each internal and external peripheral, including external clock sources.

We encourage the use of [STM32CubeMX](#) to build the clock tree, and avoid having to know all internal peripherals details: the tool allows to select the peripherals that will be present on the board, fix the clock sources frequencies and automatically find an optimized clock tree. It is then able to generate the device tree that is directly consumed by the boot chain and Linux.



3 ST boards clock tree

 See [How to change the CPU frequency](#) article for more information about the CPU frequency setting (PLL1P), including dynamic voltage and frequency scaling (DVFS) configuration.

3.1 STM32MP157x-EV1 case

This chapter shows the boot time clock tree set by the FSBL on STM32MP157x-EV1 evaluation board. Linux eventual runtime modifications are not covered here.

3.1.1 Clock tree

The following table shows what STM32MP157x-EV1 clock tree looks like, as a result of the boot chain execution with the device tree built with STM32CubeMX.

Component Comment	Parent	Frequency	Used?	
LSI	N.A.	0.032000 MHz	yes	Mandatory for IWDG, DAC
DAC	LSI	0.032000 MHz	yes	
RNG1	LSI	0.032000 MHz	yes	Input frequency should be
as low as possible				
RNG2	LSI	0.032000 MHz	yes	Input frequency should be
as low as possible				
IWDG1	LSI	0.032000 MHz	yes	
IWDG2	LSI	0.032000 MHz	yes	
LSE	N.A.	0.032768 MHz	yes	Mandatory for DTS
RTC	LSE	0.032768 MHz	yes	
TAMP	LSE	0.032768 MHz	yes	
CEC	LSE	0.032768 MHz	yes	
LPTIM4	LSE	0.032768 MHz	yes	
LPTIM5	LSE	0.032768 MHz	yes	
HSI	N.A.	64.000000 MHz	yes	
SPI4	HSI	64.000000 MHz	no	
SPI5	HSI	64.000000 MHz	no	
SPI6	HSI	64.000000 MHz	yes	
I2C4	HSI	64.000000 MHz	yes	PMIC
I2C6	HSI	64.000000 MHz	no	
I2C1	HSI	64.000000 MHz	no	
I2C2	HSI	64.000000 MHz	yes	Rpi and peripherals
I2C3	HSI	64.000000 MHz	no	
I2C5	HSI	64.000000 MHz	yes	Rpi
USART1	HSI	64.000000 MHz	yes	
USART2	HSI	64.000000 MHz	no	
USART3	HSI	64.000000 MHz	yes	Rpi
UART4	HSI	64.000000 MHz	yes	Linux console
UART5	HSI	64.000000 MHz	no	
UART6	HSI	64.000000 MHz	no	
UART7	HSI	64.000000 MHz	no	
UART8	HSI	64.000000 MHz	no	
MC01	HSI	64.000000 MHz	no	Available so can be used
HSE	N.A.	24.000000 MHz	yes	
DSIPLL	HSE	125.000000 MHz	no	DSI DPHY PLL
DSIBL	DSIPLL	125.000000 MHz	no	DSI lanebyte clock
RTCDIV	HSE	1.000000 MHz	yes	Only used when RTC source
is HSE				



ck_per	HSE	24.000000	MHz	yes	
ADC	ck_per	24.000000	MHz	yes	Can use internal divider to
be < 40MHz					
PLL1	HSE	xxx	MHz	yes	
PLL1P	PLL1	xxx	MHz	yes	
MPUDIV	PLL1P	xxx	MHz	yes	
Cortex-A7	PLL1P	xxx	MHz	yes	650 MHz or 800 MHz, see the
information box above					
MC02	Cortex-A7	xxx	MHz	no	Available so can be used
PLL2	HSE	1066.000000	MHz	yes	
PLL2P	PLL2	266.500000	MHz	yes	
AXI	PLL2P	266.500000	MHz	yes	< 266MHz
FMC	AXI	266.500000	MHz	yes	NAND flash
QSPI	AXI	266.500000	MHz	yes	NOR flash
SYSRAM	AXI	266.500000	MHz	yes	
ROM	AXI	266.500000	MHz	yes	
AHB5	AXI	266.500000	MHz	yes	< 266MHz
CRYP1	AHB5	266.500000	MHz	yes	
HASH1	AHB5	266.500000	MHz	yes	
GPIOZ	AHB5	266.500000	MHz	yes	
BKPSRAM	AHB5	266.500000	MHz	yes	
AHB6	AXI	266.500000	MHz	yes	< 266MHz
CRC1	AHB6	266.500000	MHz	yes	
MDMA	AHB6	266.500000	MHz	yes	
USBH	AHB6	266.500000	MHz	yes	USB Host
APB4	AHB6	133.250000	MHz	yes	
APB5	AHB6	66.625000	MHz	yes	
BSEC	APB5	66.625000	MHz	yes	< 67MHz
ETZPC	APB5	66.625000	MHz	yes	
TZC	APB5	66.625000	MHz	yes	
DBGAPB	AXI	133.250000	MHz	yes	JTAG & Coresight
DBGMCU	DBGAPB	66.625000	MHz	yes	
STM	DBGAPB	66.625000	MHz	yes	
PLL2Q	PLL2	533.000000	MHz	yes	
GPU	PLL2Q	533.000000	MHz	yes	< 533MHz
PLL2R	PLL2	533.000000	MHz	yes	
DDR	PLL2R	533.000000	MHz	yes	< 533MHz
PLL3	HSE	417.755859	MHz	yes	
PLL3P	PLL3	208.877930	MHz	yes	
MLAHB	PLL3P	208.877930	MHz	yes	< 209MHz
Cortex-M4	MLAHB	208.877930	MHz	yes	
SRAM1	MLAHB	208.877930	MHz	yes	
SRAM2	MLAHB	208.877930	MHz	yes	
SRAM3	MLAHB	208.877930	MHz	yes	
RETRAM	MLAHB	208.877930	MHz	yes	
AHB1	MLAHB	208.877930	MHz	yes	< 209MHz
AHB2	MLAHB	208.877930	MHz	yes	< 209MHz
DMA1	AHB2	208.877930	MHz	yes	
DMA2	AHB2	208.877930	MHz	yes	
DMAMUX	AHB2	208.877930	MHz	yes	
APB1	MLAHB	104.438965	MHz	yes	
LPTIM1	APB1	104.438965	MHz	yes	
WWDG	APB1	104.438965	MHz	yes	
APB2	MLAHB	104.438965	MHz	yes	
TIM2	MLAHB	208.877930	MHz	yes	TIM Group 1
TIM3	MLAHB	208.877930	MHz	yes	TIM Group 1
TIM4	MLAHB	208.877930	MHz	yes	TIM Group 1
TIM5	MLAHB	208.877930	MHz	yes	TIM Group 1
TIM6	MLAHB	208.877930	MHz	yes	TIM Group 1
TIM7	MLAHB	208.877930	MHz	yes	TIM Group 1
TIM12	MLAHB	208.877930	MHz	yes	TIM Group 1
TIM13	MLAHB	208.877930	MHz	yes	TIM Group 1
TIM14	MLAHB	208.877930	MHz	yes	TIM Group 1
TIM1	MLAHB	208.877930	MHz	yes	TIM Group 2
TIM8	MLAHB	208.877930	MHz	yes	TIM Group 2
TIM15	MLAHB	208.877930	MHz	yes	TIM Group 2
TIM16	MLAHB	208.877930	MHz	yes	TIM Group 2



TIM17	MLAHB	208.877930	MHz	yes	TIM Group 2
APB3	MLAHB	104.438965	MHz	yes	
LPTIM2	APB3	104.438965	MHz	yes	
LPTIM3	APB3	104.438965	MHz	yes	
SYSCFG	APB3	104.438965	MHz	yes	
VREFBUF	APB3	104.438965	MHz	yes	
DTS	APB3	104.438965	MHz	yes	
HDP	APB3	104.438965	MHz	yes	
AHB3	MLAHB	208.877930	MHz	yes	< 209MHz
CRC2	AHB3	208.877930	MHz	yes	
CRYP2	AHB3	208.877930	MHz	yes	
HASH2	AHB3	208.877930	MHz	yes	
DCMI	AHB3	208.877930	MHz	yes	Camera
IPCC	AHB3	208.877930	MHz	yes	Mailbox
AHB4	MLAHB	208.877930	MHz	yes	< 209MHz
PWR	AHB4	208.877930	MHz	yes	
RCC	AHB4	208.877930	MHz	yes	
GPIOA-K	AHB4	208.877930	MHz	yes	
EXTI	AHB4	208.877930	MHz	yes	
PLL3Q	PLL3	24.573874	MHz	yes	
SPI1	PLL3Q	24.573874	MHz	no	
SPI2	PLL3Q	24.573874	MHz	no	
SPI3	PLL3Q	24.573874	MHz	no	
DFSDM	PLL3Q	24.573874	MHz	yes	Digital micro
SAI1	PLL3Q	24.573874	MHz	no	
SAI2	PLL3Q	24.573874	MHz	yes	AudCodec 48kHz (use PLL3R
for 44.1kHz)					
SAI3	PLL3Q	24.573874	MHz	no	
SAI4	PLL3Q	24.573874	MHz	yes	SPDIF TX 48kHz (use PLL3R
for 44.1kHz)					
PLL3R	PLL3	11.290699	MHz	yes	
PLL4	HSE	594.000000	MHz	yes	
PLL4P	PLL4	99.000000	MHz	yes	
SDMMC1	PLL4P	99.000000	MHz	yes	µSD card
SDMMC2	PLL4P	99.000000	MHz	yes	eMMC
SDMMC3	PLL4P	99.000000	MHz	yes	
SPDIF	PLL4P	99.000000	MHz	yes	SPDIF RX
PLL4Q	PLL4	74.250000	MHz	yes	
LCD	PLL4Q	74.250000	MHz	yes	LTDC & DSI display pixel
clock					
PLL4R	PLL4	74.250000	MHz	yes	
FDCAN	PLL4R	74.250000	MHz	yes	Should be as high as
possible and < 100MHz					
STGEN	HSE	24.000000	MHz	yes	
USBPHYC	HSE	24.000000	MHz	yes	USB PHY Ctrl for USB Host
and OTG					
USBPLL	USBPHYC	48.000000	MHz	yes	
USB0	USBPLL	48.000000	MHz	yes	USB OTG
CSI	N.A.	4.000000	MHz	yes	Mandatory for IO
compensation					
ETH	N.A.	0.000000	MHz	no	ETH clocked by RGMII PHY

3.1.1.1 Device tree

Here are the corresponding device tree `rcc_clk` sub node properties consumed by the first stage boot loader (FSBL) to configure the clock tree above:

```
st,clksrc = <
    CLK_MPU_PLL1P
    CLK_AXI_PLL2P
    CLK_MCU_PLL3P
    CLK_PLL12_HSE
```



```

        CLK_PLL3_HSE
        CLK_PLL4_HSE
        CLK_RTC_LSE
        CLK_MC01_DISABLED
        CLK_MC02_DISABLED
    >;

    st,clkdiv = <
        1 /*MPU*/
        0 /*AXI*/
        0 /*MCU*/
        1 /*APB1*/
        1 /*APB2*/
        1 /*APB3*/
        1 /*APB4*/
        2 /*APB5*/
        23 /*RTC*/
        0 /*MC01*/
        0 /*MC02*/
    >;

    st,pkcs = <
        CLK_CKPER_HSE
        CLK_FMC_ACLK
        CLK_QSPI_ACLK
        CLK_ETH_DISABLED
        CLK_SDMMC12_PLL4P
        CLK_DSI_DSIPLL
        CLK_STGEN_HSE
        CLK_USBPHY_HSE
        CLK_SPI2S1_PLL3Q
        CLK_SPI2S23_PLL3Q
        CLK_SPI45_HSI
        CLK_SPI6_HSI
        CLK_I2C46_HSI
        CLK_SDMMC3_PLL4P
        CLK_USB0_USBPHY
        CLK_ADC_CKPER
        CLK_CEC_LSE
        CLK_I2C12_HSI
        CLK_I2C35_HSI
        CLK_UART1_HSI
        CLK_UART24_HSI
        CLK_UART35_HSI
        CLK_UART6_HSI
        CLK_UART78_HSI
        CLK_SPDIF_PLL4P
        CLK_FDCAN_PLL4R
        CLK_SAI1_PLL3Q
        CLK_SAI2_PLL3Q
        CLK_SAI3_PLL3Q
        CLK_SAI4_PLL3Q
        CLK_RNG1_LSI
        CLK_RNG2_LSI
        CLK_LPTIM1_PCLK1
        CLK_LPTIM23_PCLK3
        CLK_LPTIM45_LSE
    >;

    /* VCO = 1066.0 MHz => P = 266 (AXI), Q = 533 (GPU), R = 533 (DDR) */
    pll2: st,pll@1 {
        cfg = < 2 65 1 0 0 PQR(1,1,1) >;
        frac = < 0x1400 >;
        u-boot,dm-pre-reloc;
    };

    /* VCO = 417.8 MHz => P = 209, Q = 25, R = 11 */

```




```

pll3: st,pll@2 {
    cfg = < 1 33 1 16 36 PQR(1,1,1) >;
    frac = < 0x1a04 >;
    u-boot,dm-pre-reloc;
};

/* VCO = 594.0 MHz => P = 99, Q = 74, R = 74 */
pll4: st,pll@3 {
    cfg = < 3 98 5 7 7 PQR(1,1,1) >;
    u-boot,dm-pre-reloc;
};
    
```

3.2 STM32MP157x-DK2 case

This chapter shows the boot time clock tree set by the FSBL on STM32MP157x-DK2 DISCO board. Linux eventual runtime modifications are not covered here.

3.2.1 Clock tree

The following table shows what STM32MP157x-DK2 clock tree looks like, as a result of the boot chain execution with the device tree built with STM32CubeMX.

Component Comment	Parent	Frequency	Used?	
LSI	N.A.	0.032000 MHz	yes	Mandatory for IWDG, DAC
DAC	LSI	0.032000 MHz	no	
RNG1	LSI	0.032000 MHz	yes	Input frequency should be as low as possible
RNG2	LSI	0.032000 MHz	yes	Input frequency should be as low as possible
IWDG1	LSI	0.032000 MHz	yes	
IWDG2	LSI	0.032000 MHz	yes	
LSE	N.A.	0.032768 MHz	yes	Mandatory for DTS
RTC	LSE	0.032768 MHz	yes	
TAMP	LSE	0.032768 MHz	yes	
CEC	LSE	0.032768 MHz	yes	
LPTIM4	LSE	0.032768 MHz	yes	
LPTIM5	LSE	0.032768 MHz	yes	
HSI	N.A.	64.000000 MHz	yes	
SPI4	HSI	64.000000 MHz	yes	Arduino
SPI5	HSI	64.000000 MHz	yes	Rpi
SPI6	HSI	64.000000 MHz	no	
I2C4	HSI	64.000000 MHz	yes	PMIC
I2C6	HSI	64.000000 MHz	no	
I2C1	HSI	64.000000 MHz	yes	Rpi and peripherals
I2C2	HSI	64.000000 MHz	no	
I2C3	HSI	64.000000 MHz	no	
I2C5	HSI	64.000000 MHz	yes	Rpi and Arduino
USART1	HSI	64.000000 MHz	no	
USART2	HSI	64.000000 MHz	yes	Bluetooth
USART3	HSI	64.000000 MHz	yes	Rpi
UART4	HSI	64.000000 MHz	yes	Linux console
UART5	HSI	64.000000 MHz	no	
USART6	HSI	64.000000 MHz	no	
UART7	HSI	64.000000 MHz	yes	Arduino
UART8	HSI	64.000000 MHz	no	
MCO1	HSI	64.000000 MHz	no	Available so can be used
HSE	N.A.	24.000000 MHz	yes	
DSIPLL	HSE	125.000000 MHz	no	DSI DPHY PLL
DSIBL	DSIPLL	125.000000 MHz	no	DSI lanebyte clock



RTCDIV	HSE	1.000000 MHz	yes	Only used when RTC source
is HSE				
ck_per	HSE	24.000000 MHz	yes	
ADC	ck_per	24.000000 MHz	yes	Can use internal divider to
be < 40MHz				
PLL1	HSE	xxx MHz	yes	
PLL1P	PLL1	xxx MHz	yes	
MPUDIV	PLL1P	xxx MHz	yes	
Cortex-A7	PLL1P	xxx MHz	yes	650 MHz or 800 MHz, see the
information box above				
MC02	Cortex-A7	xxx MHz	no	Available so can be used
PLL2	HSE	1066.000000 MHz	yes	
PLL2P	PLL2	266.500000 MHz	yes	
AXI	PLL2P	266.500000 MHz	yes	< 266MHz
FMC	AXI	266.500000 MHz	no	
QSPI	AXI	266.500000 MHz	no	
SYSRAM	AXI	266.500000 MHz	yes	
ROM	AXI	266.500000 MHz	yes	
AHB5	AXI	266.500000 MHz	yes	< 266MHz
Cryp1	AHB5	266.500000 MHz	yes	
HASH1	AHB5	266.500000 MHz	yes	
GPIOZ	AHB5	266.500000 MHz	yes	
BKPSRAM	AHB5	266.500000 MHz	yes	
AHB6	AXI	266.500000 MHz	yes	< 266MHz
CRC1	AHB6	266.500000 MHz	yes	
MDMA	AHB6	266.500000 MHz	yes	
USBH	AHB6	266.500000 MHz	yes	USB Host
APB4	AHB6	133.250000 MHz	yes	
APB5	AHB6	66.625000 MHz	yes	
BSEC	APB5	66.625000 MHz	yes	< 67MHz
ETZPC	APB5	66.625000 MHz	yes	
TZC	APB5	66.625000 MHz	yes	
DBGAPB	AXI	133.250000 MHz	yes	JTAG & Coresight
DBGMCU	DBGAPB	66.625000 MHz	yes	
STM	DBGAPB	66.625000 MHz	no	
PLL2Q	PLL2	533.000000 MHz	yes	
GPU	PLL2Q	533.000000 MHz	yes	< 533MHz
PLL2R	PLL2	533.000000 MHz	yes	
DDR	PLL2R	533.000000 MHz	yes	< 533MHz
PLL3	HSE	417.755859 MHz	yes	
PLL3P	PLL3	208.877930 MHz	yes	
MLAHB	PLL3P	208.877930 MHz	yes	< 209MHz
Cortex-M4	MLAHB	208.877930 MHz	yes	
SRAM1	MLAHB	208.877930 MHz	yes	
SRAM2	MLAHB	208.877930 MHz	yes	
SRAM3	MLAHB	208.877930 MHz	yes	
RETRAM	MLAHB	208.877930 MHz	yes	
AHB1	MLAHB	208.877930 MHz	yes	< 209MHz
AHB2	MLAHB	208.877930 MHz	yes	< 209MHz
DMA1	AHB2	208.877930 MHz	yes	
DMA2	AHB2	208.877930 MHz	yes	
DMAMUX	AHB2	208.877930 MHz	yes	
APB1	MLAHB	104.438965 MHz	yes	
LPTIM1	APB1	104.438965 MHz	yes	
WWDG	APB1	104.438965 MHz	yes	
APB2	MLAHB	104.438965 MHz	yes	
TIM2	MLAHB	208.877930 MHz	yes	TIM Group 1
TIM3	MLAHB	208.877930 MHz	yes	TIM Group 1
TIM4	MLAHB	208.877930 MHz	yes	TIM Group 1
TIM5	MLAHB	208.877930 MHz	yes	TIM Group 1
TIM6	MLAHB	208.877930 MHz	yes	TIM Group 1
TIM7	MLAHB	208.877930 MHz	yes	TIM Group 1
TIM12	MLAHB	208.877930 MHz	yes	TIM Group 1
TIM13	MLAHB	208.877930 MHz	yes	TIM Group 1
TIM14	MLAHB	208.877930 MHz	yes	TIM Group 1
TIM1	MLAHB	208.877930 MHz	yes	TIM Group 2
TIM8	MLAHB	208.877930 MHz	yes	TIM Group 2



TIM15	MLAHB	208.877930	MHz	yes	TIM Group 2
TIM16	MLAHB	208.877930	MHz	yes	TIM Group 2
TIM17	MLAHB	208.877930	MHz	yes	TIM Group 2
APB3	MLAHB	104.438965	MHz	yes	
LPTIM2	APB3	104.438965	MHz	yes	
LPTIM3	APB3	104.438965	MHz	yes	
SYSCFG	APB3	104.438965	MHz	yes	
VREFBUF	APB3	104.438965	MHz	yes	
DTS	APB3	104.438965	MHz	yes	
HDP	APB3	104.438965	MHz	no	
AHB3	MLAHB	208.877930	MHz	yes	< 209MHz
CRC2	AHB3	208.877930	MHz	yes	
CRYP2	AHB3	208.877930	MHz	yes	
HASH2	AHB3	208.877930	MHz	yes	
DCMI	AHB3	208.877930	MHz	no	
IPCC	AHB3	208.877930	MHz	yes	Mailbox
AHB4	MLAHB	208.877930	MHz	yes	< 209MHz
PWR	AHB4	208.877930	MHz	yes	
RCC	AHB4	208.877930	MHz	yes	
GPIOA-K	AHB4	208.877930	MHz	yes	
EXTI	AHB4	208.877930	MHz	yes	
PLL3Q	PLL3	24.573874	MHz	yes	
SPI1	PLL3Q	24.573874	MHz	yes	SPI2S1 for BT PCM
SPI2	PLL3Q	24.573874	MHz	yes	SPI2S2 for HDMI
SPI3	PLL3Q	24.573874	MHz	no	
DFSDM	PLL3Q	24.573874	MHz	no	
SAI1	PLL3Q	24.573874	MHz	no	
SAI2	PLL3Q	24.573874	MHz	yes	AudCodec 48kHz (use PLL3R)
for 44.1kHz)					
SAI3	PLL3Q	24.573874	MHz	no	
SAI4	PLL3Q	24.573874	MHz	no	
PLL3R	PLL3	11.290699	MHz	yes	
PLL4	HSE	594.000000	MHz	yes	
PLL4P	PLL4	99.000000	MHz	yes	
SDMMC1	PLL4P	99.000000	MHz	yes	µSD card
SDMMC2	PLL4P	99.000000	MHz	yes	Wifi
SDMMC3	PLL4P	99.000000	MHz	yes	Rpi
SPDIF	PLL4P	99.000000	MHz	no	
PLL4Q	PLL4	74.250000	MHz	yes	
LCD	PLL4Q	74.250000	MHz	yes	LTDC & DSI display pixel
clock					
PLL4R	PLL4	74.250000	MHz	yes	
FDCAN	PLL4R	74.250000	MHz	no	
STGEN	HSE	24.000000	MHz	yes	
USBPHYC	HSE	24.000000	MHz	yes	USB PHY Ctrl for USB Host
and OTG					
USBPLL	USBPHYC	48.000000	MHz	yes	
USB0	USBPLL	48.000000	MHz	yes	USB OTG
CSI	N.A.	4.000000	MHz	yes	Mandatory for IO
compensation					
ETH	N.A.	0.000000	MHz	no	ETH clocked by RGMII PHY

3.2.2 Device tree

Here are the corresponding device tree `rcc_clk` sub node properties consumed by the first stage boot loader (FSBL) to configure the clock tree above:

```
st,clksrc = <
    CLK_MPU_PLL1P
    CLK_AXI_PLL2P
    CLK_MCU_PLL3P
    CLK_PLL12_HSE
```



```

        CLK_PLL3_HSE
        CLK_PLL4_HSE
        CLK_RTC_LSE
        CLK_MC01_DISABLED
        CLK_MC02_DISABLED
    >;

    st,clkdiv = <
        1 /*MPU*/
        0 /*AXI*/
        0 /*MCU*/
        1 /*APB1*/
        1 /*APB2*/
        1 /*APB3*/
        1 /*APB4*/
        2 /*APB5*/
        23 /*RTC*/
        0 /*MC01*/
        0 /*MC02*/
    >;

    st,pkcs = <
        CLK_CKPER_HSE
        CLK_FMC_ACLK
        CLK_QSPI_ACLK
        CLK_ETH_DISABLED
        CLK_SDMMC12_PLL4P
        CLK_DSI_DSIPLL
        CLK_STGEN_HSE
        CLK_USBPHY_HSE
        CLK_SPI2S1_PLL3Q
        CLK_SPI2S23_PLL3Q
        CLK_SPI45_HSI
        CLK_SPI6_HSI
        CLK_I2C46_HSI
        CLK_SDMMC3_PLL4P
        CLK_USB0_USBPHY
        CLK_ADC_CKPER
        CLK_CEC_LSE
        CLK_I2C12_HSI
        CLK_I2C35_HSI
        CLK_UART1_HSI
        CLK_UART24_HSI
        CLK_UART35_HSI
        CLK_UART6_HSI
        CLK_UART78_HSI
        CLK_SPDIF_PLL4P
        CLK_FDCAN_PLL4R
        CLK_SAI1_PLL3Q
        CLK_SAI2_PLL3Q
        CLK_SAI3_PLL3Q
        CLK_SAI4_PLL3Q
        CLK_RNG1_LSI
        CLK_RNG2_LSI
        CLK_LPTIM1_PCLK1
        CLK_LPTIM23_PCLK3
        CLK_LPTIM45_LSE
    >;

    /* VCO = 1066.0 MHz => P = 266 (AXI), Q = 533 (GPU), R = 533 (DDR) */
    pll2: st,pll@1 {
        cfg = < 2 65 1 0 0 PQR(1,1,1) >;
        frac = < 0x1400 >;
        u-boot,dm-pre-reloc;
    };

    /* VCO = 417.8 MHz => P = 209, Q = 25, R = 11 */

```



```

pll3: st,pll@2 {
    cfg = < 1 33 1 16 36 PQR(1,1,1) >;
    frac = < 0x1a04 >;
    u-boot,dm-pre-reloc;
};

/* VCO = 594.0 MHz => P = 99, Q = 74, R = 74 */
pll4: st,pll@3 {
    cfg = < 3 98 5 7 7 PQR(1,1,1) >;
    u-boot,dm-pre-reloc;
};

```

Reset and Clock Control

Low Speed Internal oscillator (STM32 clock source)

Low Speed External oscillator (STM32 clock source)

High Speed Internal oscillator (STM32 clock source) or High Speed Synchronous Serial Interface (MIPI® Alliance standard)

High Speed External oscillator (STM32 clock source)

Multi Speed Internal oscillator (STM32 clock source)

Cortex®

First Stage Boot Loader

Linux® is a registered trademark of Linus Torvalds.

Das U-Boot -- the Universal Boot Loader (see [U-Boot_overview](#))

Secondary Program Loader, *Also known as **U-Boot SPL***

Read Only Memory

Central processing unit

Dynamic voltage and frequency scaling

Independent Watchdog

Digital-to-analog converter (Electronic circuit that converts a binary number into a continuously varying value.)

Device Tree Source (in software context) or Digital Temperature Sensor (in peripheral context)

Real Time Clock

Tamper

Consumer Electronics Control (HDMI standard)

Power Management Integrated Circuit

Display Serial Interface (MIPI® Alliance standard)

Analog-to-digital converter. The process of converting a sampled analog signal to a digital code that represents the amplitude of the original signal sample.

USB Host (STM32 specific)

Boot and Security and OTP control

Extended TrustZone Protection Controller



TrustZone® address space Controller for DDR

Arm® and TrustZone® are registered trademarks of Arm Limited (or its subsidiaries) in the US and/or elsewhere.

debug and test protocol, named from the Joint Test Action Group that developed it

System Trace Module

Graphics Processing Units

Doubledata rate (memory domain)

System Configuration

voltage reference buffer (STM32 specific)

Hardware Debug Port

Digital Camera Memory Interface

Inter-Processor Communication Controller

External Interrupt

Digital Filter for Sigma-Delta Modulator

Transmit

former spelling for e•MMC ('e' in italic)

Receive

LCD TFT Display Controller (STM32 specific)

System Time Generator

USB On-The-Go (Capability/type of USB port, acting primarily as USB device, to also act as USB host. Also known as USB OTG.)

input/output

Ethernet

Microprocessor Unit

Microcontroller Unit (MCUs have internal flash memory and are intended to operate with a minimum amount of external support ICs. They commonly are a self-contained, system-on-chip (SoC) designs.)

Discovery kit

BlueTooth

High-Definition Multimedia Interface (HDMI standard)



Pages in category "STM32MP15 Evaluation boards"

The following 9 pages are in this category, out of 9 total.

0

- [STM32MP15 Evaluation boards - getting started](#)
- [STM32MP157x-EV1 - hardware description](#)
- [STM32MP15 Evaluation boards - Starter Package](#)

L

- [LEDs and buttons on STM32 MPU boards](#)

M

- [MB1230](#)
- [MB1262](#)
- [MB1263](#)
- [MB1379](#)

S

- [STM32MP15 Evaluation boards - Starter Package for Android](#)

Stable: 24.06.2020 - 09:44 / Revision: 23.06.2020 - 13:56

All the peripherals receive one or several clocks that are generated via [RCC internal peripheral](#). RCC relies on several clock sources (LSI, LSE, HSI, HSE, CSI) and four PLL in order to provide adequate input frequencies to all the peripherals. The clock tree covers all the system clock distribution aspects, from the clock sources to the consumer peripherals (internal and external), except clock gating management that is locally controlled by each peripheral driver.

Contents

1 Overview	56
2 How to build a clock tree?	57
3 ST boards clock tree	58
3.1 STM32MP157x-EV1 case	58
3.1.1 Clock tree	58
3.1.1.1 Device tree	60
3.2 STM32MP157x-DK2 case	62
3.2.1 Clock tree	62
3.2.2 Device tree	64



1 Overview

The clock tree is managed via RCC internal peripheral hardware block and it is configured at different steps from the Cortex-A7:

- When the device is **reset**, all RCC registers take their reset values: the four PLL are disabled and most of the clock source selectors are pointing to the HSI.
- The ROM Code configures the minimum clock tree needed to boot on the selected boot device.
- The **FSBL** (Cf. [Boot chain overview](#)) completely configures the clock tree as expected, all the way up to Linux, thanks to the configuration given in the [device tree](#).
 - U-Boot SPL (basic boot chain) binding is available in [doc/device-tree-bindings/clock/st,stm32mp1.txt](#)
 - TF-A (trusted boot chain) binding follows the same syntax as U-Boot SPL
- **Linux** is responsible for the clock tree at runtime and it may partly modify it thanks to information taken from the [device tree](#):
 - Linux binding is available in [Documentation/devicetree/bindings/clock/clock-bindings.txt](#) (and surrounding files): 'fixed-clock' compatible, 'clocks' and 'assigned-clocks' properties are important concepts to understand the management of clocks providers/consumers.

Notice that [STM32Cube](#) (running on the Cortex-M4) does not control the clock tree setup so the configuration applied by the Cortex-A7 must provide suitable clocks to Cortex-M4, like it is explained in [resource manager](#) description. One exception to this rule is the [engineering boot](#), allowing to directly load (and debug) the Cortex-M4 whereas the Cortex-A7 execution is stalled in the ROM code.



2 How to build a clock tree?

Building a clock tree is quite complex as it needs to take into account the constraints set by each internal and external peripheral, including external clock sources.

We encourage the use of [STM32CubeMX](#) to build the clock tree, and avoid having to know all internal peripherals details: the tool allows to select the peripherals that will be present on the board, fix the clock sources frequencies and automatically find an optimized clock tree. It is then able to generate the device tree that is directly consumed by the boot chain and Linux.



3 ST boards clock tree

 See [How to change the CPU frequency](#) article for more information about the CPU frequency setting (PLL1P), including dynamic voltage and frequency scaling (DVFS) configuration.

3.1 STM32MP157x-EV1 case

This chapter shows the boot time clock tree set by the FSBL on STM32MP157x-EV1 evaluation board. Linux eventual runtime modifications are not covered here.

3.1.1 Clock tree

The following table shows what STM32MP157x-EV1 clock tree looks like, as a result of the boot chain execution with the device tree built with STM32CubeMX.

Component Comment	Parent	Frequency	Used?	
LSI	N.A.	0.032000 MHz	yes	Mandatory for IWDG, DAC
DAC	LSI	0.032000 MHz	yes	
RNG1	LSI	0.032000 MHz	yes	Input frequency should be
as low as possible				
RNG2	LSI	0.032000 MHz	yes	Input frequency should be
as low as possible				
IWDG1	LSI	0.032000 MHz	yes	
IWDG2	LSI	0.032000 MHz	yes	
LSE	N.A.	0.032768 MHz	yes	Mandatory for DTS
RTC	LSE	0.032768 MHz	yes	
TAMP	LSE	0.032768 MHz	yes	
CEC	LSE	0.032768 MHz	yes	
LPTIM4	LSE	0.032768 MHz	yes	
LPTIM5	LSE	0.032768 MHz	yes	
HSI	N.A.	64.000000 MHz	yes	
SPI4	HSI	64.000000 MHz	no	
SPI5	HSI	64.000000 MHz	no	
SPI6	HSI	64.000000 MHz	yes	
I2C4	HSI	64.000000 MHz	yes	PMIC
I2C6	HSI	64.000000 MHz	no	
I2C1	HSI	64.000000 MHz	no	
I2C2	HSI	64.000000 MHz	yes	Rpi and peripherals
I2C3	HSI	64.000000 MHz	no	
I2C5	HSI	64.000000 MHz	yes	Rpi
USART1	HSI	64.000000 MHz	yes	
USART2	HSI	64.000000 MHz	no	
USART3	HSI	64.000000 MHz	yes	Rpi
UART4	HSI	64.000000 MHz	yes	Linux console
UART5	HSI	64.000000 MHz	no	
USART6	HSI	64.000000 MHz	no	
UART7	HSI	64.000000 MHz	no	
UART8	HSI	64.000000 MHz	no	
MC01	HSI	64.000000 MHz	no	Available so can be used
HSE	N.A.	24.000000 MHz	yes	
DSIPLL	HSE	125.000000 MHz	no	DSI DPHY PLL
DSIBL	DSIPLL	125.000000 MHz	no	DSI lanebyte clock
RTCDIV	HSE	1.000000 MHz	yes	Only used when RTC source
is HSE				



ck_per	HSE	24.000000	MHz	yes	
ADC	ck_per	24.000000	MHz	yes	Can use internal divider to
be < 40MHz					
PLL1	HSE	xxx	MHz	yes	
PLL1P	PLL1	xxx	MHz	yes	
MPUDIV	PLL1P	xxx	MHz	yes	
Cortex-A7	PLL1P	xxx	MHz	yes	650 MHz or 800 MHz, see the
information box above					
MC02	Cortex-A7	xxx	MHz	no	Available so can be used
PLL2	HSE	1066.000000	MHz	yes	
PLL2P	PLL2	266.500000	MHz	yes	
AXI	PLL2P	266.500000	MHz	yes	< 266MHz
FMC	AXI	266.500000	MHz	yes	NAND flash
QSPI	AXI	266.500000	MHz	yes	NOR flash
SYSRAM	AXI	266.500000	MHz	yes	
ROM	AXI	266.500000	MHz	yes	
AHB5	AXI	266.500000	MHz	yes	< 266MHz
CRYP1	AHB5	266.500000	MHz	yes	
HASH1	AHB5	266.500000	MHz	yes	
GPI0Z	AHB5	266.500000	MHz	yes	
BKPSRAM	AHB5	266.500000	MHz	yes	
AHB6	AXI	266.500000	MHz	yes	< 266MHz
CRC1	AHB6	266.500000	MHz	yes	
MDMA	AHB6	266.500000	MHz	yes	
USBH	AHB6	266.500000	MHz	yes	USB Host
APB4	AHB6	133.250000	MHz	yes	
APB5	AHB6	66.625000	MHz	yes	
BSEC	APB5	66.625000	MHz	yes	< 67MHz
ETZPC	APB5	66.625000	MHz	yes	
TZC	APB5	66.625000	MHz	yes	
DBGAPB	AXI	133.250000	MHz	yes	JTAG & Coresight
DBGMCU	DBGAPB	66.625000	MHz	yes	
STM	DBGAPB	66.625000	MHz	yes	
PLL2Q	PLL2	533.000000	MHz	yes	
GPU	PLL2Q	533.000000	MHz	yes	< 533MHz
PLL2R	PLL2	533.000000	MHz	yes	
DDR	PLL2R	533.000000	MHz	yes	< 533MHz
PLL3	HSE	417.755859	MHz	yes	
PLL3P	PLL3	208.877930	MHz	yes	
MLAHB	PLL3P	208.877930	MHz	yes	< 209MHz
Cortex-M4	MLAHB	208.877930	MHz	yes	
SRAM1	MLAHB	208.877930	MHz	yes	
SRAM2	MLAHB	208.877930	MHz	yes	
SRAM3	MLAHB	208.877930	MHz	yes	
RETRAM	MLAHB	208.877930	MHz	yes	
AHB1	MLAHB	208.877930	MHz	yes	< 209MHz
AHB2	MLAHB	208.877930	MHz	yes	< 209MHz
DMA1	AHB2	208.877930	MHz	yes	
DMA2	AHB2	208.877930	MHz	yes	
DMAMUX	AHB2	208.877930	MHz	yes	
APB1	MLAHB	104.438965	MHz	yes	
LPTIM1	APB1	104.438965	MHz	yes	
WWDG	APB1	104.438965	MHz	yes	
APB2	MLAHB	104.438965	MHz	yes	
TIM2	MLAHB	208.877930	MHz	yes	TIM Group 1
TIM3	MLAHB	208.877930	MHz	yes	TIM Group 1
TIM4	MLAHB	208.877930	MHz	yes	TIM Group 1
TIM5	MLAHB	208.877930	MHz	yes	TIM Group 1
TIM6	MLAHB	208.877930	MHz	yes	TIM Group 1
TIM7	MLAHB	208.877930	MHz	yes	TIM Group 1
TIM12	MLAHB	208.877930	MHz	yes	TIM Group 1
TIM13	MLAHB	208.877930	MHz	yes	TIM Group 1
TIM14	MLAHB	208.877930	MHz	yes	TIM Group 1
TIM1	MLAHB	208.877930	MHz	yes	TIM Group 2
TIM8	MLAHB	208.877930	MHz	yes	TIM Group 2
TIM15	MLAHB	208.877930	MHz	yes	TIM Group 2
TIM16	MLAHB	208.877930	MHz	yes	TIM Group 2



TIM17	MLAHB	208.877930	MHz	yes	TIM Group 2
APB3	MLAHB	104.438965	MHz	yes	
LPTIM2	APB3	104.438965	MHz	yes	
LPTIM3	APB3	104.438965	MHz	yes	
SYSCFG	APB3	104.438965	MHz	yes	
VREFBUF	APB3	104.438965	MHz	yes	
DTS	APB3	104.438965	MHz	yes	
HDP	APB3	104.438965	MHz	yes	
AHB3	MLAHB	208.877930	MHz	yes	< 209MHz
CRC2	AHB3	208.877930	MHz	yes	
CRYP2	AHB3	208.877930	MHz	yes	
HASH2	AHB3	208.877930	MHz	yes	
DCMI	AHB3	208.877930	MHz	yes	Camera
IPCC	AHB3	208.877930	MHz	yes	Mailbox
AHB4	MLAHB	208.877930	MHz	yes	< 209MHz
PWR	AHB4	208.877930	MHz	yes	
RCC	AHB4	208.877930	MHz	yes	
GPIOA-K	AHB4	208.877930	MHz	yes	
EXTI	AHB4	208.877930	MHz	yes	
PLL3Q	PLL3	24.573874	MHz	yes	
SPI1	PLL3Q	24.573874	MHz	no	
SPI2	PLL3Q	24.573874	MHz	no	
SPI3	PLL3Q	24.573874	MHz	no	
DFSDM	PLL3Q	24.573874	MHz	yes	Digital micro
SAI1	PLL3Q	24.573874	MHz	no	
SAI2	PLL3Q	24.573874	MHz	yes	AudCodec 48kHz (use PLL3R
for 44.1kHz)					
SAI3	PLL3Q	24.573874	MHz	no	
SAI4	PLL3Q	24.573874	MHz	yes	SPDIF TX 48kHz (use PLL3R
for 44.1kHz)					
PLL3R	PLL3	11.290699	MHz	yes	
PLL4	HSE	594.000000	MHz	yes	
PLL4P	PLL4	99.000000	MHz	yes	
SDMMC1	PLL4P	99.000000	MHz	yes	µSD card
SDMMC2	PLL4P	99.000000	MHz	yes	eMMC
SDMMC3	PLL4P	99.000000	MHz	yes	
SPDIF	PLL4P	99.000000	MHz	yes	SPDIF RX
PLL4Q	PLL4	74.250000	MHz	yes	
LCD	PLL4Q	74.250000	MHz	yes	LTDC & DSI display pixel
clock					
PLL4R	PLL4	74.250000	MHz	yes	
FDCAN	PLL4R	74.250000	MHz	yes	Should be as high as
possible and < 100MHz					
STGEN	HSE	24.000000	MHz	yes	
USBPHYC	HSE	24.000000	MHz	yes	USB PHY Ctrl for USB Host
and OTG					
USBPLL	USBPHYC	48.000000	MHz	yes	
USB0	USBPLL	48.000000	MHz	yes	USB OTG
CSI	N.A.	4.000000	MHz	yes	Mandatory for IO
compensation					
ETH	N.A.	0.000000	MHz	no	ETH clocked by RGMII PHY

3.1.1.1 Device tree

Here are the corresponding device tree `rcc_clk` sub node properties consumed by the first stage boot loader (FSBL) to configure the clock tree above:

```
st,clksrc = <
    CLK_MPU_PLL1P
    CLK_AXI_PLL2P
    CLK_MCU_PLL3P
    CLK_PLL12_HSE
```



```

        CLK_PLL3_HSE
        CLK_PLL4_HSE
        CLK_RTC_LSE
        CLK_MC01_DISABLED
        CLK_MC02_DISABLED
    >;

    st,clkdiv = <
        1 /*MPU*/
        0 /*AXI*/
        0 /*MCU*/
        1 /*APB1*/
        1 /*APB2*/
        1 /*APB3*/
        1 /*APB4*/
        2 /*APB5*/
        23 /*RTC*/
        0 /*MC01*/
        0 /*MC02*/
    >;

    st,pkcs = <
        CLK_CKPER_HSE
        CLK_FMC_ACLK
        CLK_QSPI_ACLK
        CLK_ETH_DISABLED
        CLK_SDMMC12_PLL4P
        CLK_DSI_DSIPLL
        CLK_STGEN_HSE
        CLK_USBPHY_HSE
        CLK_SPI2S1_PLL3Q
        CLK_SPI2S23_PLL3Q
        CLK_SPI45_HSI
        CLK_SPI6_HSI
        CLK_I2C46_HSI
        CLK_SDMMC3_PLL4P
        CLK_USB0_USBPHY
        CLK_ADC_CKPER
        CLK_CEC_LSE
        CLK_I2C12_HSI
        CLK_I2C35_HSI
        CLK_UART1_HSI
        CLK_UART24_HSI
        CLK_UART35_HSI
        CLK_UART6_HSI
        CLK_UART78_HSI
        CLK_SPDIF_PLL4P
        CLK_FDCAN_PLL4R
        CLK_SAI1_PLL3Q
        CLK_SAI2_PLL3Q
        CLK_SAI3_PLL3Q
        CLK_SAI4_PLL3Q
        CLK_RNG1_LSI
        CLK_RNG2_LSI
        CLK_LPTIM1_PCLK1
        CLK_LPTIM23_PCLK3
        CLK_LPTIM45_LSE
    >;

    /* VCO = 1066.0 MHz => P = 266 (AXI), Q = 533 (GPU), R = 533 (DDR) */
    pll2: st,pll@1 {
        cfg = < 2 65 1 0 0 PQR(1,1,1) >;
        frac = < 0x1400 >;
        u-boot,dm-pre-reloc;
    };

    /* VCO = 417.8 MHz => P = 209, Q = 25, R = 11 */

```



```

pll3: st,pll@2 {
    cfg = < 1 33 1 16 36 PQR(1,1,1) >;
    frac = < 0x1a04 >;
    u-boot,dm-pre-reloc;
};

/* VCO = 594.0 MHz => P = 99, Q = 74, R = 74 */
pll4: st,pll@3 {
    cfg = < 3 98 5 7 7 PQR(1,1,1) >;
    u-boot,dm-pre-reloc;
};
    
```

3.2 STM32MP157x-DK2 case

This chapter shows the boot time clock tree set by the FSBL on STM32MP157x-DK2 DISCO board. Linux eventual runtime modifications are not covered here.

3.2.1 Clock tree

The following table shows what STM32MP157x-DK2 clock tree looks like, as a result of the boot chain execution with the device tree built with STM32CubeMX.

Component Comment	Parent	Frequency	Used?	
LSI	N.A.	0.032000 MHz	yes	Mandatory for IWDG, DAC
DAC	LSI	0.032000 MHz	no	
RNG1	LSI	0.032000 MHz	yes	Input frequency should be as low as possible
RNG2	LSI	0.032000 MHz	yes	Input frequency should be as low as possible
IWDG1	LSI	0.032000 MHz	yes	
IWDG2	LSI	0.032000 MHz	yes	
LSE	N.A.	0.032768 MHz	yes	Mandatory for DTS
RTC	LSE	0.032768 MHz	yes	
TAMP	LSE	0.032768 MHz	yes	
CEC	LSE	0.032768 MHz	yes	
LPTIM4	LSE	0.032768 MHz	yes	
LPTIM5	LSE	0.032768 MHz	yes	
HSI	N.A.	64.000000 MHz	yes	
SPI4	HSI	64.000000 MHz	yes	Arduino
SPI5	HSI	64.000000 MHz	yes	Rpi
SPI6	HSI	64.000000 MHz	no	
I2C4	HSI	64.000000 MHz	yes	PMIC
I2C6	HSI	64.000000 MHz	no	
I2C1	HSI	64.000000 MHz	yes	Rpi and peripherals
I2C2	HSI	64.000000 MHz	no	
I2C3	HSI	64.000000 MHz	no	
I2C5	HSI	64.000000 MHz	yes	Rpi and Arduino
USART1	HSI	64.000000 MHz	no	
USART2	HSI	64.000000 MHz	yes	Bluetooth
USART3	HSI	64.000000 MHz	yes	Rpi
UART4	HSI	64.000000 MHz	yes	Linux console
UART5	HSI	64.000000 MHz	no	
USART6	HSI	64.000000 MHz	no	
UART7	HSI	64.000000 MHz	yes	Arduino
UART8	HSI	64.000000 MHz	no	
MCO1	HSI	64.000000 MHz	no	Available so can be used
HSE	N.A.	24.000000 MHz	yes	
DSIPLL	HSE	125.000000 MHz	no	DSI DPHY PLL
DSIBL	DSIPLL	125.000000 MHz	no	DSI lanebyte clock



RTCDIV	HSE	1.000000 MHz	yes	Only used when RTC source
is HSE				
ck_per	HSE	24.000000 MHz	yes	
ADC	ck_per	24.000000 MHz	yes	Can use internal divider to
be < 40MHz				
PLL1	HSE	xxx MHz	yes	
PLL1P	PLL1	xxx MHz	yes	
MPUDIV	PLL1P	xxx MHz	yes	
Cortex-A7	PLL1P	xxx MHz	yes	650 MHz or 800 MHz, see the
information box above				
MC02	Cortex-A7	xxx MHz	no	Available so can be used
PLL2	HSE	1066.000000 MHz	yes	
PLL2P	PLL2	266.500000 MHz	yes	
AXI	PLL2P	266.500000 MHz	yes	< 266MHz
FMC	AXI	266.500000 MHz	no	
QSPI	AXI	266.500000 MHz	no	
SYSRAM	AXI	266.500000 MHz	yes	
ROM	AXI	266.500000 MHz	yes	
AHB5	AXI	266.500000 MHz	yes	< 266MHz
Cryp1	AHB5	266.500000 MHz	yes	
HASH1	AHB5	266.500000 MHz	yes	
GPIOZ	AHB5	266.500000 MHz	yes	
BKPSRAM	AHB5	266.500000 MHz	yes	
AHB6	AXI	266.500000 MHz	yes	< 266MHz
CRC1	AHB6	266.500000 MHz	yes	
MDMA	AHB6	266.500000 MHz	yes	
USBH	AHB6	266.500000 MHz	yes	USB Host
APB4	AHB6	133.250000 MHz	yes	
APB5	AHB6	66.625000 MHz	yes	
BSEC	APB5	66.625000 MHz	yes	< 67MHz
ETZPC	APB5	66.625000 MHz	yes	
TZC	APB5	66.625000 MHz	yes	
DBGAPB	AXI	133.250000 MHz	yes	JTAG & Coresight
DBGMCU	DBGAPB	66.625000 MHz	yes	
STM	DBGAPB	66.625000 MHz	no	
PLL2Q	PLL2	533.000000 MHz	yes	
GPU	PLL2Q	533.000000 MHz	yes	< 533MHz
PLL2R	PLL2	533.000000 MHz	yes	
DDR	PLL2R	533.000000 MHz	yes	< 533MHz
PLL3	HSE	417.755859 MHz	yes	
PLL3P	PLL3	208.877930 MHz	yes	
MLAHB	PLL3P	208.877930 MHz	yes	< 209MHz
Cortex-M4	MLAHB	208.877930 MHz	yes	
SRAM1	MLAHB	208.877930 MHz	yes	
SRAM2	MLAHB	208.877930 MHz	yes	
SRAM3	MLAHB	208.877930 MHz	yes	
RETRAM	MLAHB	208.877930 MHz	yes	
AHB1	MLAHB	208.877930 MHz	yes	< 209MHz
AHB2	MLAHB	208.877930 MHz	yes	< 209MHz
DMA1	AHB2	208.877930 MHz	yes	
DMA2	AHB2	208.877930 MHz	yes	
DMAMUX	AHB2	208.877930 MHz	yes	
APB1	MLAHB	104.438965 MHz	yes	
LPTIM1	APB1	104.438965 MHz	yes	
WWDG	APB1	104.438965 MHz	yes	
APB2	MLAHB	104.438965 MHz	yes	
TIM2	MLAHB	208.877930 MHz	yes	TIM Group 1
TIM3	MLAHB	208.877930 MHz	yes	TIM Group 1
TIM4	MLAHB	208.877930 MHz	yes	TIM Group 1
TIM5	MLAHB	208.877930 MHz	yes	TIM Group 1
TIM6	MLAHB	208.877930 MHz	yes	TIM Group 1
TIM7	MLAHB	208.877930 MHz	yes	TIM Group 1
TIM12	MLAHB	208.877930 MHz	yes	TIM Group 1
TIM13	MLAHB	208.877930 MHz	yes	TIM Group 1
TIM14	MLAHB	208.877930 MHz	yes	TIM Group 1
TIM1	MLAHB	208.877930 MHz	yes	TIM Group 2
TIM8	MLAHB	208.877930 MHz	yes	TIM Group 2



TIM15	MLAHB	208.877930	MHz	yes	TIM Group 2
TIM16	MLAHB	208.877930	MHz	yes	TIM Group 2
TIM17	MLAHB	208.877930	MHz	yes	TIM Group 2
APB3	MLAHB	104.438965	MHz	yes	
LPTIM2	APB3	104.438965	MHz	yes	
LPTIM3	APB3	104.438965	MHz	yes	
SYSCFG	APB3	104.438965	MHz	yes	
VREFBUF	APB3	104.438965	MHz	yes	
DTS	APB3	104.438965	MHz	yes	
HDP	APB3	104.438965	MHz	no	
AHB3	MLAHB	208.877930	MHz	yes	< 209MHz
CRC2	AHB3	208.877930	MHz	yes	
CRYP2	AHB3	208.877930	MHz	yes	
HASH2	AHB3	208.877930	MHz	yes	
DCMI	AHB3	208.877930	MHz	no	
IPCC	AHB3	208.877930	MHz	yes	Mailbox
AHB4	MLAHB	208.877930	MHz	yes	< 209MHz
PWR	AHB4	208.877930	MHz	yes	
RCC	AHB4	208.877930	MHz	yes	
GPIOA-K	AHB4	208.877930	MHz	yes	
EXTI	AHB4	208.877930	MHz	yes	
PLL3Q	PLL3	24.573874	MHz	yes	
SPI1	PLL3Q	24.573874	MHz	yes	SPI2S1 for BT PCM
SPI2	PLL3Q	24.573874	MHz	yes	SPI2S2 for HDMI
SPI3	PLL3Q	24.573874	MHz	no	
DFSDM	PLL3Q	24.573874	MHz	no	
SAI1	PLL3Q	24.573874	MHz	no	
SAI2	PLL3Q	24.573874	MHz	yes	AudCodec 48kHz (use PLL3R)
for 44.1kHz)					
SAI3	PLL3Q	24.573874	MHz	no	
SAI4	PLL3Q	24.573874	MHz	no	
PLL3R	PLL3	11.290699	MHz	yes	
PLL4	HSE	594.000000	MHz	yes	
PLL4P	PLL4	99.000000	MHz	yes	
SDMMC1	PLL4P	99.000000	MHz	yes	µSD card
SDMMC2	PLL4P	99.000000	MHz	yes	Wifi
SDMMC3	PLL4P	99.000000	MHz	yes	Rpi
SPDIF	PLL4P	99.000000	MHz	no	
PLL4Q	PLL4	74.250000	MHz	yes	
LCD	PLL4Q	74.250000	MHz	yes	LTDC & DSI display pixel
clock					
PLL4R	PLL4	74.250000	MHz	yes	
FDCAN	PLL4R	74.250000	MHz	no	
STGEN	HSE	24.000000	MHz	yes	
USBPHYC	HSE	24.000000	MHz	yes	USB PHY Ctrl for USB Host
and OTG					
USBPLL	USBPHYC	48.000000	MHz	yes	
USB0	USBPLL	48.000000	MHz	yes	USB OTG
CSI	N.A.	4.000000	MHz	yes	Mandatory for IO
compensation					
ETH	N.A.	0.000000	MHz	no	ETH clocked by RGMII PHY

3.2.2 Device tree

Here are the corresponding device tree `rcc_clk` sub node properties consumed by the first stage boot loader (FSBL) to configure the clock tree above:

```
st,clksrc = <
    CLK_MPU_PLL1P
    CLK_AXI_PLL2P
    CLK_MCU_PLL3P
    CLK_PLL12_HSE
```




```

        CLK_PLL3_HSE
        CLK_PLL4_HSE
        CLK_RTC_LSE
        CLK_MC01_DISABLED
        CLK_MC02_DISABLED
    >;

    st,clkdiv = <
        1 /*MPU*/
        0 /*AXI*/
        0 /*MCU*/
        1 /*APB1*/
        1 /*APB2*/
        1 /*APB3*/
        1 /*APB4*/
        2 /*APB5*/
        23 /*RTC*/
        0 /*MC01*/
        0 /*MC02*/
    >;

    st,pkcs = <
        CLK_CKPER_HSE
        CLK_FMC_ACLK
        CLK_QSPI_ACLK
        CLK_ETH_DISABLED
        CLK_SDMMC12_PLL4P
        CLK_DSI_DSIPLL
        CLK_STGEN_HSE
        CLK_USBPHY_HSE
        CLK_SPI2S1_PLL3Q
        CLK_SPI2S23_PLL3Q
        CLK_SPI45_HSI
        CLK_SPI6_HSI
        CLK_I2C46_HSI
        CLK_SDMMC3_PLL4P
        CLK_USB0_USBPHY
        CLK_ADC_CKPER
        CLK_CEC_LSE
        CLK_I2C12_HSI
        CLK_I2C35_HSI
        CLK_UART1_HSI
        CLK_UART24_HSI
        CLK_UART35_HSI
        CLK_UART6_HSI
        CLK_UART78_HSI
        CLK_SPDIF_PLL4P
        CLK_FDCAN_PLL4R
        CLK_SAI1_PLL3Q
        CLK_SAI2_PLL3Q
        CLK_SAI3_PLL3Q
        CLK_SAI4_PLL3Q
        CLK_RNG1_LSI
        CLK_RNG2_LSI
        CLK_LPTIM1_PCLK1
        CLK_LPTIM23_PCLK3
        CLK_LPTIM45_LSE
    >;

    /* VCO = 1066.0 MHz => P = 266 (AXI), Q = 533 (GPU), R = 533 (DDR) */
    pll2: st,pll@1 {
        cfg = < 2 65 1 0 0 PQR(1,1,1) >;
        frac = < 0x1400 >;
        u-boot,dm-pre-reloc;
    };

    /* VCO = 417.8 MHz => P = 209, Q = 25, R = 11 */

```



```

pll3: st,pll@2 {
    cfg = < 1 33 1 16 36 PQR(1,1,1) >;
    frac = < 0x1a04 >;
    u-boot,dm-pre-reloc;
};

/* VCO = 594.0 MHz => P = 99, Q = 74, R = 74 */
pll4: st,pll@3 {
    cfg = < 3 98 5 7 7 PQR(1,1,1) >;
    u-boot,dm-pre-reloc;
};

```

Reset and Clock Control

Low Speed Internal oscillator (STM32 clock source)

Low Speed External oscillator (STM32 clock source)

High Speed Internal oscillator (STM32 clock source) or High Speed Synchronous Serial Interface (MIPI® Alliance standard)

High Speed External oscillator (STM32 clock source)

Multi Speed Internal oscillator (STM32 clock source)

Cortex®

First Stage Boot Loader

Linux® is a registered trademark of Linus Torvalds.

Das U-Boot -- the Universal Boot Loader (see [U-Boot_overview](#))

Secondary Program Loader, *Also known as **U-Boot SPL***

Read Only Memory

Central processing unit

Dynamic voltage and frequency scaling

Independent Watchdog

Digital-to-analog converter (Electronic circuit that converts a binary number into a continuously varying value.)

Device Tree Source (in software context) or Digital Temperature Sensor (in peripheral context)

Real Time Clock

Tamper

Consumer Electronics Control (HDMI standard)

Power Management Integrated Circuit

Display Serial Interface (MIPI® Alliance standard)

Analog-to-digital converter. The process of converting a sampled analog signal to a digital code that represents the amplitude of the original signal sample.

USB Host (STM32 specific)

Boot and Security and OTP control

Extended TrustZone Protection Controller



TrustZone® address space Controller for DDR

Arm® and TrustZone® are registered trademarks of Arm Limited (or its subsidiaries) in the US and/or elsewhere.

debug and test protocol, named from the Joint Test Action Group that developed it

System Trace Module

Graphics Processing Units

Doubledata rate (memory domain)

System Configuration

voltage reference buffer (STM32 specific)

Hardware Debug Port

Digital Camera Memory Interface

Inter-Processor Communication Controller

External Interrupt

Digital Filter for Sigma-Delta Modulator

Transmit

former spelling for e•MMC ('e' in italic)

Receive

LCD TFT Display Controller (STM32 specific)

System Time Generator

USB On-The-Go (Capability/type of USB port, acting primarily as USB device, to also act as USB host. Also known as USB OTG.)

input/output

Ethernet

Microprocessor Unit

Microcontroller Unit (MCUs have internal flash memory and are intended to operate with a minimum amount of external support ICs. They commonly are a self-contained, system-on-chip (SoC) designs.)

Discovery kit

BlueTooth

High-Definition Multimedia Interface (HDMI standard)

Stable: 25.09.2020 - 09:10 / Revision: 25.09.2020 - 09:09

All the peripherals receive one or several clocks that are generated via [RCC internal peripheral](#). RCC relies on several clocks sources (LSI, LSE, HSI, HSE, CSI) and four PLL in order to provide adequate input frequencies to all the peripherals. The clock tree covers all the system clock distribution aspects, from the clock sources to the consumer peripherals (internal and external), except clock gating management that is locally controlled by each peripheral driver.

Contents

1 Overview	69
2 How to build a clock tree?	70



3 ST boards clock tree	71
3.1 STM32MP157x-EV1 case	71
3.1.1 Clock tree	71
3.1.1.1 Device tree	73
3.2 STM32MP157x-DK2 case	75
3.2.1 Clock tree	75
3.2.2 Device tree	77



1 Overview

The clock tree is managed via RCC internal peripheral hardware block and it is configured at different steps from the Cortex-A7:

- When the device is **reset**, all RCC registers take their reset values: the four PLL are disabled and most of the clock source selectors are pointing to the HSI.
- The ROM Code configures the minimum clock tree needed to boot on the selected boot device.
- The **FSBL** (Cf. [Boot chain overview](#)) completely configures the clock tree as expected, all the way up to Linux, thanks to the configuration given in the [device tree](#).
 - U-Boot SPL (basic boot chain) binding is available in [doc/device-tree-bindings/clock/st,stm32mp1.txt](#)
 - TF-A (trusted boot chain) binding follows the same syntax as U-Boot SPL
- **Linux** is responsible for the clock tree at runtime and it may partly modify it thanks to information taken from the [device tree](#):
 - Linux binding is available in [Documentation/devicetree/bindings/clock/clock-bindings.txt](#) (and surrounding files): 'fixed-clock' compatible, 'clocks' and 'assigned-clocks' properties are important concepts to understand the management of clocks providers/consumers.

Notice that [STM32Cube](#) (running on the Cortex-M4) does not control the clock tree setup so the configuration applied by the Cortex-A7 must provide suitable clocks to Cortex-M4, like it is explained in [resource manager](#) description. One exception to this rule is the [engineering boot](#), allowing to directly load (and debug) the Cortex-M4 whereas the Cortex-A7 execution is stalled in the ROM code.



2 How to build a clock tree?

Building a clock tree is quite complex as it needs to take into account the constraints set by each internal and external peripheral, including external clock sources.

We encourage the use of [STM32CubeMX](#) to build the clock tree, and avoid having to know all internal peripherals details: the tool allows to select the peripherals that will be present on the board, fix the clock sources frequencies and automatically find an optimized clock tree. It is then able to generate the device tree that is directly consumed by the boot chain and Linux.



3 ST boards clock tree

 See [How to change the CPU frequency](#) article for more information about the CPU frequency setting (PLL1P), including dynamic voltage and frequency scaling (DVFS) configuration.

3.1 STM32MP157x-EV1 case

This chapter shows the boot time clock tree set by the FSBL on STM32MP157x-EV1 evaluation board. Linux eventual runtime modifications are not covered here.

3.1.1 Clock tree

The following table shows what STM32MP157x-EV1 clock tree looks like, as a result of the boot chain execution with the device tree built with STM32CubeMX.

Component Comment	Parent	Frequency	Used?	
LSI	N.A.	0.032000 MHz	yes	Mandatory for IWDG, DAC
DAC	LSI	0.032000 MHz	yes	
RNG1	LSI	0.032000 MHz	yes	Input frequency should be
as low as possible				
RNG2	LSI	0.032000 MHz	yes	Input frequency should be
as low as possible				
IWDG1	LSI	0.032000 MHz	yes	
IWDG2	LSI	0.032000 MHz	yes	
LSE	N.A.	0.032768 MHz	yes	Mandatory for DTS
RTC	LSE	0.032768 MHz	yes	
TAMP	LSE	0.032768 MHz	yes	
CEC	LSE	0.032768 MHz	yes	
LPTIM4	LSE	0.032768 MHz	yes	
LPTIM5	LSE	0.032768 MHz	yes	
HSI	N.A.	64.000000 MHz	yes	
SPI4	HSI	64.000000 MHz	no	
SPI5	HSI	64.000000 MHz	no	
SPI6	HSI	64.000000 MHz	yes	
I2C4	HSI	64.000000 MHz	yes	PMIC
I2C6	HSI	64.000000 MHz	no	
I2C1	HSI	64.000000 MHz	no	
I2C2	HSI	64.000000 MHz	yes	Rpi and peripherals
I2C3	HSI	64.000000 MHz	no	
I2C5	HSI	64.000000 MHz	yes	Rpi
USART1	HSI	64.000000 MHz	yes	
USART2	HSI	64.000000 MHz	no	
USART3	HSI	64.000000 MHz	yes	Rpi
UART4	HSI	64.000000 MHz	yes	Linux console
UART5	HSI	64.000000 MHz	no	
USART6	HSI	64.000000 MHz	no	
UART7	HSI	64.000000 MHz	no	
UART8	HSI	64.000000 MHz	no	
MC01	HSI	64.000000 MHz	no	Available so can be used
HSE	N.A.	24.000000 MHz	yes	
DSIPLL	HSE	125.000000 MHz	no	DSI DPHY PLL
DSIBL	DSIPLL	125.000000 MHz	no	DSI lanebyte clock
RTCDIV	HSE	1.000000 MHz	yes	Only used when RTC source
is HSE				



ck_per	HSE	24.000000	MHz	yes	
ADC	ck_per	24.000000	MHz	yes	Can use internal divider to
be < 40MHz					
PLL1	HSE	xxx	MHz	yes	
PLL1P	PLL1	xxx	MHz	yes	
MPUDIV	PLL1P	xxx	MHz	yes	
Cortex-A7	PLL1P	xxx	MHz	yes	650 MHz or 800 MHz, see the
information box above					
MC02	Cortex-A7	xxx	MHz	no	Available so can be used
PLL2	HSE	1066.000000	MHz	yes	
PLL2P	PLL2	266.500000	MHz	yes	
AXI	PLL2P	266.500000	MHz	yes	< 266MHz
FMC	AXI	266.500000	MHz	yes	NAND flash
QSPI	AXI	266.500000	MHz	yes	NOR flash
SYSRAM	AXI	266.500000	MHz	yes	
ROM	AXI	266.500000	MHz	yes	
AHB5	AXI	266.500000	MHz	yes	< 266MHz
CRYP1	AHB5	266.500000	MHz	yes	
HASH1	AHB5	266.500000	MHz	yes	
GPI0Z	AHB5	266.500000	MHz	yes	
BKPSRAM	AHB5	266.500000	MHz	yes	
AHB6	AXI	266.500000	MHz	yes	< 266MHz
CRC1	AHB6	266.500000	MHz	yes	
MDMA	AHB6	266.500000	MHz	yes	
USBH	AHB6	266.500000	MHz	yes	USB Host
APB4	AHB6	133.250000	MHz	yes	
APB5	AHB6	66.625000	MHz	yes	
BSEC	APB5	66.625000	MHz	yes	< 67MHz
ETZPC	APB5	66.625000	MHz	yes	
TZC	APB5	66.625000	MHz	yes	
DBGAPB	AXI	133.250000	MHz	yes	JTAG & Coresight
DBGMCU	DBGAPB	66.625000	MHz	yes	
STM	DBGAPB	66.625000	MHz	yes	
PLL2Q	PLL2	533.000000	MHz	yes	
GPU	PLL2Q	533.000000	MHz	yes	< 533MHz
PLL2R	PLL2	533.000000	MHz	yes	
DDR	PLL2R	533.000000	MHz	yes	< 533MHz
PLL3	HSE	417.755859	MHz	yes	
PLL3P	PLL3	208.877930	MHz	yes	
MLAHB	PLL3P	208.877930	MHz	yes	< 209MHz
Cortex-M4	MLAHB	208.877930	MHz	yes	
SRAM1	MLAHB	208.877930	MHz	yes	
SRAM2	MLAHB	208.877930	MHz	yes	
SRAM3	MLAHB	208.877930	MHz	yes	
RETRAM	MLAHB	208.877930	MHz	yes	
AHB1	MLAHB	208.877930	MHz	yes	< 209MHz
AHB2	MLAHB	208.877930	MHz	yes	< 209MHz
DMA1	AHB2	208.877930	MHz	yes	
DMA2	AHB2	208.877930	MHz	yes	
DMAMUX	AHB2	208.877930	MHz	yes	
APB1	MLAHB	104.438965	MHz	yes	
LPTIM1	APB1	104.438965	MHz	yes	
WWDG	APB1	104.438965	MHz	yes	
APB2	MLAHB	104.438965	MHz	yes	
TIM2	MLAHB	208.877930	MHz	yes	TIM Group 1
TIM3	MLAHB	208.877930	MHz	yes	TIM Group 1
TIM4	MLAHB	208.877930	MHz	yes	TIM Group 1
TIM5	MLAHB	208.877930	MHz	yes	TIM Group 1
TIM6	MLAHB	208.877930	MHz	yes	TIM Group 1
TIM7	MLAHB	208.877930	MHz	yes	TIM Group 1
TIM12	MLAHB	208.877930	MHz	yes	TIM Group 1
TIM13	MLAHB	208.877930	MHz	yes	TIM Group 1
TIM14	MLAHB	208.877930	MHz	yes	TIM Group 1
TIM1	MLAHB	208.877930	MHz	yes	TIM Group 2
TIM8	MLAHB	208.877930	MHz	yes	TIM Group 2
TIM15	MLAHB	208.877930	MHz	yes	TIM Group 2
TIM16	MLAHB	208.877930	MHz	yes	TIM Group 2



TIM17	MLAHB	208.877930	MHz	yes	TIM Group 2
APB3	MLAHB	104.438965	MHz	yes	
LPTIM2	APB3	104.438965	MHz	yes	
LPTIM3	APB3	104.438965	MHz	yes	
SYSCFG	APB3	104.438965	MHz	yes	
VREFBUF	APB3	104.438965	MHz	yes	
DTS	APB3	104.438965	MHz	yes	
HDP	APB3	104.438965	MHz	yes	
AHB3	MLAHB	208.877930	MHz	yes	< 209MHz
CRC2	AHB3	208.877930	MHz	yes	
CRYP2	AHB3	208.877930	MHz	yes	
HASH2	AHB3	208.877930	MHz	yes	
DCMI	AHB3	208.877930	MHz	yes	Camera
IPCC	AHB3	208.877930	MHz	yes	Mailbox
AHB4	MLAHB	208.877930	MHz	yes	< 209MHz
PWR	AHB4	208.877930	MHz	yes	
RCC	AHB4	208.877930	MHz	yes	
GPIOA-K	AHB4	208.877930	MHz	yes	
EXTI	AHB4	208.877930	MHz	yes	
PLL3Q	PLL3	24.573874	MHz	yes	
SPI1	PLL3Q	24.573874	MHz	no	
SPI2	PLL3Q	24.573874	MHz	no	
SPI3	PLL3Q	24.573874	MHz	no	
DFSDM	PLL3Q	24.573874	MHz	yes	Digital micro
SAI1	PLL3Q	24.573874	MHz	no	
SAI2	PLL3Q	24.573874	MHz	yes	AudCodec 48kHz (use PLL3R)
for 44.1kHz)					
SAI3	PLL3Q	24.573874	MHz	no	
SAI4	PLL3Q	24.573874	MHz	yes	SPDIF TX 48kHz (use PLL3R)
for 44.1kHz)					
PLL3R	PLL3	11.290699	MHz	yes	
PLL4	HSE	594.000000	MHz	yes	
PLL4P	PLL4	99.000000	MHz	yes	
SDMMC1	PLL4P	99.000000	MHz	yes	µSD card
SDMMC2	PLL4P	99.000000	MHz	yes	eMMC
SDMMC3	PLL4P	99.000000	MHz	yes	
SPDIF	PLL4P	99.000000	MHz	yes	SPDIF RX
PLL4Q	PLL4	74.250000	MHz	yes	
LCD	PLL4Q	74.250000	MHz	yes	LTDC & DSI display pixel
clock					
PLL4R	PLL4	74.250000	MHz	yes	
FDCAN	PLL4R	74.250000	MHz	yes	Should be as high as
possible and < 100MHz					
STGEN	HSE	24.000000	MHz	yes	
USBPHYC	HSE	24.000000	MHz	yes	USB PHY Ctrl for USB Host
and OTG					
USBPLL	USBPHYC	48.000000	MHz	yes	
USB0	USBPLL	48.000000	MHz	yes	USB OTG
CSI	N.A.	4.000000	MHz	yes	Mandatory for IO
compensation					
ETH	N.A.	0.000000	MHz	no	ETH clocked by RGMII PHY

3.1.1.1 Device tree

Here are the corresponding device tree `rcc_clk` sub node properties consumed by the first stage boot loader (FSBL) to configure the clock tree above:

```
st,clksrc = <
    CLK_MPU_PLL1P
    CLK_AXI_PLL2P
    CLK_MCU_PLL3P
    CLK_PLL12_HSE
```



```

        CLK_PLL3_HSE
        CLK_PLL4_HSE
        CLK_RTC_LSE
        CLK_MC01_DISABLED
        CLK_MC02_DISABLED
    >;

    st,clkdiv = <
        1 /*MPU*/
        0 /*AXI*/
        0 /*MCU*/
        1 /*APB1*/
        1 /*APB2*/
        1 /*APB3*/
        1 /*APB4*/
        2 /*APB5*/
        23 /*RTC*/
        0 /*MC01*/
        0 /*MC02*/
    >;

    st,pkcs = <
        CLK_CKPER_HSE
        CLK_FMC_ACLK
        CLK_QSPI_ACLK
        CLK_ETH_DISABLED
        CLK_SDMMC12_PLL4P
        CLK_DSI_DSIPLL
        CLK_STGEN_HSE
        CLK_USBPHY_HSE
        CLK_SPI2S1_PLL3Q
        CLK_SPI2S23_PLL3Q
        CLK_SPI45_HSI
        CLK_SPI6_HSI
        CLK_I2C46_HSI
        CLK_SDMMC3_PLL4P
        CLK_USB0_USBPHY
        CLK_ADC_CKPER
        CLK_CEC_LSE
        CLK_I2C12_HSI
        CLK_I2C35_HSI
        CLK_UART1_HSI
        CLK_UART24_HSI
        CLK_UART35_HSI
        CLK_UART6_HSI
        CLK_UART78_HSI
        CLK_SPDIF_PLL4P
        CLK_FDCAN_PLL4R
        CLK_SAI1_PLL3Q
        CLK_SAI2_PLL3Q
        CLK_SAI3_PLL3Q
        CLK_SAI4_PLL3Q
        CLK_RNG1_LSI
        CLK_RNG2_LSI
        CLK_LPTIM1_PCLK1
        CLK_LPTIM23_PCLK3
        CLK_LPTIM45_LSE
    >;

    /* VCO = 1066.0 MHz => P = 266 (AXI), Q = 533 (GPU), R = 533 (DDR) */
    pll2: st,pll@1 {
        cfg = < 2 65 1 0 0 PQR(1,1,1) >;
        frac = < 0x1400 >;
        u-boot,dm-pre-reloc;
    };

    /* VCO = 417.8 MHz => P = 209, Q = 25, R = 11 */

```



```

pll3: st,pll@2 {
    cfg = < 1 33 1 16 36 PQR(1,1,1) >;
    frac = < 0x1a04 >;
    u-boot,dm-pre-reloc;
};

/* VCO = 594.0 MHz => P = 99, Q = 74, R = 74 */
pll4: st,pll@3 {
    cfg = < 3 98 5 7 7 PQR(1,1,1) >;
    u-boot,dm-pre-reloc;
};
    
```

3.2 STM32MP157x-DK2 case

This chapter shows the boot time clock tree set by the FSBL on STM32MP157x-DK2 DISCO board. Linux eventual runtime modifications are not covered here.

3.2.1 Clock tree

The following table shows what STM32MP157x-DK2 clock tree looks like, as a result of the boot chain execution with the device tree built with STM32CubeMX.

Component Comment	Parent	Frequency	Used?	
LSI	N.A.	0.032000 MHz	yes	Mandatory for IWDG, DAC
DAC	LSI	0.032000 MHz	no	
RNG1	LSI	0.032000 MHz	yes	Input frequency should be as low as possible
RNG2	LSI	0.032000 MHz	yes	Input frequency should be as low as possible
IWDG1	LSI	0.032000 MHz	yes	
IWDG2	LSI	0.032000 MHz	yes	
LSE	N.A.	0.032768 MHz	yes	Mandatory for DTS
RTC	LSE	0.032768 MHz	yes	
TAMP	LSE	0.032768 MHz	yes	
CEC	LSE	0.032768 MHz	yes	
LPTIM4	LSE	0.032768 MHz	yes	
LPTIM5	LSE	0.032768 MHz	yes	
HSI	N.A.	64.000000 MHz	yes	
SPI4	HSI	64.000000 MHz	yes	Arduino
SPI5	HSI	64.000000 MHz	yes	Rpi
SPI6	HSI	64.000000 MHz	no	
I2C4	HSI	64.000000 MHz	yes	PMIC
I2C6	HSI	64.000000 MHz	no	
I2C1	HSI	64.000000 MHz	yes	Rpi and peripherals
I2C2	HSI	64.000000 MHz	no	
I2C3	HSI	64.000000 MHz	no	
I2C5	HSI	64.000000 MHz	yes	Rpi and Arduino
USART1	HSI	64.000000 MHz	no	
USART2	HSI	64.000000 MHz	yes	Bluetooth
USART3	HSI	64.000000 MHz	yes	Rpi
UART4	HSI	64.000000 MHz	yes	Linux console
UART5	HSI	64.000000 MHz	no	
USART6	HSI	64.000000 MHz	no	
UART7	HSI	64.000000 MHz	yes	Arduino
UART8	HSI	64.000000 MHz	no	
MCO1	HSI	64.000000 MHz	no	Available so can be used
HSE	N.A.	24.000000 MHz	yes	
DSIPLL	HSE	125.000000 MHz	no	DSI DPHY PLL
DSIBL	DSIPLL	125.000000 MHz	no	DSI lanebyte clock



RTCDIV	HSE	1.000000 MHz	yes	Only used when RTC source
is HSE				
ck_per	HSE	24.000000 MHz	yes	
ADC	ck_per	24.000000 MHz	yes	Can use internal divider to
be < 40MHz				
PLL1	HSE	xxx MHz	yes	
PLL1P	PLL1	xxx MHz	yes	
MPUDIV	PLL1P	xxx MHz	yes	
Cortex-A7	PLL1P	xxx MHz	yes	650 MHz or 800 MHz, see the
information box above				
MC02	Cortex-A7	xxx MHz	no	Available so can be used
PLL2	HSE	1066.000000 MHz	yes	
PLL2P	PLL2	266.500000 MHz	yes	
AXI	PLL2P	266.500000 MHz	yes	< 266MHz
FMC	AXI	266.500000 MHz	no	
QSPI	AXI	266.500000 MHz	no	
SYSRAM	AXI	266.500000 MHz	yes	
ROM	AXI	266.500000 MHz	yes	
AHB5	AXI	266.500000 MHz	yes	< 266MHz
CRYP1	AHB5	266.500000 MHz	yes	
HASH1	AHB5	266.500000 MHz	yes	
GPI0Z	AHB5	266.500000 MHz	yes	
BKPSRAM	AHB5	266.500000 MHz	yes	
AHB6	AXI	266.500000 MHz	yes	< 266MHz
CRC1	AHB6	266.500000 MHz	yes	
MDMA	AHB6	266.500000 MHz	yes	
USBH	AHB6	266.500000 MHz	yes	USB Host
APB4	AHB6	133.250000 MHz	yes	
APB5	AHB6	66.625000 MHz	yes	
BSEC	APB5	66.625000 MHz	yes	< 67MHz
ETZPC	APB5	66.625000 MHz	yes	
TZC	APB5	66.625000 MHz	yes	
DBGAPB	AXI	133.250000 MHz	yes	JTAG & Coresight
DBGMCU	DBGAPB	66.625000 MHz	yes	
STM	DBGAPB	66.625000 MHz	no	
PLL2Q	PLL2	533.000000 MHz	yes	
GPU	PLL2Q	533.000000 MHz	yes	< 533MHz
PLL2R	PLL2	533.000000 MHz	yes	
DDR	PLL2R	533.000000 MHz	yes	< 533MHz
PLL3	HSE	417.755859 MHz	yes	
PLL3P	PLL3	208.877930 MHz	yes	
MLAHB	PLL3P	208.877930 MHz	yes	< 209MHz
Cortex-M4	MLAHB	208.877930 MHz	yes	
SRAM1	MLAHB	208.877930 MHz	yes	
SRAM2	MLAHB	208.877930 MHz	yes	
SRAM3	MLAHB	208.877930 MHz	yes	
RETRAM	MLAHB	208.877930 MHz	yes	
AHB1	MLAHB	208.877930 MHz	yes	< 209MHz
AHB2	MLAHB	208.877930 MHz	yes	< 209MHz
DMA1	AHB2	208.877930 MHz	yes	
DMA2	AHB2	208.877930 MHz	yes	
DMAMUX	AHB2	208.877930 MHz	yes	
APB1	MLAHB	104.438965 MHz	yes	
LPTIM1	APB1	104.438965 MHz	yes	
WWDG	APB1	104.438965 MHz	yes	
APB2	MLAHB	104.438965 MHz	yes	
TIM2	MLAHB	208.877930 MHz	yes	TIM Group 1
TIM3	MLAHB	208.877930 MHz	yes	TIM Group 1
TIM4	MLAHB	208.877930 MHz	yes	TIM Group 1
TIM5	MLAHB	208.877930 MHz	yes	TIM Group 1
TIM6	MLAHB	208.877930 MHz	yes	TIM Group 1
TIM7	MLAHB	208.877930 MHz	yes	TIM Group 1
TIM12	MLAHB	208.877930 MHz	yes	TIM Group 1
TIM13	MLAHB	208.877930 MHz	yes	TIM Group 1
TIM14	MLAHB	208.877930 MHz	yes	TIM Group 1
TIM1	MLAHB	208.877930 MHz	yes	TIM Group 2
TIM8	MLAHB	208.877930 MHz	yes	TIM Group 2



TIM15	MLAHB	208.877930	MHz	yes	TIM Group 2
TIM16	MLAHB	208.877930	MHz	yes	TIM Group 2
TIM17	MLAHB	208.877930	MHz	yes	TIM Group 2
APB3	MLAHB	104.438965	MHz	yes	
LPTIM2	APB3	104.438965	MHz	yes	
LPTIM3	APB3	104.438965	MHz	yes	
SYSCFG	APB3	104.438965	MHz	yes	
VREFBUF	APB3	104.438965	MHz	yes	
DTS	APB3	104.438965	MHz	yes	
HDP	APB3	104.438965	MHz	no	
AHB3	MLAHB	208.877930	MHz	yes	< 209MHz
CRC2	AHB3	208.877930	MHz	yes	
CRYP2	AHB3	208.877930	MHz	yes	
HASH2	AHB3	208.877930	MHz	yes	
DCMI	AHB3	208.877930	MHz	no	
IPCC	AHB3	208.877930	MHz	yes	Mailbox
AHB4	MLAHB	208.877930	MHz	yes	< 209MHz
PWR	AHB4	208.877930	MHz	yes	
RCC	AHB4	208.877930	MHz	yes	
GPIOA-K	AHB4	208.877930	MHz	yes	
EXTI	AHB4	208.877930	MHz	yes	
PLL3Q	PLL3	24.573874	MHz	yes	
SPI1	PLL3Q	24.573874	MHz	yes	SPI2S1 for BT PCM
SPI2	PLL3Q	24.573874	MHz	yes	SPI2S2 for HDMI
SPI3	PLL3Q	24.573874	MHz	no	
DFSDM	PLL3Q	24.573874	MHz	no	
SAI1	PLL3Q	24.573874	MHz	no	
SAI2	PLL3Q	24.573874	MHz	yes	AudCodec 48kHz (use PLL3R)
for 44.1kHz)					
SAI3	PLL3Q	24.573874	MHz	no	
SAI4	PLL3Q	24.573874	MHz	no	
PLL3R	PLL3	11.290699	MHz	yes	
PLL4	HSE	594.000000	MHz	yes	
PLL4P	PLL4	99.000000	MHz	yes	
SDMMC1	PLL4P	99.000000	MHz	yes	µSD card
SDMMC2	PLL4P	99.000000	MHz	yes	Wifi
SDMMC3	PLL4P	99.000000	MHz	yes	Rpi
SPDIF	PLL4P	99.000000	MHz	no	
PLL4Q	PLL4	74.250000	MHz	yes	
LCD	PLL4Q	74.250000	MHz	yes	LTDC & DSI display pixel
clock					
PLL4R	PLL4	74.250000	MHz	yes	
FDCAN	PLL4R	74.250000	MHz	no	
STGEN	HSE	24.000000	MHz	yes	
USBPHYC	HSE	24.000000	MHz	yes	USB PHY Ctrl for USB Host
and OTG					
USBPLL	USBPHYC	48.000000	MHz	yes	
USB0	USBPLL	48.000000	MHz	yes	USB OTG
CSI	N.A.	4.000000	MHz	yes	Mandatory for IO
compensation					
ETH	N.A.	0.000000	MHz	no	ETH clocked by RGMII PHY

3.2.2 Device tree

Here are the corresponding device tree `rcc_clk` sub node properties consumed by the first stage boot loader (FSBL) to configure the clock tree above:

```

st,clksrc = <
    CLK_MPU_PLL1P
    CLK_AXI_PLL2P
    CLK_MCU_PLL3P
    CLK_PLL12_HSE
    
```



```

        CLK_PLL3_HSE
        CLK_PLL4_HSE
        CLK_RTC_LSE
        CLK_MC01_DISABLED
        CLK_MC02_DISABLED
    >;

    st,clkdiv = <
        1 /*MPU*/
        0 /*AXI*/
        0 /*MCU*/
        1 /*APB1*/
        1 /*APB2*/
        1 /*APB3*/
        1 /*APB4*/
        2 /*APB5*/
        23 /*RTC*/
        0 /*MC01*/
        0 /*MC02*/
    >;

    st,pkcs = <
        CLK_CKPER_HSE
        CLK_FMC_ACLK
        CLK_QSPI_ACLK
        CLK_ETH_DISABLED
        CLK_SDMMC12_PLL4P
        CLK_DSI_DSIPLL
        CLK_STGEN_HSE
        CLK_USBPHY_HSE
        CLK_SPI2S1_PLL3Q
        CLK_SPI2S23_PLL3Q
        CLK_SPI45_HSI
        CLK_SPI6_HSI
        CLK_I2C46_HSI
        CLK_SDMMC3_PLL4P
        CLK_USB0_USBPHY
        CLK_ADC_CKPER
        CLK_CEC_LSE
        CLK_I2C12_HSI
        CLK_I2C35_HSI
        CLK_UART1_HSI
        CLK_UART24_HSI
        CLK_UART35_HSI
        CLK_UART6_HSI
        CLK_UART78_HSI
        CLK_SPDIF_PLL4P
        CLK_FDCAN_PLL4R
        CLK_SAI1_PLL3Q
        CLK_SAI2_PLL3Q
        CLK_SAI3_PLL3Q
        CLK_SAI4_PLL3Q
        CLK_RNG1_LSI
        CLK_RNG2_LSI
        CLK_LPTIM1_PCLK1
        CLK_LPTIM23_PCLK3
        CLK_LPTIM45_LSE
    >;

    /* VCO = 1066.0 MHz => P = 266 (AXI), Q = 533 (GPU), R = 533 (DDR) */
    pll2: st,pll@1 {
        cfg = < 2 65 1 0 0 PQR(1,1,1) >;
        frac = < 0x1400 >;
        u-boot,dm-pre-reloc;
    };

    /* VCO = 417.8 MHz => P = 209, Q = 25, R = 11 */

```



```

pll3: st,pll@2 {
    cfg = < 1 33 1 16 36 PQR(1,1,1) >;
    frac = < 0x1a04 >;
    u-boot,dm-pre-reloc;
};

/* VCO = 594.0 MHz => P = 99, Q = 74, R = 74 */
pll4: st,pll@3 {
    cfg = < 3 98 5 7 7 PQR(1,1,1) >;
    u-boot,dm-pre-reloc;
};

```

Reset and Clock Control

Low Speed Internal oscillator (STM32 clock source)

Low Speed External oscillator (STM32 clock source)

High Speed Internal oscillator (STM32 clock source) or High Speed Synchronous Serial Interface (MIPI® Alliance standard)

High Speed External oscillator (STM32 clock source)

Multi Speed Internal oscillator (STM32 clock source)

Cortex®

First Stage Boot Loader

Linux® is a registered trademark of Linus Torvalds.

Das U-Boot -- the Universal Boot Loader (see [U-Boot_overview](#))

Secondary Program Loader, *Also known as **U-Boot SPL***

Read Only Memory

Central processing unit

Dynamic voltage and frequency scaling

Independent Watchdog

Digital-to-analog converter (Electronic circuit that converts a binary number into a continuously varying value.)

Device Tree Source (in software context) or Digital Temperature Sensor (in peripheral context)

Real Time Clock

Tamper

Consumer Electronics Control (HDMI standard)

Power Management Integrated Circuit

Display Serial Interface (MIPI® Alliance standard)

Analog-to-digital converter. The process of converting a sampled analog signal to a digital code that represents the amplitude of the original signal sample.

USB Host (STM32 specific)

Boot and Security and OTP control

Extended TrustZone Protection Controller



TrustZone® address space Controller for DDR

Arm® and TrustZone® are registered trademarks of Arm Limited (or its subsidiaries) in the US and/or elsewhere.

debug and test protocol, named from the Joint Test Action Group that developed it

System Trace Module

Graphics Processing Units

Doubledata rate (memory domain)

System Configuration

voltage reference buffer (STM32 specific)

Hardware Debug Port

Digital Camera Memory Interface

Inter-Processor Communication Controller

External Interrupt

Digital Filter for Sigma-Delta Modulator

Transmit

former spelling for e•MMC ('e' in italic)

Receive

LCD TFT Display Controller (STM32 specific)

System Time Generator

USB On-The-Go (Capability/type of USB port, acting primarily as USB device, to also act as USB host. Also known as USB OTG.)

input/output

Ethernet

Microprocessor Unit

Microcontroller Unit (MCUs have internal flash memory and are intended to operate with a minimum amount of external support ICs. They commonly are a self-contained, system-on-chip (SoC) designs.)

Discovery kit

BlueTooth

High-Definition Multimedia Interface (HDMI standard)

Stable: 19.03.2021 - 14:44 / Revision: 19.03.2021 - 14:37

All the peripherals receive one or several clocks that are generated via [RCC internal peripheral](#). RCC relies on several clocks sources (LSI, LSE, HSI, HSE, CSI) and four PLL in order to provide adequate input frequencies to all the peripherals. The clock tree covers all the system clock distribution aspects, from the clock sources to the consumer peripherals (internal and external), except clock gating management that is locally controlled by each peripheral driver.

Contents

1 Overview	82
2 How to build a clock tree?	83



3 ST boards clock tree	84
3.1 STM32MP157x-EV1 case	84
3.1.1 Clock tree	84
3.1.1.1 Device tree	86
3.2 STM32MP157x-DK2 case	88
3.2.1 Clock tree	88
3.2.2 Device tree	90



1 Overview

The clock tree is managed via RCC internal peripheral hardware block and it is configured at different steps from the Cortex-A7:

- When the device is **reset**, all RCC registers take their reset values: the four PLL are disabled and most of the clock source selectors are pointing to the HSI.
- The ROM Code configures the minimum clock tree needed to boot on the selected boot device.
- The **FSBL** (Cf. [Boot chain overview](#)) completely configures the clock tree as expected, all the way up to Linux, thanks to the configuration given in the [device tree](#).
 - U-Boot SPL (basic boot chain) binding is available in [doc/device-tree-bindings/clock/st,stm32mp1.txt](#)
 - TF-A (trusted boot chain) binding follows the same syntax as U-Boot SPL
- **Linux** is responsible for the clock tree at runtime and it may partly modify it thanks to information taken from the [device tree](#):
 - Linux binding is available in [Documentation/devicetree/bindings/clock/clock-bindings.txt](#) (and surrounding files): 'fixed-clock' compatible, 'clocks' and 'assigned-clocks' properties are important concepts to understand the management of clocks providers/consumers.

Notice that [STM32Cube](#) (running on the Cortex-M4) does not control the clock tree setup so the configuration applied by the Cortex-A7 must provide suitable clocks to Cortex-M4, like it is explained in [resource manager](#) description. One exception to this rule is the [engineering boot](#), allowing to directly load (and debug) the Cortex-M4 whereas the Cortex-A7 execution is stalled in the ROM code.



2 How to build a clock tree?

Building a clock tree is quite complex as it needs to take into account the constraints set by each internal and external peripheral, including external clock sources.

We encourage the use of [STM32CubeMX](#) to build the clock tree, and avoid having to know all internal peripherals details: the tool allows to select the peripherals that will be present on the board, fix the clock sources frequencies and automatically find an optimized clock tree. It is then able to generate the device tree that is directly consumed by the boot chain and Linux.



3 ST boards clock tree



See [How to change the CPU frequency](#) article for more information about the CPU frequency setting (PLL1P), including dynamic voltage and frequency scaling (DVFS) configuration.

3.1 STM32MP157x-EV1 case

This chapter shows the boot time clock tree set by the FSBL on STM32MP157x-EV1 evaluation board. Linux eventual runtime modifications are not covered here.

3.1.1 Clock tree

The following table shows what STM32MP157x-EV1 clock tree looks like, as a result of the boot chain execution with the device tree built with STM32CubeMX.

Component Comment	Parent	Frequency	Used?	
LSI	N.A.	0.032000 MHz	yes	Mandatory for IWDG, DAC
DAC	LSI	0.032000 MHz	yes	
RNG1	LSI	0.032000 MHz	yes	Input frequency should be
as low as possible				
RNG2	LSI	0.032000 MHz	yes	Input frequency should be
as low as possible				
IWDG1	LSI	0.032000 MHz	yes	
IWDG2	LSI	0.032000 MHz	yes	
LSE	N.A.	0.032768 MHz	yes	Mandatory for DTS
RTC	LSE	0.032768 MHz	yes	
TAMP	LSE	0.032768 MHz	yes	
CEC	LSE	0.032768 MHz	yes	
LPTIM4	LSE	0.032768 MHz	yes	
LPTIM5	LSE	0.032768 MHz	yes	
HSI	N.A.	64.000000 MHz	yes	
SPI4	HSI	64.000000 MHz	no	
SPI5	HSI	64.000000 MHz	no	
SPI6	HSI	64.000000 MHz	yes	
I2C4	HSI	64.000000 MHz	yes	PMIC
I2C6	HSI	64.000000 MHz	no	
I2C1	HSI	64.000000 MHz	no	
I2C2	HSI	64.000000 MHz	yes	Rpi and peripherals
I2C3	HSI	64.000000 MHz	no	
I2C5	HSI	64.000000 MHz	yes	Rpi
USART1	HSI	64.000000 MHz	yes	
USART2	HSI	64.000000 MHz	no	
USART3	HSI	64.000000 MHz	yes	Rpi
UART4	HSI	64.000000 MHz	yes	Linux console
UART5	HSI	64.000000 MHz	no	
USART6	HSI	64.000000 MHz	no	
UART7	HSI	64.000000 MHz	no	
UART8	HSI	64.000000 MHz	no	
MC01	HSI	64.000000 MHz	no	Available so can be used
HSE	N.A.	24.000000 MHz	yes	
DSIPLL	HSE	125.000000 MHz	no	DSI DPHY PLL
DSIBL	DSIPLL	125.000000 MHz	no	DSI lanebyte clock
RTCDIV	HSE	1.000000 MHz	yes	Only used when RTC source
is HSE				



ck_per	HSE	24.000000	MHz	yes	
ADC	ck_per	24.000000	MHz	yes	Can use internal divider to
be < 40MHz					
PLL1	HSE	xxx	MHz	yes	
PLL1P	PLL1	xxx	MHz	yes	
MPUDIV	PLL1P	xxx	MHz	yes	
Cortex-A7	PLL1P	xxx	MHz	yes	650 MHz or 800 MHz, see the
information box above					
MC02	Cortex-A7	xxx	MHz	no	Available so can be used
PLL2	HSE	1066.000000	MHz	yes	
PLL2P	PLL2	266.500000	MHz	yes	
AXI	PLL2P	266.500000	MHz	yes	< 266MHz
FMC	AXI	266.500000	MHz	yes	NAND flash
QSPI	AXI	266.500000	MHz	yes	NOR flash
SYSRAM	AXI	266.500000	MHz	yes	
ROM	AXI	266.500000	MHz	yes	
AHB5	AXI	266.500000	MHz	yes	< 266MHz
CRYP1	AHB5	266.500000	MHz	yes	
HASH1	AHB5	266.500000	MHz	yes	
GPI0Z	AHB5	266.500000	MHz	yes	
BKPSRAM	AHB5	266.500000	MHz	yes	
AHB6	AXI	266.500000	MHz	yes	< 266MHz
CRC1	AHB6	266.500000	MHz	yes	
MDMA	AHB6	266.500000	MHz	yes	
USBH	AHB6	266.500000	MHz	yes	USB Host
APB4	AHB6	133.250000	MHz	yes	
APB5	AHB6	66.625000	MHz	yes	
BSEC	APB5	66.625000	MHz	yes	< 67MHz
ETZPC	APB5	66.625000	MHz	yes	
TZC	APB5	66.625000	MHz	yes	
DBGAPB	AXI	133.250000	MHz	yes	JTAG & Coresight
DBGMCU	DBGAPB	66.625000	MHz	yes	
STM	DBGAPB	66.625000	MHz	yes	
PLL2Q	PLL2	533.000000	MHz	yes	
GPU	PLL2Q	533.000000	MHz	yes	< 533MHz
PLL2R	PLL2	533.000000	MHz	yes	
DDR	PLL2R	533.000000	MHz	yes	< 533MHz
PLL3	HSE	417.755859	MHz	yes	
PLL3P	PLL3	208.877930	MHz	yes	
MLAHB	PLL3P	208.877930	MHz	yes	< 209MHz
Cortex-M4	MLAHB	208.877930	MHz	yes	
SRAM1	MLAHB	208.877930	MHz	yes	
SRAM2	MLAHB	208.877930	MHz	yes	
SRAM3	MLAHB	208.877930	MHz	yes	
RETRAM	MLAHB	208.877930	MHz	yes	
AHB1	MLAHB	208.877930	MHz	yes	< 209MHz
AHB2	MLAHB	208.877930	MHz	yes	< 209MHz
DMA1	AHB2	208.877930	MHz	yes	
DMA2	AHB2	208.877930	MHz	yes	
DMAMUX	AHB2	208.877930	MHz	yes	
APB1	MLAHB	104.438965	MHz	yes	
LPTIM1	APB1	104.438965	MHz	yes	
WWDG	APB1	104.438965	MHz	yes	
APB2	MLAHB	104.438965	MHz	yes	
TIM2	MLAHB	208.877930	MHz	yes	TIM Group 1
TIM3	MLAHB	208.877930	MHz	yes	TIM Group 1
TIM4	MLAHB	208.877930	MHz	yes	TIM Group 1
TIM5	MLAHB	208.877930	MHz	yes	TIM Group 1
TIM6	MLAHB	208.877930	MHz	yes	TIM Group 1
TIM7	MLAHB	208.877930	MHz	yes	TIM Group 1
TIM12	MLAHB	208.877930	MHz	yes	TIM Group 1
TIM13	MLAHB	208.877930	MHz	yes	TIM Group 1
TIM14	MLAHB	208.877930	MHz	yes	TIM Group 1
TIM1	MLAHB	208.877930	MHz	yes	TIM Group 2
TIM8	MLAHB	208.877930	MHz	yes	TIM Group 2
TIM15	MLAHB	208.877930	MHz	yes	TIM Group 2
TIM16	MLAHB	208.877930	MHz	yes	TIM Group 2



TIM17	MLAHB	208.877930	MHz	yes	TIM Group 2
APB3	MLAHB	104.438965	MHz	yes	
LPTIM2	APB3	104.438965	MHz	yes	
LPTIM3	APB3	104.438965	MHz	yes	
SYSCFG	APB3	104.438965	MHz	yes	
VREFBUF	APB3	104.438965	MHz	yes	
DTS	APB3	104.438965	MHz	yes	
HDP	APB3	104.438965	MHz	yes	
AHB3	MLAHB	208.877930	MHz	yes	< 209MHz
CRC2	AHB3	208.877930	MHz	yes	
CRYP2	AHB3	208.877930	MHz	yes	
HASH2	AHB3	208.877930	MHz	yes	
DCMI	AHB3	208.877930	MHz	yes	Camera
IPCC	AHB3	208.877930	MHz	yes	Mailbox
AHB4	MLAHB	208.877930	MHz	yes	< 209MHz
PWR	AHB4	208.877930	MHz	yes	
RCC	AHB4	208.877930	MHz	yes	
GPIOA-K	AHB4	208.877930	MHz	yes	
EXTI	AHB4	208.877930	MHz	yes	
PLL3Q	PLL3	24.573874	MHz	yes	
SPI1	PLL3Q	24.573874	MHz	no	
SPI2	PLL3Q	24.573874	MHz	no	
SPI3	PLL3Q	24.573874	MHz	no	
DFSDM	PLL3Q	24.573874	MHz	yes	Digital micro
SAI1	PLL3Q	24.573874	MHz	no	
SAI2	PLL3Q	24.573874	MHz	yes	AudCodec 48kHz (use PLL3R)
for 44.1kHz)					
SAI3	PLL3Q	24.573874	MHz	no	
SAI4	PLL3Q	24.573874	MHz	yes	SPDIF TX 48kHz (use PLL3R)
for 44.1kHz)					
PLL3R	PLL3	11.290699	MHz	yes	
PLL4	HSE	594.000000	MHz	yes	
PLL4P	PLL4	99.000000	MHz	yes	
SDMMC1	PLL4P	99.000000	MHz	yes	µSD card
SDMMC2	PLL4P	99.000000	MHz	yes	eMMC
SDMMC3	PLL4P	99.000000	MHz	yes	
SPDIF	PLL4P	99.000000	MHz	yes	SPDIF RX
PLL4Q	PLL4	74.250000	MHz	yes	
LCD	PLL4Q	74.250000	MHz	yes	LTDC & DSI display pixel
clock					
PLL4R	PLL4	74.250000	MHz	yes	
FDCAN	PLL4R	74.250000	MHz	yes	Should be as high as
possible and < 100MHz					
STGEN	HSE	24.000000	MHz	yes	
USBPHYC	HSE	24.000000	MHz	yes	USB PHY Ctrl for USB Host
and OTG					
USBPLL	USBPHYC	48.000000	MHz	yes	
USB0	USBPLL	48.000000	MHz	yes	USB OTG
CSI	N.A.	4.000000	MHz	yes	Mandatory for IO
compensation					
ETH	N.A.	0.000000	MHz	no	ETH clocked by RGMII PHY

3.1.1.1 Device tree

Here are the corresponding device tree `rcc_clk` sub node properties consumed by the first stage boot loader (FSBL) to configure the clock tree above:

```
st,clksrc = <
    CLK_MPU_PLL1P
    CLK_AXI_PLL2P
    CLK_MCU_PLL3P
    CLK_PLL12_HSE
```



```

        CLK_PLL3_HSE
        CLK_PLL4_HSE
        CLK_RTC_LSE
        CLK_MC01_DISABLED
        CLK_MC02_DISABLED
    >;

    st,clkdiv = <
        1 /*MPU*/
        0 /*AXI*/
        0 /*MCU*/
        1 /*APB1*/
        1 /*APB2*/
        1 /*APB3*/
        1 /*APB4*/
        2 /*APB5*/
        23 /*RTC*/
        0 /*MC01*/
        0 /*MC02*/
    >;

    st,pkcs = <
        CLK_CKPER_HSE
        CLK_FMC_ACLK
        CLK_QSPI_ACLK
        CLK_ETH_DISABLED
        CLK_SDMMC12_PLL4P
        CLK_DSI_DSIPLL
        CLK_STGEN_HSE
        CLK_USBPHY_HSE
        CLK_SPI2S1_PLL3Q
        CLK_SPI2S23_PLL3Q
        CLK_SPI45_HSI
        CLK_SPI6_HSI
        CLK_I2C46_HSI
        CLK_SDMMC3_PLL4P
        CLK_USB0_USBPHY
        CLK_ADC_CKPER
        CLK_CEC_LSE
        CLK_I2C12_HSI
        CLK_I2C35_HSI
        CLK_UART1_HSI
        CLK_UART24_HSI
        CLK_UART35_HSI
        CLK_UART6_HSI
        CLK_UART78_HSI
        CLK_SPDIF_PLL4P
        CLK_FDCAN_PLL4R
        CLK_SAI1_PLL3Q
        CLK_SAI2_PLL3Q
        CLK_SAI3_PLL3Q
        CLK_SAI4_PLL3Q
        CLK_RNG1_LSI
        CLK_RNG2_LSI
        CLK_LPTIM1_PCLK1
        CLK_LPTIM23_PCLK3
        CLK_LPTIM45_LSE
    >;

    /* VCO = 1066.0 MHz => P = 266 (AXI), Q = 533 (GPU), R = 533 (DDR) */
    pll2: st,pll@1 {
        cfg = < 2 65 1 0 0 PQR(1,1,1) >;
        frac = < 0x1400 >;
        u-boot,dm-pre-reloc;
    };

    /* VCO = 417.8 MHz => P = 209, Q = 25, R = 11 */

```



```

pll3: st,pll@2 {
    cfg = < 1 33 1 16 36 PQR(1,1,1) >;
    frac = < 0x1a04 >;
    u-boot,dm-pre-reloc;
};

/* VCO = 594.0 MHz => P = 99, Q = 74, R = 74 */
pll4: st,pll@3 {
    cfg = < 3 98 5 7 7 PQR(1,1,1) >;
    u-boot,dm-pre-reloc;
};
    
```

3.2 STM32MP157x-DK2 case

This chapter shows the boot time clock tree set by the FSBL on STM32MP157x-DK2 DISCO board. Linux eventual runtime modifications are not covered here.

3.2.1 Clock tree

The following table shows what STM32MP157x-DK2 clock tree looks like, as a result of the boot chain execution with the device tree built with STM32CubeMX.

Component Comment	Parent	Frequency	Used?	
LSI	N.A.	0.032000 MHz	yes	Mandatory for IWDG, DAC
DAC	LSI	0.032000 MHz	no	
RNG1	LSI	0.032000 MHz	yes	Input frequency should be as low as possible
RNG2	LSI	0.032000 MHz	yes	Input frequency should be as low as possible
IWDG1	LSI	0.032000 MHz	yes	
IWDG2	LSI	0.032000 MHz	yes	
LSE	N.A.	0.032768 MHz	yes	Mandatory for DTS
RTC	LSE	0.032768 MHz	yes	
TAMP	LSE	0.032768 MHz	yes	
CEC	LSE	0.032768 MHz	yes	
LPTIM4	LSE	0.032768 MHz	yes	
LPTIM5	LSE	0.032768 MHz	yes	
HSI	N.A.	64.000000 MHz	yes	
SPI4	HSI	64.000000 MHz	yes	Arduino
SPI5	HSI	64.000000 MHz	yes	Rpi
SPI6	HSI	64.000000 MHz	no	
I2C4	HSI	64.000000 MHz	yes	PMIC
I2C6	HSI	64.000000 MHz	no	
I2C1	HSI	64.000000 MHz	yes	Rpi and peripherals
I2C2	HSI	64.000000 MHz	no	
I2C3	HSI	64.000000 MHz	no	
I2C5	HSI	64.000000 MHz	yes	Rpi and Arduino
USART1	HSI	64.000000 MHz	no	
USART2	HSI	64.000000 MHz	yes	Bluetooth
USART3	HSI	64.000000 MHz	yes	Rpi
UART4	HSI	64.000000 MHz	yes	Linux console
UART5	HSI	64.000000 MHz	no	
USART6	HSI	64.000000 MHz	no	
UART7	HSI	64.000000 MHz	yes	Arduino
UART8	HSI	64.000000 MHz	no	
MCO1	HSI	64.000000 MHz	no	Available so can be used
HSE	N.A.	24.000000 MHz	yes	
DSIPLL	HSE	125.000000 MHz	no	DSI DPHY PLL
DSIBL	DSIPLL	125.000000 MHz	no	DSI lanebyte clock



RTCDIV	HSE	1.000000 MHz	yes	Only used when RTC source
is HSE				
ck_per	HSE	24.000000 MHz	yes	
ADC	ck_per	24.000000 MHz	yes	Can use internal divider to
be < 40MHz				
PLL1	HSE	xxx MHz	yes	
PLL1P	PLL1	xxx MHz	yes	
MPUDIV	PLL1P	xxx MHz	yes	
Cortex-A7	PLL1P	xxx MHz	yes	650 MHz or 800 MHz, see the
information box above				
MC02	Cortex-A7	xxx MHz	no	Available so can be used
PLL2	HSE	1066.000000 MHz	yes	
PLL2P	PLL2	266.500000 MHz	yes	
AXI	PLL2P	266.500000 MHz	yes	< 266MHz
FMC	AXI	266.500000 MHz	no	
QSPI	AXI	266.500000 MHz	no	
SYSRAM	AXI	266.500000 MHz	yes	
ROM	AXI	266.500000 MHz	yes	
AHB5	AXI	266.500000 MHz	yes	< 266MHz
CRYP1	AHB5	266.500000 MHz	yes	
HASH1	AHB5	266.500000 MHz	yes	
GPI0Z	AHB5	266.500000 MHz	yes	
BKPSRAM	AHB5	266.500000 MHz	yes	
AHB6	AXI	266.500000 MHz	yes	< 266MHz
CRC1	AHB6	266.500000 MHz	yes	
MDMA	AHB6	266.500000 MHz	yes	
USBH	AHB6	266.500000 MHz	yes	USB Host
APB4	AHB6	133.250000 MHz	yes	
APB5	AHB6	66.625000 MHz	yes	
BSEC	APB5	66.625000 MHz	yes	< 67MHz
ETZPC	APB5	66.625000 MHz	yes	
TZC	APB5	66.625000 MHz	yes	
DBGAPB	AXI	133.250000 MHz	yes	JTAG & Coresight
DBGMCU	DBGAPB	66.625000 MHz	yes	
STM	DBGAPB	66.625000 MHz	no	
PLL2Q	PLL2	533.000000 MHz	yes	
GPU	PLL2Q	533.000000 MHz	yes	< 533MHz
PLL2R	PLL2	533.000000 MHz	yes	
DDR	PLL2R	533.000000 MHz	yes	< 533MHz
PLL3	HSE	417.755859 MHz	yes	
PLL3P	PLL3	208.877930 MHz	yes	
MLAHB	PLL3P	208.877930 MHz	yes	< 209MHz
Cortex-M4	MLAHB	208.877930 MHz	yes	
SRAM1	MLAHB	208.877930 MHz	yes	
SRAM2	MLAHB	208.877930 MHz	yes	
SRAM3	MLAHB	208.877930 MHz	yes	
RETRAM	MLAHB	208.877930 MHz	yes	
AHB1	MLAHB	208.877930 MHz	yes	< 209MHz
AHB2	MLAHB	208.877930 MHz	yes	< 209MHz
DMA1	AHB2	208.877930 MHz	yes	
DMA2	AHB2	208.877930 MHz	yes	
DMAMUX	AHB2	208.877930 MHz	yes	
APB1	MLAHB	104.438965 MHz	yes	
LPTIM1	APB1	104.438965 MHz	yes	
WWDG	APB1	104.438965 MHz	yes	
APB2	MLAHB	104.438965 MHz	yes	
TIM2	MLAHB	208.877930 MHz	yes	TIM Group 1
TIM3	MLAHB	208.877930 MHz	yes	TIM Group 1
TIM4	MLAHB	208.877930 MHz	yes	TIM Group 1
TIM5	MLAHB	208.877930 MHz	yes	TIM Group 1
TIM6	MLAHB	208.877930 MHz	yes	TIM Group 1
TIM7	MLAHB	208.877930 MHz	yes	TIM Group 1
TIM12	MLAHB	208.877930 MHz	yes	TIM Group 1
TIM13	MLAHB	208.877930 MHz	yes	TIM Group 1
TIM14	MLAHB	208.877930 MHz	yes	TIM Group 1
TIM1	MLAHB	208.877930 MHz	yes	TIM Group 2
TIM8	MLAHB	208.877930 MHz	yes	TIM Group 2



TIM15	MLAHB	208.877930	MHz	yes	TIM Group 2
TIM16	MLAHB	208.877930	MHz	yes	TIM Group 2
TIM17	MLAHB	208.877930	MHz	yes	TIM Group 2
APB3	MLAHB	104.438965	MHz	yes	
LPTIM2	APB3	104.438965	MHz	yes	
LPTIM3	APB3	104.438965	MHz	yes	
SYSCFG	APB3	104.438965	MHz	yes	
VREFBUF	APB3	104.438965	MHz	yes	
DTS	APB3	104.438965	MHz	yes	
HDP	APB3	104.438965	MHz	no	
AHB3	MLAHB	208.877930	MHz	yes	< 209MHz
CRC2	AHB3	208.877930	MHz	yes	
CRYP2	AHB3	208.877930	MHz	yes	
HASH2	AHB3	208.877930	MHz	yes	
DCMI	AHB3	208.877930	MHz	no	
IPCC	AHB3	208.877930	MHz	yes	Mailbox
AHB4	MLAHB	208.877930	MHz	yes	< 209MHz
PWR	AHB4	208.877930	MHz	yes	
RCC	AHB4	208.877930	MHz	yes	
GPIOA-K	AHB4	208.877930	MHz	yes	
EXTI	AHB4	208.877930	MHz	yes	
PLL3Q	PLL3	24.573874	MHz	yes	
SPI1	PLL3Q	24.573874	MHz	yes	SPI2S1 for BT PCM
SPI2	PLL3Q	24.573874	MHz	yes	SPI2S2 for HDMI
SPI3	PLL3Q	24.573874	MHz	no	
DFSDM	PLL3Q	24.573874	MHz	no	
SAI1	PLL3Q	24.573874	MHz	no	
SAI2	PLL3Q	24.573874	MHz	yes	AudCodec 48kHz (use PLL3R)
for 44.1kHz)					
SAI3	PLL3Q	24.573874	MHz	no	
SAI4	PLL3Q	24.573874	MHz	no	
PLL3R	PLL3	11.290699	MHz	yes	
PLL4	HSE	594.000000	MHz	yes	
PLL4P	PLL4	99.000000	MHz	yes	
SDMMC1	PLL4P	99.000000	MHz	yes	µSD card
SDMMC2	PLL4P	99.000000	MHz	yes	Wifi
SDMMC3	PLL4P	99.000000	MHz	yes	Rpi
SPDIF	PLL4P	99.000000	MHz	no	
PLL4Q	PLL4	74.250000	MHz	yes	
LCD	PLL4Q	74.250000	MHz	yes	LTDC & DSI display pixel
clock					
PLL4R	PLL4	74.250000	MHz	yes	
FDCAN	PLL4R	74.250000	MHz	no	
STGEN	HSE	24.000000	MHz	yes	
USBPHYC	HSE	24.000000	MHz	yes	USB PHY Ctrl for USB Host
and OTG					
USBPLL	USBPHYC	48.000000	MHz	yes	
USB0	USBPLL	48.000000	MHz	yes	USB OTG
CSI	N.A.	4.000000	MHz	yes	Mandatory for IO
compensation					
ETH	N.A.	0.000000	MHz	no	ETH clocked by RGMII PHY

3.2.2 Device tree

Here are the corresponding device tree `rcc_clk` sub node properties consumed by the first stage boot loader (FSBL) to configure the clock tree above:

```
st,clksrc = <
    CLK_MPU_PLL1P
    CLK_AXI_PLL2P
    CLK_MCU_PLL3P
    CLK_PLL12_HSE
```



```

    CLK_PLL3_HSE
    CLK_PLL4_HSE
    CLK_RTC_LSE
    CLK_MC01_DISABLED
    CLK_MC02_DISABLED
>;

st,clkdiv = <
    1 /*MPU*/
    0 /*AXI*/
    0 /*MCU*/
    1 /*APB1*/
    1 /*APB2*/
    1 /*APB3*/
    1 /*APB4*/
    2 /*APB5*/
    23 /*RTC*/
    0 /*MC01*/
    0 /*MC02*/
>;

st,pkcs = <
    CLK_CKPER_HSE
    CLK_FMC_ACLK
    CLK_QSPI_ACLK
    CLK_ETH_DISABLED
    CLK_SDMMC12_PLL4P
    CLK_DSI_DSIPLL
    CLK_STGEN_HSE
    CLK_USBPHY_HSE
    CLK_SPI2S1_PLL3Q
    CLK_SPI2S23_PLL3Q
    CLK_SPI45_HSI
    CLK_SPI6_HSI
    CLK_I2C46_HSI
    CLK_SDMMC3_PLL4P
    CLK_USB0_USBPHY
    CLK_ADC_CKPER
    CLK_CEC_LSE
    CLK_I2C12_HSI
    CLK_I2C35_HSI
    CLK_UART1_HSI
    CLK_UART24_HSI
    CLK_UART35_HSI
    CLK_UART6_HSI
    CLK_UART78_HSI
    CLK_SPDIF_PLL4P
    CLK_FDCAN_PLL4R
    CLK_SAI1_PLL3Q
    CLK_SAI2_PLL3Q
    CLK_SAI3_PLL3Q
    CLK_SAI4_PLL3Q
    CLK_RNG1_LSI
    CLK_RNG2_LSI
    CLK_LPTIM1_PCLK1
    CLK_LPTIM23_PCLK3
    CLK_LPTIM45_LSE
>;

/* VCO = 1066.0 MHz => P = 266 (AXI), Q = 533 (GPU), R = 533 (DDR) */
pll2: st,pll@1 {
    cfg = < 2 65 1 0 0 PQR(1,1,1) >;
    frac = < 0x1400 >;
    u-boot,dm-pre-reloc;
};

/* VCO = 417.8 MHz => P = 209, Q = 25, R = 11 */

```



```

pll3: st,pll@2 {
    cfg = < 1 33 1 16 36 PQR(1,1,1) >;
    frac = < 0x1a04 >;
    u-boot,dm-pre-reloc;
};

/* VCO = 594.0 MHz => P = 99, Q = 74, R = 74 */
pll4: st,pll@3 {
    cfg = < 3 98 5 7 7 PQR(1,1,1) >;
    u-boot,dm-pre-reloc;
};

```

Reset and Clock Control

Low Speed Internal oscillator (STM32 clock source)

Low Speed External oscillator (STM32 clock source)

High Speed Internal oscillator (STM32 clock source) or High Speed Synchronous Serial Interface (MIPI® Alliance standard)

High Speed External oscillator (STM32 clock source)

Multi Speed Internal oscillator (STM32 clock source)

Cortex®

First Stage Boot Loader

Linux® is a registered trademark of Linus Torvalds.

Das U-Boot -- the Universal Boot Loader (see [U-Boot_overview](#))

Secondary Program Loader, *Also known as **U-Boot SPL***

Read Only Memory

Central processing unit

Dynamic voltage and frequency scaling

Independent Watchdog

Digital-to-analog converter (Electronic circuit that converts a binary number into a continuously varying value.)

Device Tree Source (in software context) or Digital Temperature Sensor (in peripheral context)

Real Time Clock

Tamper

Consumer Electronics Control (HDMI standard)

Power Management Integrated Circuit

Display Serial Interface (MIPI® Alliance standard)

Analog-to-digital converter. The process of converting a sampled analog signal to a digital code that represents the amplitude of the original signal sample.

USB Host (STM32 specific)

Boot and Security and OTP control

Extended TrustZone Protection Controller



TrustZone® address space Controller for DDR

Arm® and TrustZone® are registered trademarks of Arm Limited (or its subsidiaries) in the US and/or elsewhere.

debug and test protocol, named from the Joint Test Action Group that developed it

System Trace Module

Graphics Processing Units

Doubledata rate (memory domain)

System Configuration

voltage reference buffer (STM32 specific)

Hardware Debug Port

Digital Camera Memory Interface

Inter-Processor Communication Controller

External Interrupt

Digital Filter for Sigma-Delta Modulator

Transmit

former spelling for e•MMC ('e' in italic)

Receive

LCD TFT Display Controller (STM32 specific)

System Time Generator

USB On-The-Go (Capability/type of USB port, acting primarily as USB device, to also act as USB host. Also known as USB OTG.)

input/output

Ethernet

Microprocessor Unit

Microcontroller Unit (MCUs have internal flash memory and are intended to operate with a minimum amount of external support ICs. They commonly are a self-contained, system-on-chip (SoC) designs.)

Discovery kit

BlueTooth

High-Definition Multimedia Interface (HDMI standard)

Stable: 17.11.2020 - 15:36 / Revision: 12.11.2020 - 15:05

All the peripherals receive one or several clocks that are generated via [RCC internal peripheral](#). RCC relies on several clocks sources (LSI, LSE, HSI, HSE, CSI) and four PLL in order to provide adequate input frequencies to all the peripherals. The clock tree covers all the system clock distribution aspects, from the clock sources to the consumer peripherals (internal and external), except clock gating management that is locally controlled by each peripheral driver.

Contents

1 Overview	95
2 How to build a clock tree?	96



3 ST boards clock tree	97
3.1 STM32MP157x-EV1 case	97
3.1.1 Clock tree	97
3.1.1.1 Device tree	99
3.2 STM32MP157x-DK2 case	101
3.2.1 Clock tree	101
3.2.2 Device tree	103



1 Overview

The clock tree is managed via RCC internal peripheral hardware block and it is configured at different steps from the Cortex-A7:

- When the device is **reset**, all RCC registers take their reset values: the four PLL are disabled and most of the clock source selectors are pointing to the HSI.
- The ROM Code configures the minimum clock tree needed to boot on the selected boot device.
- The **FSBL** (Cf. [Boot chain overview](#)) completely configures the clock tree as expected, all the way up to Linux, thanks to the configuration given in the [device tree](#).
 - U-Boot SPL (basic boot chain) binding is available in [doc/device-tree-bindings/clock/st,stm32mp1.txt](#)
 - TF-A (trusted boot chain) binding follows the same syntax as U-Boot SPL
- **Linux** is responsible for the clock tree at runtime and it may partly modify it thanks to information taken from the [device tree](#):
 - Linux binding is available in [Documentation/devicetree/bindings/clock/clock-bindings.txt](#) (and surrounding files): 'fixed-clock' compatible, 'clocks' and 'assigned-clocks' properties are important concepts to understand the management of clocks providers/consumers.

Notice that [STM32Cube](#) (running on the Cortex-M4) does not control the clock tree setup so the configuration applied by the Cortex-A7 must provide suitable clocks to Cortex-M4, like it is explained in [resource manager](#) description. One exception to this rule is the [engineering boot](#), allowing to directly load (and debug) the Cortex-M4 whereas the Cortex-A7 execution is stalled in the ROM code.



2 How to build a clock tree?

Building a clock tree is quite complex as it needs to take into account the constraints set by each internal and external peripheral, including external clock sources.

We encourage the use of [STM32CubeMX](#) to build the clock tree, and avoid having to know all internal peripherals details: the tool allows to select the peripherals that will be present on the board, fix the clock sources frequencies and automatically find an optimized clock tree. It is then able to generate the device tree that is directly consumed by the boot chain and Linux.



3 ST boards clock tree



See [How to change the CPU frequency](#) article for more information about the CPU frequency setting (PLL1P), including dynamic voltage and frequency scaling (DVFS) configuration.

3.1 STM32MP157x-EV1 case

This chapter shows the boot time clock tree set by the FSBL on STM32MP157x-EV1 evaluation board. Linux eventual runtime modifications are not covered here.

3.1.1 Clock tree

The following table shows what STM32MP157x-EV1 clock tree looks like, as a result of the boot chain execution with the device tree built with STM32CubeMX.

Component Comment	Parent	Frequency	Used?	
LSI	N.A.	0.032000 MHz	yes	Mandatory for IWDG, DAC
DAC	LSI	0.032000 MHz	yes	
RNG1	LSI	0.032000 MHz	yes	Input frequency should be
as low as possible				
RNG2	LSI	0.032000 MHz	yes	Input frequency should be
as low as possible				
IWDG1	LSI	0.032000 MHz	yes	
IWDG2	LSI	0.032000 MHz	yes	
LSE	N.A.	0.032768 MHz	yes	Mandatory for DTS
RTC	LSE	0.032768 MHz	yes	
TAMP	LSE	0.032768 MHz	yes	
CEC	LSE	0.032768 MHz	yes	
LPTIM4	LSE	0.032768 MHz	yes	
LPTIM5	LSE	0.032768 MHz	yes	
HSI	N.A.	64.000000 MHz	yes	
SPI4	HSI	64.000000 MHz	no	
SPI5	HSI	64.000000 MHz	no	
SPI6	HSI	64.000000 MHz	yes	
I2C4	HSI	64.000000 MHz	yes	PMIC
I2C6	HSI	64.000000 MHz	no	
I2C1	HSI	64.000000 MHz	no	
I2C2	HSI	64.000000 MHz	yes	Rpi and peripherals
I2C3	HSI	64.000000 MHz	no	
I2C5	HSI	64.000000 MHz	yes	Rpi
USART1	HSI	64.000000 MHz	yes	
USART2	HSI	64.000000 MHz	no	
USART3	HSI	64.000000 MHz	yes	Rpi
UART4	HSI	64.000000 MHz	yes	Linux console
UART5	HSI	64.000000 MHz	no	
USART6	HSI	64.000000 MHz	no	
UART7	HSI	64.000000 MHz	no	
UART8	HSI	64.000000 MHz	no	
MC01	HSI	64.000000 MHz	no	Available so can be used
HSE	N.A.	24.000000 MHz	yes	
DSIPLL	HSE	125.000000 MHz	no	DSI DPHY PLL
DSIBL	DSIPLL	125.000000 MHz	no	DSI lanebyte clock
RTCDIV	HSE	1.000000 MHz	yes	Only used when RTC source
is HSE				



ck_per	HSE	24.000000	MHz	yes	
ADC	ck_per	24.000000	MHz	yes	Can use internal divider to
be < 40MHz					
PLL1	HSE	xxx	MHz	yes	
PLL1P	PLL1	xxx	MHz	yes	
MPUDIV	PLL1P	xxx	MHz	yes	
Cortex-A7	PLL1P	xxx	MHz	yes	650 MHz or 800 MHz, see the
information box above					
MC02	Cortex-A7	xxx	MHz	no	Available so can be used
PLL2	HSE	1066.000000	MHz	yes	
PLL2P	PLL2	266.500000	MHz	yes	
AXI	PLL2P	266.500000	MHz	yes	< 266MHz
FMC	AXI	266.500000	MHz	yes	NAND flash
QSPI	AXI	266.500000	MHz	yes	NOR flash
SYSRAM	AXI	266.500000	MHz	yes	
ROM	AXI	266.500000	MHz	yes	
AHB5	AXI	266.500000	MHz	yes	< 266MHz
CRYP1	AHB5	266.500000	MHz	yes	
HASH1	AHB5	266.500000	MHz	yes	
GPIOZ	AHB5	266.500000	MHz	yes	
BKPSRAM	AHB5	266.500000	MHz	yes	
AHB6	AXI	266.500000	MHz	yes	< 266MHz
CRC1	AHB6	266.500000	MHz	yes	
MDMA	AHB6	266.500000	MHz	yes	
USBH	AHB6	266.500000	MHz	yes	USB Host
APB4	AHB6	133.250000	MHz	yes	
APB5	AHB6	66.625000	MHz	yes	
BSEC	APB5	66.625000	MHz	yes	< 67MHz
ETZPC	APB5	66.625000	MHz	yes	
TZC	APB5	66.625000	MHz	yes	
DBGAPB	AXI	133.250000	MHz	yes	JTAG & Coresight
DBGMCU	DBGAPB	66.625000	MHz	yes	
STM	DBGAPB	66.625000	MHz	yes	
PLL2Q	PLL2	533.000000	MHz	yes	
GPU	PLL2Q	533.000000	MHz	yes	< 533MHz
PLL2R	PLL2	533.000000	MHz	yes	
DDR	PLL2R	533.000000	MHz	yes	< 533MHz
PLL3	HSE	417.755859	MHz	yes	
PLL3P	PLL3	208.877930	MHz	yes	
MLAHB	PLL3P	208.877930	MHz	yes	< 209MHz
Cortex-M4	MLAHB	208.877930	MHz	yes	
SRAM1	MLAHB	208.877930	MHz	yes	
SRAM2	MLAHB	208.877930	MHz	yes	
SRAM3	MLAHB	208.877930	MHz	yes	
RETRAM	MLAHB	208.877930	MHz	yes	
AHB1	MLAHB	208.877930	MHz	yes	< 209MHz
AHB2	MLAHB	208.877930	MHz	yes	< 209MHz
DMA1	AHB2	208.877930	MHz	yes	
DMA2	AHB2	208.877930	MHz	yes	
DMAMUX	AHB2	208.877930	MHz	yes	
APB1	MLAHB	104.438965	MHz	yes	
LPTIM1	APB1	104.438965	MHz	yes	
WWDG	APB1	104.438965	MHz	yes	
APB2	MLAHB	104.438965	MHz	yes	
TIM2	MLAHB	208.877930	MHz	yes	TIM Group 1
TIM3	MLAHB	208.877930	MHz	yes	TIM Group 1
TIM4	MLAHB	208.877930	MHz	yes	TIM Group 1
TIM5	MLAHB	208.877930	MHz	yes	TIM Group 1
TIM6	MLAHB	208.877930	MHz	yes	TIM Group 1
TIM7	MLAHB	208.877930	MHz	yes	TIM Group 1
TIM12	MLAHB	208.877930	MHz	yes	TIM Group 1
TIM13	MLAHB	208.877930	MHz	yes	TIM Group 1
TIM14	MLAHB	208.877930	MHz	yes	TIM Group 1
TIM1	MLAHB	208.877930	MHz	yes	TIM Group 2
TIM8	MLAHB	208.877930	MHz	yes	TIM Group 2
TIM15	MLAHB	208.877930	MHz	yes	TIM Group 2
TIM16	MLAHB	208.877930	MHz	yes	TIM Group 2



TIM17	MLAHB	208.877930	MHz	yes	TIM Group 2
APB3	MLAHB	104.438965	MHz	yes	
LPTIM2	APB3	104.438965	MHz	yes	
LPTIM3	APB3	104.438965	MHz	yes	
SYSCFG	APB3	104.438965	MHz	yes	
VREFBUF	APB3	104.438965	MHz	yes	
DTS	APB3	104.438965	MHz	yes	
HDP	APB3	104.438965	MHz	yes	
AHB3	MLAHB	208.877930	MHz	yes	< 209MHz
CRC2	AHB3	208.877930	MHz	yes	
CRYP2	AHB3	208.877930	MHz	yes	
HASH2	AHB3	208.877930	MHz	yes	
DCMI	AHB3	208.877930	MHz	yes	Camera
IPCC	AHB3	208.877930	MHz	yes	Mailbox
AHB4	MLAHB	208.877930	MHz	yes	< 209MHz
PWR	AHB4	208.877930	MHz	yes	
RCC	AHB4	208.877930	MHz	yes	
GPIOA-K	AHB4	208.877930	MHz	yes	
EXTI	AHB4	208.877930	MHz	yes	
PLL3Q	PLL3	24.573874	MHz	yes	
SPI1	PLL3Q	24.573874	MHz	no	
SPI2	PLL3Q	24.573874	MHz	no	
SPI3	PLL3Q	24.573874	MHz	no	
DFSDM	PLL3Q	24.573874	MHz	yes	Digital micro
SAI1	PLL3Q	24.573874	MHz	no	
SAI2	PLL3Q	24.573874	MHz	yes	AudCodec 48kHz (use PLL3R)
for 44.1kHz)					
SAI3	PLL3Q	24.573874	MHz	no	
SAI4	PLL3Q	24.573874	MHz	yes	SPDIF TX 48kHz (use PLL3R)
for 44.1kHz)					
PLL3R	PLL3	11.290699	MHz	yes	
PLL4	HSE	594.000000	MHz	yes	
PLL4P	PLL4	99.000000	MHz	yes	
SDMMC1	PLL4P	99.000000	MHz	yes	µSD card
SDMMC2	PLL4P	99.000000	MHz	yes	eMMC
SDMMC3	PLL4P	99.000000	MHz	yes	
SPDIF	PLL4P	99.000000	MHz	yes	SPDIF RX
PLL4Q	PLL4	74.250000	MHz	yes	
LCD	PLL4Q	74.250000	MHz	yes	LTDC & DSI display pixel
clock					
PLL4R	PLL4	74.250000	MHz	yes	
FDCAN	PLL4R	74.250000	MHz	yes	Should be as high as
possible and < 100MHz					
STGEN	HSE	24.000000	MHz	yes	
USBPHYC	HSE	24.000000	MHz	yes	USB PHY Ctrl for USB Host
and OTG					
USBPLL	USBPHYC	48.000000	MHz	yes	
USB0	USBPLL	48.000000	MHz	yes	USB OTG
CSI	N.A.	4.000000	MHz	yes	Mandatory for IO
compensation					
ETH	N.A.	0.000000	MHz	no	ETH clocked by RGMII PHY

3.1.1.1 Device tree

Here are the corresponding device tree `rcm_clk` sub node properties consumed by the first stage boot loader (FSBL) to configure the clock tree above:

```
st,clksrc = <
    CLK_MPU_PLL1P
    CLK_AXI_PLL2P
    CLK_MCU_PLL3P
    CLK_PLL12_HSE
```



```

        CLK_PLL3_HSE
        CLK_PLL4_HSE
        CLK_RTC_LSE
        CLK_MC01_DISABLED
        CLK_MC02_DISABLED
    >;

    st,clkdiv = <
        1 /*MPU*/
        0 /*AXI*/
        0 /*MCU*/
        1 /*APB1*/
        1 /*APB2*/
        1 /*APB3*/
        1 /*APB4*/
        2 /*APB5*/
        23 /*RTC*/
        0 /*MC01*/
        0 /*MC02*/
    >;

    st,pkcs = <
        CLK_CKPER_HSE
        CLK_FMC_ACLK
        CLK_QSPI_ACLK
        CLK_ETH_DISABLED
        CLK_SDMMC12_PLL4P
        CLK_DSI_DSIPLL
        CLK_STGEN_HSE
        CLK_USBPHY_HSE
        CLK_SPI2S1_PLL3Q
        CLK_SPI2S23_PLL3Q
        CLK_SPI45_HSI
        CLK_SPI6_HSI
        CLK_I2C46_HSI
        CLK_SDMMC3_PLL4P
        CLK_USB0_USBPHY
        CLK_ADC_CKPER
        CLK_CEC_LSE
        CLK_I2C12_HSI
        CLK_I2C35_HSI
        CLK_UART1_HSI
        CLK_UART24_HSI
        CLK_UART35_HSI
        CLK_UART6_HSI
        CLK_UART78_HSI
        CLK_SPDIF_PLL4P
        CLK_FDCAN_PLL4R
        CLK_SAI1_PLL3Q
        CLK_SAI2_PLL3Q
        CLK_SAI3_PLL3Q
        CLK_SAI4_PLL3Q
        CLK_RNG1_LSI
        CLK_RNG2_LSI
        CLK_LPTIM1_PCLK1
        CLK_LPTIM23_PCLK3
        CLK_LPTIM45_LSE
    >;

    /* VCO = 1066.0 MHz => P = 266 (AXI), Q = 533 (GPU), R = 533 (DDR) */
    pll2: st,pll@1 {
        cfg = < 2 65 1 0 0 PQR(1,1,1) >;
        frac = < 0x1400 >;
        u-boot,dm-pre-reloc;
    };

    /* VCO = 417.8 MHz => P = 209, Q = 25, R = 11 */

```



```

pll3: st,pll@2 {
    cfg = < 1 33 1 16 36 PQR(1,1,1) >;
    frac = < 0x1a04 >;
    u-boot,dm-pre-reloc;
};

/* VCO = 594.0 MHz => P = 99, Q = 74, R = 74 */
pll4: st,pll@3 {
    cfg = < 3 98 5 7 7 PQR(1,1,1) >;
    u-boot,dm-pre-reloc;
};
    
```

3.2 STM32MP157x-DK2 case

This chapter shows the boot time clock tree set by the FSBL on STM32MP157x-DK2 DISCO board. Linux eventual runtime modifications are not covered here.

3.2.1 Clock tree

The following table shows what STM32MP157x-DK2 clock tree looks like, as a result of the boot chain execution with the device tree built with STM32CubeMX.

Component Comment	Parent	Frequency	Used?	
LSI	N.A.	0.032000 MHz	yes	Mandatory for IWDG, DAC
DAC	LSI	0.032000 MHz	no	
RNG1	LSI	0.032000 MHz	yes	Input frequency should be as low as possible
RNG2	LSI	0.032000 MHz	yes	Input frequency should be as low as possible
IWDG1	LSI	0.032000 MHz	yes	
IWDG2	LSI	0.032000 MHz	yes	
LSE	N.A.	0.032768 MHz	yes	Mandatory for DTS
RTC	LSE	0.032768 MHz	yes	
TAMP	LSE	0.032768 MHz	yes	
CEC	LSE	0.032768 MHz	yes	
LPTIM4	LSE	0.032768 MHz	yes	
LPTIM5	LSE	0.032768 MHz	yes	
HSI	N.A.	64.000000 MHz	yes	
SPI4	HSI	64.000000 MHz	yes	Arduino
SPI5	HSI	64.000000 MHz	yes	Rpi
SPI6	HSI	64.000000 MHz	no	
I2C4	HSI	64.000000 MHz	yes	PMIC
I2C6	HSI	64.000000 MHz	no	
I2C1	HSI	64.000000 MHz	yes	Rpi and peripherals
I2C2	HSI	64.000000 MHz	no	
I2C3	HSI	64.000000 MHz	no	
I2C5	HSI	64.000000 MHz	yes	Rpi and Arduino
USART1	HSI	64.000000 MHz	no	
USART2	HSI	64.000000 MHz	yes	Bluetooth
USART3	HSI	64.000000 MHz	yes	Rpi
UART4	HSI	64.000000 MHz	yes	Linux console
UART5	HSI	64.000000 MHz	no	
USART6	HSI	64.000000 MHz	no	
UART7	HSI	64.000000 MHz	yes	Arduino
UART8	HSI	64.000000 MHz	no	
MCO1	HSI	64.000000 MHz	no	Available so can be used
HSE	N.A.	24.000000 MHz	yes	
DSIPLL	HSE	125.000000 MHz	no	DSI DPHY PLL
DSIBL	DSIPLL	125.000000 MHz	no	DSI lanebyte clock



RTCDIV	HSE	1.000000 MHz	yes	Only used when RTC source
is HSE				
ck_per	HSE	24.000000 MHz	yes	
ADC	ck_per	24.000000 MHz	yes	Can use internal divider to
be < 40MHz				
PLL1	HSE	xxx MHz	yes	
PLL1P	PLL1	xxx MHz	yes	
MPUDIV	PLL1P	xxx MHz	yes	
Cortex-A7	PLL1P	xxx MHz	yes	650 MHz or 800 MHz, see the
information box above				
MC02	Cortex-A7	xxx MHz	no	Available so can be used
PLL2	HSE	1066.000000 MHz	yes	
PLL2P	PLL2	266.500000 MHz	yes	
AXI	PLL2P	266.500000 MHz	yes	< 266MHz
FMC	AXI	266.500000 MHz	no	
QSPI	AXI	266.500000 MHz	no	
SYSRAM	AXI	266.500000 MHz	yes	
ROM	AXI	266.500000 MHz	yes	
AHB5	AXI	266.500000 MHz	yes	< 266MHz
CRYP1	AHB5	266.500000 MHz	yes	
HASH1	AHB5	266.500000 MHz	yes	
GPIOZ	AHB5	266.500000 MHz	yes	
BKPSRAM	AHB5	266.500000 MHz	yes	
AHB6	AXI	266.500000 MHz	yes	< 266MHz
CRC1	AHB6	266.500000 MHz	yes	
MDMA	AHB6	266.500000 MHz	yes	
USBH	AHB6	266.500000 MHz	yes	USB Host
APB4	AHB6	133.250000 MHz	yes	
APB5	AHB6	66.625000 MHz	yes	
BSEC	APB5	66.625000 MHz	yes	< 67MHz
ETZPC	APB5	66.625000 MHz	yes	
TZC	APB5	66.625000 MHz	yes	
DBGAPB	AXI	133.250000 MHz	yes	JTAG & Coresight
DBGMCU	DBGAPB	66.625000 MHz	yes	
STM	DBGAPB	66.625000 MHz	no	
PLL2Q	PLL2	533.000000 MHz	yes	
GPU	PLL2Q	533.000000 MHz	yes	< 533MHz
PLL2R	PLL2	533.000000 MHz	yes	
DDR	PLL2R	533.000000 MHz	yes	< 533MHz
PLL3	HSE	417.755859 MHz	yes	
PLL3P	PLL3	208.877930 MHz	yes	
MLAHB	PLL3P	208.877930 MHz	yes	< 209MHz
Cortex-M4	MLAHB	208.877930 MHz	yes	
SRAM1	MLAHB	208.877930 MHz	yes	
SRAM2	MLAHB	208.877930 MHz	yes	
SRAM3	MLAHB	208.877930 MHz	yes	
RETRAM	MLAHB	208.877930 MHz	yes	
AHB1	MLAHB	208.877930 MHz	yes	< 209MHz
AHB2	MLAHB	208.877930 MHz	yes	< 209MHz
DMA1	AHB2	208.877930 MHz	yes	
DMA2	AHB2	208.877930 MHz	yes	
DMAMUX	AHB2	208.877930 MHz	yes	
APB1	MLAHB	104.438965 MHz	yes	
LPTIM1	APB1	104.438965 MHz	yes	
WWDG	APB1	104.438965 MHz	yes	
APB2	MLAHB	104.438965 MHz	yes	
TIM2	MLAHB	208.877930 MHz	yes	TIM Group 1
TIM3	MLAHB	208.877930 MHz	yes	TIM Group 1
TIM4	MLAHB	208.877930 MHz	yes	TIM Group 1
TIM5	MLAHB	208.877930 MHz	yes	TIM Group 1
TIM6	MLAHB	208.877930 MHz	yes	TIM Group 1
TIM7	MLAHB	208.877930 MHz	yes	TIM Group 1
TIM12	MLAHB	208.877930 MHz	yes	TIM Group 1
TIM13	MLAHB	208.877930 MHz	yes	TIM Group 1
TIM14	MLAHB	208.877930 MHz	yes	TIM Group 1
TIM1	MLAHB	208.877930 MHz	yes	TIM Group 2
TIM8	MLAHB	208.877930 MHz	yes	TIM Group 2



TIM15	MLAHB	208.877930	MHz	yes	TIM Group 2
TIM16	MLAHB	208.877930	MHz	yes	TIM Group 2
TIM17	MLAHB	208.877930	MHz	yes	TIM Group 2
APB3	MLAHB	104.438965	MHz	yes	
LPTIM2	APB3	104.438965	MHz	yes	
LPTIM3	APB3	104.438965	MHz	yes	
SYSCFG	APB3	104.438965	MHz	yes	
VREFBUF	APB3	104.438965	MHz	yes	
DTS	APB3	104.438965	MHz	yes	
HDP	APB3	104.438965	MHz	no	
AHB3	MLAHB	208.877930	MHz	yes	< 209MHz
CRC2	AHB3	208.877930	MHz	yes	
CRYP2	AHB3	208.877930	MHz	yes	
HASH2	AHB3	208.877930	MHz	yes	
DCMI	AHB3	208.877930	MHz	no	
IPCC	AHB3	208.877930	MHz	yes	Mailbox
AHB4	MLAHB	208.877930	MHz	yes	< 209MHz
PWR	AHB4	208.877930	MHz	yes	
RCC	AHB4	208.877930	MHz	yes	
GPIOA-K	AHB4	208.877930	MHz	yes	
EXTI	AHB4	208.877930	MHz	yes	
PLL3Q	PLL3	24.573874	MHz	yes	
SPI1	PLL3Q	24.573874	MHz	yes	SPI2S1 for BT PCM
SPI2	PLL3Q	24.573874	MHz	yes	SPI2S2 for HDMI
SPI3	PLL3Q	24.573874	MHz	no	
DFSDM	PLL3Q	24.573874	MHz	no	
SAI1	PLL3Q	24.573874	MHz	no	
SAI2	PLL3Q	24.573874	MHz	yes	AudCodec 48kHz (use PLL3R)
for 44.1kHz)					
SAI3	PLL3Q	24.573874	MHz	no	
SAI4	PLL3Q	24.573874	MHz	no	
PLL3R	PLL3	11.290699	MHz	yes	
PLL4	HSE	594.000000	MHz	yes	
PLL4P	PLL4	99.000000	MHz	yes	
SDMMC1	PLL4P	99.000000	MHz	yes	µSD card
SDMMC2	PLL4P	99.000000	MHz	yes	Wifi
SDMMC3	PLL4P	99.000000	MHz	yes	Rpi
SPDIF	PLL4P	99.000000	MHz	no	
PLL4Q	PLL4	74.250000	MHz	yes	
LCD	PLL4Q	74.250000	MHz	yes	LTDC & DSI display pixel
clock					
PLL4R	PLL4	74.250000	MHz	yes	
FDCAN	PLL4R	74.250000	MHz	no	
STGEN	HSE	24.000000	MHz	yes	
USBPHYC	HSE	24.000000	MHz	yes	USB PHY Ctrl for USB Host
and OTG					
USBPLL	USBPHYC	48.000000	MHz	yes	
USB0	USBPLL	48.000000	MHz	yes	USB OTG
CSI	N.A.	4.000000	MHz	yes	Mandatory for IO
compensation					
ETH	N.A.	0.000000	MHz	no	ETH clocked by RGMII PHY

3.2.2 Device tree

Here are the corresponding device tree `rcc_clk` sub node properties consumed by the first stage boot loader (FSBL) to configure the clock tree above:

```

st,clksrc = <
    CLK_MPU_PLL1P
    CLK_AXI_PLL2P
    CLK_MCU_PLL3P
    CLK_PLL12_HSE
    
```



```

        CLK_PLL3_HSE
        CLK_PLL4_HSE
        CLK_RTC_LSE
        CLK_MC01_DISABLED
        CLK_MC02_DISABLED
    >;

    st,clkdiv = <
        1 /*MPU*/
        0 /*AXI*/
        0 /*MCU*/
        1 /*APB1*/
        1 /*APB2*/
        1 /*APB3*/
        1 /*APB4*/
        2 /*APB5*/
        23 /*RTC*/
        0 /*MC01*/
        0 /*MC02*/
    >;

    st,pkcs = <
        CLK_CKPER_HSE
        CLK_FMC_ACLK
        CLK_QSPI_ACLK
        CLK_ETH_DISABLED
        CLK_SDMMC12_PLL4P
        CLK_DSI_DSIPLL
        CLK_STGEN_HSE
        CLK_USBPHY_HSE
        CLK_SPI2S1_PLL3Q
        CLK_SPI2S23_PLL3Q
        CLK_SPI45_HSI
        CLK_SPI6_HSI
        CLK_I2C46_HSI
        CLK_SDMMC3_PLL4P
        CLK_USB0_USBPHY
        CLK_ADC_CKPER
        CLK_CEC_LSE
        CLK_I2C12_HSI
        CLK_I2C35_HSI
        CLK_UART1_HSI
        CLK_UART24_HSI
        CLK_UART35_HSI
        CLK_UART6_HSI
        CLK_UART78_HSI
        CLK_SPDIF_PLL4P
        CLK_FDCAN_PLL4R
        CLK_SAI1_PLL3Q
        CLK_SAI2_PLL3Q
        CLK_SAI3_PLL3Q
        CLK_SAI4_PLL3Q
        CLK_RNG1_LSI
        CLK_RNG2_LSI
        CLK_LPTIM1_PCLK1
        CLK_LPTIM23_PCLK3
        CLK_LPTIM45_LSE
    >;

    /* VCO = 1066.0 MHz => P = 266 (AXI), Q = 533 (GPU), R = 533 (DDR) */
    pll2: st,pll@1 {
        cfg = < 2 65 1 0 0 PQR(1,1,1) >;
        frac = < 0x1400 >;
        u-boot,dm-pre-reloc;
    };

    /* VCO = 417.8 MHz => P = 209, Q = 25, R = 11 */

```




```

pll3: st,pll@2 {
    cfg = < 1 33 1 16 36 PQR(1,1,1) >;
    frac = < 0x1a04 >;
    u-boot,dm-pre-reloc;
};

/* VCO = 594.0 MHz => P = 99, Q = 74, R = 74 */
pll4: st,pll@3 {
    cfg = < 3 98 5 7 7 PQR(1,1,1) >;
    u-boot,dm-pre-reloc;
};

```

Reset and Clock Control

Low Speed Internal oscillator (STM32 clock source)

Low Speed External oscillator (STM32 clock source)

High Speed Internal oscillator (STM32 clock source) or High Speed Synchronous Serial Interface (MIPI® Alliance standard)

High Speed External oscillator (STM32 clock source)

Multi Speed Internal oscillator (STM32 clock source)

Cortex®

First Stage Boot Loader

Linux® is a registered trademark of Linus Torvalds.

Das U-Boot -- the Universal Boot Loader (see [U-Boot_overview](#))

Secondary Program Loader, *Also known as **U-Boot SPL***

Read Only Memory

Central processing unit

Dynamic voltage and frequency scaling

Independent Watchdog

Digital-to-analog converter (Electronic circuit that converts a binary number into a continuously varying value.)

Device Tree Source (in software context) or Digital Temperature Sensor (in peripheral context)

Real Time Clock

Tamper

Consumer Electronics Control (HDMI standard)

Power Management Integrated Circuit

Display Serial Interface (MIPI® Alliance standard)

Analog-to-digital converter. The process of converting a sampled analog signal to a digital code that represents the amplitude of the original signal sample.

USB Host (STM32 specific)

Boot and Security and OTP control

Extended TrustZone Protection Controller



TrustZone[®] address space Controller for DDR

Arm[®] and TrustZone[®] are registered trademarks of Arm Limited (or its subsidiaries) in the US and/or elsewhere.

debug and test protocol, named from the Joint Test Action Group that developed it

System Trace Module

Graphics Processing Units

Doubledata rate (memory domain)

System Configuration

voltage reference buffer (STM32 specific)

Hardware Debug Port

Digital Camera Memory Interface

Inter-Processor Communication Controller

External Interrupt

Digital Filter for Sigma-Delta Modulator

Transmit

former spelling for e•MMC ('e' in italic)

Receive

LCD TFT Display Controller (STM32 specific)

System Time Generator

USB On-The-Go (Capability/type of USB port, acting primarily as USB device, to also act as USB host. Also known as USB OTG.)

input/output

Ethernet

Microprocessor Unit

Microcontroller Unit (MCUs have internal flash memory and are intended to operate with a minimum amount of external support ICs. They commonly are a self-contained, system-on-chip (SoC) designs.)

Discovery kit

BlueTooth

High-Definition Multimedia Interface (HDMI standard)

Stable: 23.09.2020 - 13:22 / Revision: 12.06.2020 - 13:25

All the peripherals receive one or several clocks that are generated via [RCC internal peripheral](#). RCC relies on several clocks sources (LSI, LSE, HSI, HSE, CSI) and four PLL in order to provide adequate input frequencies to all the peripherals. The clock tree covers all the system clock distribution aspects, from the clock sources to the consumer peripherals (internal and external), except clock gating management that is locally controlled by each peripheral driver.

Contents

1 Overview	108
2 How to build a clock tree?	109



3 ST boards clock tree	110
3.1 STM32MP157x-EV1 case	110
3.1.1 Clock tree	110
3.1.1.1 Device tree	112
3.2 STM32MP157x-DK2 case	114
3.2.1 Clock tree	114
3.2.2 Device tree	116



1 Overview

The clock tree is managed via RCC internal peripheral hardware block and it is configured at different steps from the Cortex-A7:

- When the device is **reset**, all RCC registers take their reset values: the four PLL are disabled and most of the clock source selectors are pointing to the HSI.
- The ROM Code configures the minimum clock tree needed to boot on the selected boot device.
- The **FSBL** (Cf. [Boot chain overview](#)) completely configures the clock tree as expected, all the way up to Linux, thanks to the configuration given in the [device tree](#).
 - U-Boot SPL (basic boot chain) binding is available in [doc/device-tree-bindings/clock/st,stm32mp1.txt](#)
 - TF-A (trusted boot chain) binding follows the same syntax as U-Boot SPL
- **Linux** is responsible for the clock tree at runtime and it may partly modify it thanks to information taken from the [device tree](#):
 - Linux binding is available in [Documentation/devicetree/bindings/clock/clock-bindings.txt](#) (and surrounding files): 'fixed-clock' compatible, 'clocks' and 'assigned-clocks' properties are important concepts to understand the management of clocks providers/consumers.

Notice that [STM32Cube](#) (running on the Cortex-M4) does not control the clock tree setup so the configuration applied by the Cortex-A7 must provide suitable clocks to Cortex-M4, like it is explained in [resource manager](#) description. One exception to this rule is the [engineering boot](#), allowing to directly load (and debug) the Cortex-M4 whereas the Cortex-A7 execution is stalled in the ROM code.



2 How to build a clock tree?

Building a clock tree is quite complex as it needs to take into account the constraints set by each internal and external peripheral, including external clock sources.

We encourage the use of [STM32CubeMX](#) to build the clock tree, and avoid having to know all internal peripherals details: the tool allows to select the peripherals that will be present on the board, fix the clock sources frequencies and automatically find an optimized clock tree. It is then able to generate the device tree that is directly consumed by the boot chain and Linux.



3 ST boards clock tree

 See [How to change the CPU frequency](#) article for more information about the CPU frequency setting (PLL1P), including dynamic voltage and frequency scaling (DVFS) configuration.

3.1 STM32MP157x-EV1 case

This chapter shows the boot time clock tree set by the FSBL on STM32MP157x-EV1 evaluation board. Linux eventual runtime modifications are not covered here.

3.1.1 Clock tree

The following table shows what STM32MP157x-EV1 clock tree looks like, as a result of the boot chain execution with the device tree built with STM32CubeMX.

Component Comment	Parent	Frequency	Used?	
LSI	N.A.	0.032000 MHz	yes	Mandatory for IWDG, DAC
DAC	LSI	0.032000 MHz	yes	
RNG1	LSI	0.032000 MHz	yes	Input frequency should be
as low as possible				
RNG2	LSI	0.032000 MHz	yes	Input frequency should be
as low as possible				
IWDG1	LSI	0.032000 MHz	yes	
IWDG2	LSI	0.032000 MHz	yes	
LSE	N.A.	0.032768 MHz	yes	Mandatory for DTS
RTC	LSE	0.032768 MHz	yes	
TAMP	LSE	0.032768 MHz	yes	
CEC	LSE	0.032768 MHz	yes	
LPTIM4	LSE	0.032768 MHz	yes	
LPTIM5	LSE	0.032768 MHz	yes	
HSI	N.A.	64.000000 MHz	yes	
SPI4	HSI	64.000000 MHz	no	
SPI5	HSI	64.000000 MHz	no	
SPI6	HSI	64.000000 MHz	yes	
I2C4	HSI	64.000000 MHz	yes	PMIC
I2C6	HSI	64.000000 MHz	no	
I2C1	HSI	64.000000 MHz	no	
I2C2	HSI	64.000000 MHz	yes	Rpi and peripherals
I2C3	HSI	64.000000 MHz	no	
I2C5	HSI	64.000000 MHz	yes	Rpi
USART1	HSI	64.000000 MHz	yes	
USART2	HSI	64.000000 MHz	no	
USART3	HSI	64.000000 MHz	yes	Rpi
UART4	HSI	64.000000 MHz	yes	Linux console
UART5	HSI	64.000000 MHz	no	
USART6	HSI	64.000000 MHz	no	
UART7	HSI	64.000000 MHz	no	
UART8	HSI	64.000000 MHz	no	
MC01	HSI	64.000000 MHz	no	Available so can be used
HSE	N.A.	24.000000 MHz	yes	
DSIPLL	HSE	125.000000 MHz	no	DSI DPHY PLL
DSIBL	DSIPLL	125.000000 MHz	no	DSI lanebyte clock
RTCDIV	HSE	1.000000 MHz	yes	Only used when RTC source
is HSE				



ck_per	HSE	24.000000	MHz	yes	
ADC	ck_per	24.000000	MHz	yes	Can use internal divider to
be < 40MHz					
PLL1	HSE	xxx	MHz	yes	
PLL1P	PLL1	xxx	MHz	yes	
MPUDIV	PLL1P	xxx	MHz	yes	
Cortex-A7	PLL1P	xxx	MHz	yes	650 MHz or 800 MHz, see the
information box above					
MC02	Cortex-A7	xxx	MHz	no	Available so can be used
PLL2	HSE	1066.000000	MHz	yes	
PLL2P	PLL2	266.500000	MHz	yes	
AXI	PLL2P	266.500000	MHz	yes	< 266MHz
FMC	AXI	266.500000	MHz	yes	NAND flash
QSPI	AXI	266.500000	MHz	yes	NOR flash
SYSRAM	AXI	266.500000	MHz	yes	
ROM	AXI	266.500000	MHz	yes	
AHB5	AXI	266.500000	MHz	yes	< 266MHz
CRYP1	AHB5	266.500000	MHz	yes	
HASH1	AHB5	266.500000	MHz	yes	
GPI0Z	AHB5	266.500000	MHz	yes	
BKPSRAM	AHB5	266.500000	MHz	yes	
AHB6	AXI	266.500000	MHz	yes	< 266MHz
CRC1	AHB6	266.500000	MHz	yes	
MDMA	AHB6	266.500000	MHz	yes	
USBH	AHB6	266.500000	MHz	yes	USB Host
APB4	AHB6	133.250000	MHz	yes	
APB5	AHB6	66.625000	MHz	yes	
BSEC	APB5	66.625000	MHz	yes	< 67MHz
ETZPC	APB5	66.625000	MHz	yes	
TZC	APB5	66.625000	MHz	yes	
DBGAPB	AXI	133.250000	MHz	yes	JTAG & Coresight
DBGMCU	DBGAPB	66.625000	MHz	yes	
STM	DBGAPB	66.625000	MHz	yes	
PLL2Q	PLL2	533.000000	MHz	yes	
GPU	PLL2Q	533.000000	MHz	yes	< 533MHz
PLL2R	PLL2	533.000000	MHz	yes	
DDR	PLL2R	533.000000	MHz	yes	< 533MHz
PLL3	HSE	417.755859	MHz	yes	
PLL3P	PLL3	208.877930	MHz	yes	
MLAHB	PLL3P	208.877930	MHz	yes	< 209MHz
Cortex-M4	MLAHB	208.877930	MHz	yes	
SRAM1	MLAHB	208.877930	MHz	yes	
SRAM2	MLAHB	208.877930	MHz	yes	
SRAM3	MLAHB	208.877930	MHz	yes	
RETRAM	MLAHB	208.877930	MHz	yes	
AHB1	MLAHB	208.877930	MHz	yes	< 209MHz
AHB2	MLAHB	208.877930	MHz	yes	< 209MHz
DMA1	AHB2	208.877930	MHz	yes	
DMA2	AHB2	208.877930	MHz	yes	
DMAMUX	AHB2	208.877930	MHz	yes	
APB1	MLAHB	104.438965	MHz	yes	
LPTIM1	APB1	104.438965	MHz	yes	
WWDG	APB1	104.438965	MHz	yes	
APB2	MLAHB	104.438965	MHz	yes	
TIM2	MLAHB	208.877930	MHz	yes	TIM Group 1
TIM3	MLAHB	208.877930	MHz	yes	TIM Group 1
TIM4	MLAHB	208.877930	MHz	yes	TIM Group 1
TIM5	MLAHB	208.877930	MHz	yes	TIM Group 1
TIM6	MLAHB	208.877930	MHz	yes	TIM Group 1
TIM7	MLAHB	208.877930	MHz	yes	TIM Group 1
TIM12	MLAHB	208.877930	MHz	yes	TIM Group 1
TIM13	MLAHB	208.877930	MHz	yes	TIM Group 1
TIM14	MLAHB	208.877930	MHz	yes	TIM Group 1
TIM1	MLAHB	208.877930	MHz	yes	TIM Group 2
TIM8	MLAHB	208.877930	MHz	yes	TIM Group 2
TIM15	MLAHB	208.877930	MHz	yes	TIM Group 2
TIM16	MLAHB	208.877930	MHz	yes	TIM Group 2



TIM17	MLAHB	208.877930	MHz	yes	TIM Group 2
APB3	MLAHB	104.438965	MHz	yes	
LPTIM2	APB3	104.438965	MHz	yes	
LPTIM3	APB3	104.438965	MHz	yes	
SYSCFG	APB3	104.438965	MHz	yes	
VREFBUF	APB3	104.438965	MHz	yes	
DTS	APB3	104.438965	MHz	yes	
HDP	APB3	104.438965	MHz	yes	
AHB3	MLAHB	208.877930	MHz	yes	< 209MHz
CRC2	AHB3	208.877930	MHz	yes	
CRYP2	AHB3	208.877930	MHz	yes	
HASH2	AHB3	208.877930	MHz	yes	
DCMI	AHB3	208.877930	MHz	yes	Camera
IPCC	AHB3	208.877930	MHz	yes	Mailbox
AHB4	MLAHB	208.877930	MHz	yes	< 209MHz
PWR	AHB4	208.877930	MHz	yes	
RCC	AHB4	208.877930	MHz	yes	
GPIOA-K	AHB4	208.877930	MHz	yes	
EXTI	AHB4	208.877930	MHz	yes	
PLL3Q	PLL3	24.573874	MHz	yes	
SPI1	PLL3Q	24.573874	MHz	no	
SPI2	PLL3Q	24.573874	MHz	no	
SPI3	PLL3Q	24.573874	MHz	no	
DFSDM	PLL3Q	24.573874	MHz	yes	Digital micro
SAI1	PLL3Q	24.573874	MHz	no	
SAI2	PLL3Q	24.573874	MHz	yes	AudCodec 48kHz (use PLL3R
for 44.1kHz)					
SAI3	PLL3Q	24.573874	MHz	no	
SAI4	PLL3Q	24.573874	MHz	yes	SPDIF TX 48kHz (use PLL3R
for 44.1kHz)					
PLL3R	PLL3	11.290699	MHz	yes	
PLL4	HSE	594.000000	MHz	yes	
PLL4P	PLL4	99.000000	MHz	yes	
SDMMC1	PLL4P	99.000000	MHz	yes	µSD card
SDMMC2	PLL4P	99.000000	MHz	yes	eMMC
SDMMC3	PLL4P	99.000000	MHz	yes	
SPDIF	PLL4P	99.000000	MHz	yes	SPDIF RX
PLL4Q	PLL4	74.250000	MHz	yes	
LCD	PLL4Q	74.250000	MHz	yes	LTDC & DSI display pixel
clock					
PLL4R	PLL4	74.250000	MHz	yes	
FDCAN	PLL4R	74.250000	MHz	yes	Should be as high as
possible and < 100MHz					
STGEN	HSE	24.000000	MHz	yes	
USBPHYC	HSE	24.000000	MHz	yes	USB PHY Ctrl for USB Host
and OTG					
USBPLL	USBPHYC	48.000000	MHz	yes	
USB0	USBPLL	48.000000	MHz	yes	USB OTG
CSI	N.A.	4.000000	MHz	yes	Mandatory for IO
compensation					
ETH	N.A.	0.000000	MHz	no	ETH clocked by RGMII PHY

3.1.1.1 Device tree

Here are the corresponding device tree `rcc_clk` sub node properties consumed by the first stage boot loader (FSBL) to configure the clock tree above:

```
st,clksrc = <
    CLK_MPU_PLL1P
    CLK_AXI_PLL2P
    CLK_MCU_PLL3P
    CLK_PLL12_HSE
```




```

        CLK_PLL3_HSE
        CLK_PLL4_HSE
        CLK_RTC_LSE
        CLK_MC01_DISABLED
        CLK_MC02_DISABLED
    >;

    st,clkdiv = <
        1 /*MPU*/
        0 /*AXI*/
        0 /*MCU*/
        1 /*APB1*/
        1 /*APB2*/
        1 /*APB3*/
        1 /*APB4*/
        2 /*APB5*/
        23 /*RTC*/
        0 /*MC01*/
        0 /*MC02*/
    >;

    st,pkcs = <
        CLK_CKPER_HSE
        CLK_FMC_ACLK
        CLK_QSPI_ACLK
        CLK_ETH_DISABLED
        CLK_SDMMC12_PLL4P
        CLK_DSI_DSIPLL
        CLK_STGEN_HSE
        CLK_USBPHY_HSE
        CLK_SPI2S1_PLL3Q
        CLK_SPI2S23_PLL3Q
        CLK_SPI45_HSI
        CLK_SPI6_HSI
        CLK_I2C46_HSI
        CLK_SDMMC3_PLL4P
        CLK_USB0_USBPHY
        CLK_ADC_CKPER
        CLK_CEC_LSE
        CLK_I2C12_HSI
        CLK_I2C35_HSI
        CLK_UART1_HSI
        CLK_UART24_HSI
        CLK_UART35_HSI
        CLK_UART6_HSI
        CLK_UART78_HSI
        CLK_SPDIF_PLL4P
        CLK_FDCAN_PLL4R
        CLK_SAI1_PLL3Q
        CLK_SAI2_PLL3Q
        CLK_SAI3_PLL3Q
        CLK_SAI4_PLL3Q
        CLK_RNG1_LSI
        CLK_RNG2_LSI
        CLK_LPTIM1_PCLK1
        CLK_LPTIM23_PCLK3
        CLK_LPTIM45_LSE
    >;

    /* VCO = 1066.0 MHz => P = 266 (AXI), Q = 533 (GPU), R = 533 (DDR) */
    pll2: st,pll@1 {
        cfg = < 2 65 1 0 0 PQR(1,1,1) >;
        frac = < 0x1400 >;
        u-boot,dm-pre-reloc;
    };

    /* VCO = 417.8 MHz => P = 209, Q = 25, R = 11 */

```



```

pll3: st,pll@2 {
    cfg = < 1 33 1 16 36 PQR(1,1,1) >;
    frac = < 0x1a04 >;
    u-boot,dm-pre-reloc;
};

/* VCO = 594.0 MHz => P = 99, Q = 74, R = 74 */
pll4: st,pll@3 {
    cfg = < 3 98 5 7 7 PQR(1,1,1) >;
    u-boot,dm-pre-reloc;
};
    
```

3.2 STM32MP157x-DK2 case

This chapter shows the boot time clock tree set by the FSBL on STM32MP157x-DK2 DISCO board. Linux eventual runtime modifications are not covered here.

3.2.1 Clock tree

The following table shows what STM32MP157x-DK2 clock tree looks like, as a result of the boot chain execution with the device tree built with STM32CubeMX.

Component Comment	Parent	Frequency	Used?	
LSI	N.A.	0.032000 MHz	yes	Mandatory for IWDG, DAC
DAC	LSI	0.032000 MHz	no	
RNG1	LSI	0.032000 MHz	yes	Input frequency should be as low as possible
RNG2	LSI	0.032000 MHz	yes	Input frequency should be as low as possible
IWDG1	LSI	0.032000 MHz	yes	
IWDG2	LSI	0.032000 MHz	yes	
LSE	N.A.	0.032768 MHz	yes	Mandatory for DTS
RTC	LSE	0.032768 MHz	yes	
TAMP	LSE	0.032768 MHz	yes	
CEC	LSE	0.032768 MHz	yes	
LPTIM4	LSE	0.032768 MHz	yes	
LPTIM5	LSE	0.032768 MHz	yes	
HSI	N.A.	64.000000 MHz	yes	
SPI4	HSI	64.000000 MHz	yes	Arduino
SPI5	HSI	64.000000 MHz	yes	Rpi
SPI6	HSI	64.000000 MHz	no	
I2C4	HSI	64.000000 MHz	yes	PMIC
I2C6	HSI	64.000000 MHz	no	
I2C1	HSI	64.000000 MHz	yes	Rpi and peripherals
I2C2	HSI	64.000000 MHz	no	
I2C3	HSI	64.000000 MHz	no	
I2C5	HSI	64.000000 MHz	yes	Rpi and Arduino
USART1	HSI	64.000000 MHz	no	
USART2	HSI	64.000000 MHz	yes	Bluetooth
USART3	HSI	64.000000 MHz	yes	Rpi
UART4	HSI	64.000000 MHz	yes	Linux console
UART5	HSI	64.000000 MHz	no	
USART6	HSI	64.000000 MHz	no	
UART7	HSI	64.000000 MHz	yes	Arduino
UART8	HSI	64.000000 MHz	no	
MCO1	HSI	64.000000 MHz	no	Available so can be used
HSE	N.A.	24.000000 MHz	yes	
DSIPLL	HSE	125.000000 MHz	no	DSI DPHY PLL
DSIBL	DSIPLL	125.000000 MHz	no	DSI lanebyte clock



RTCDIV	HSE	1.000000 MHz	yes	Only used when RTC source
is HSE				
ck_per	HSE	24.000000 MHz	yes	
ADC	ck_per	24.000000 MHz	yes	Can use internal divider to
be < 40MHz				
PLL1	HSE	xxx MHz	yes	
PLL1P	PLL1	xxx MHz	yes	
MPUDIV	PLL1P	xxx MHz	yes	
Cortex-A7	PLL1P	xxx MHz	yes	650 MHz or 800 MHz, see the
information box above				
MC02	Cortex-A7	xxx MHz	no	Available so can be used
PLL2	HSE	1066.000000 MHz	yes	
PLL2P	PLL2	266.500000 MHz	yes	
AXI	PLL2P	266.500000 MHz	yes	< 266MHz
FMC	AXI	266.500000 MHz	no	
QSPI	AXI	266.500000 MHz	no	
SYSRAM	AXI	266.500000 MHz	yes	
ROM	AXI	266.500000 MHz	yes	
AHB5	AXI	266.500000 MHz	yes	< 266MHz
CRYP1	AHB5	266.500000 MHz	yes	
HASH1	AHB5	266.500000 MHz	yes	
GPIOZ	AHB5	266.500000 MHz	yes	
BKPSRAM	AHB5	266.500000 MHz	yes	
AHB6	AXI	266.500000 MHz	yes	< 266MHz
CRC1	AHB6	266.500000 MHz	yes	
MDMA	AHB6	266.500000 MHz	yes	
USBH	AHB6	266.500000 MHz	yes	USB Host
APB4	AHB6	133.250000 MHz	yes	
APB5	AHB6	66.625000 MHz	yes	
BSEC	APB5	66.625000 MHz	yes	< 67MHz
ETZPC	APB5	66.625000 MHz	yes	
TZC	APB5	66.625000 MHz	yes	
DBGAPB	AXI	133.250000 MHz	yes	JTAG & Coresight
DBGMCU	DBGAPB	66.625000 MHz	yes	
STM	DBGAPB	66.625000 MHz	no	
PLL2Q	PLL2	533.000000 MHz	yes	
GPU	PLL2Q	533.000000 MHz	yes	< 533MHz
PLL2R	PLL2	533.000000 MHz	yes	
DDR	PLL2R	533.000000 MHz	yes	< 533MHz
PLL3	HSE	417.755859 MHz	yes	
PLL3P	PLL3	208.877930 MHz	yes	
MLAHB	PLL3P	208.877930 MHz	yes	< 209MHz
Cortex-M4	MLAHB	208.877930 MHz	yes	
SRAM1	MLAHB	208.877930 MHz	yes	
SRAM2	MLAHB	208.877930 MHz	yes	
SRAM3	MLAHB	208.877930 MHz	yes	
RETRAM	MLAHB	208.877930 MHz	yes	
AHB1	MLAHB	208.877930 MHz	yes	< 209MHz
AHB2	MLAHB	208.877930 MHz	yes	< 209MHz
DMA1	AHB2	208.877930 MHz	yes	
DMA2	AHB2	208.877930 MHz	yes	
DMAMUX	AHB2	208.877930 MHz	yes	
APB1	MLAHB	104.438965 MHz	yes	
LPTIM1	APB1	104.438965 MHz	yes	
WWDG	APB1	104.438965 MHz	yes	
APB2	MLAHB	104.438965 MHz	yes	
TIM2	MLAHB	208.877930 MHz	yes	TIM Group 1
TIM3	MLAHB	208.877930 MHz	yes	TIM Group 1
TIM4	MLAHB	208.877930 MHz	yes	TIM Group 1
TIM5	MLAHB	208.877930 MHz	yes	TIM Group 1
TIM6	MLAHB	208.877930 MHz	yes	TIM Group 1
TIM7	MLAHB	208.877930 MHz	yes	TIM Group 1
TIM12	MLAHB	208.877930 MHz	yes	TIM Group 1
TIM13	MLAHB	208.877930 MHz	yes	TIM Group 1
TIM14	MLAHB	208.877930 MHz	yes	TIM Group 1
TIM1	MLAHB	208.877930 MHz	yes	TIM Group 2
TIM8	MLAHB	208.877930 MHz	yes	TIM Group 2



TIM15	MLAHB	208.877930	MHz	yes	TIM Group 2
TIM16	MLAHB	208.877930	MHz	yes	TIM Group 2
TIM17	MLAHB	208.877930	MHz	yes	TIM Group 2
APB3	MLAHB	104.438965	MHz	yes	
LPTIM2	APB3	104.438965	MHz	yes	
LPTIM3	APB3	104.438965	MHz	yes	
SYSCFG	APB3	104.438965	MHz	yes	
VREFBUF	APB3	104.438965	MHz	yes	
DTS	APB3	104.438965	MHz	yes	
HDP	APB3	104.438965	MHz	no	
AHB3	MLAHB	208.877930	MHz	yes	< 209MHz
CRC2	AHB3	208.877930	MHz	yes	
CRYP2	AHB3	208.877930	MHz	yes	
HASH2	AHB3	208.877930	MHz	yes	
DCMI	AHB3	208.877930	MHz	no	
IPCC	AHB3	208.877930	MHz	yes	Mailbox
AHB4	MLAHB	208.877930	MHz	yes	< 209MHz
PWR	AHB4	208.877930	MHz	yes	
RCC	AHB4	208.877930	MHz	yes	
GPIOA-K	AHB4	208.877930	MHz	yes	
EXTI	AHB4	208.877930	MHz	yes	
PLL3Q	PLL3	24.573874	MHz	yes	
SPI1	PLL3Q	24.573874	MHz	yes	SPI2S1 for BT PCM
SPI2	PLL3Q	24.573874	MHz	yes	SPI2S2 for HDMI
SPI3	PLL3Q	24.573874	MHz	no	
DFSDM	PLL3Q	24.573874	MHz	no	
SAI1	PLL3Q	24.573874	MHz	no	
SAI2	PLL3Q	24.573874	MHz	yes	AudCodec 48kHz (use PLL3R)
for 44.1kHz)					
SAI3	PLL3Q	24.573874	MHz	no	
SAI4	PLL3Q	24.573874	MHz	no	
PLL3R	PLL3	11.290699	MHz	yes	
PLL4	HSE	594.000000	MHz	yes	
PLL4P	PLL4	99.000000	MHz	yes	
SDMMC1	PLL4P	99.000000	MHz	yes	µSD card
SDMMC2	PLL4P	99.000000	MHz	yes	Wifi
SDMMC3	PLL4P	99.000000	MHz	yes	Rpi
SPDIF	PLL4P	99.000000	MHz	no	
PLL4Q	PLL4	74.250000	MHz	yes	
LCD	PLL4Q	74.250000	MHz	yes	LTDC & DSI display pixel
clock					
PLL4R	PLL4	74.250000	MHz	yes	
FDCAN	PLL4R	74.250000	MHz	no	
STGEN	HSE	24.000000	MHz	yes	
USBPHYC	HSE	24.000000	MHz	yes	USB PHY Ctrl for USB Host
and OTG					
USBPLL	USBPHYC	48.000000	MHz	yes	
USB0	USBPLL	48.000000	MHz	yes	USB OTG
CSI	N.A.	4.000000	MHz	yes	Mandatory for IO
compensation					
ETH	N.A.	0.000000	MHz	no	ETH clocked by RGMII PHY

3.2.2 Device tree

Here are the corresponding device tree `rcc_clk` sub node properties consumed by the first stage boot loader (FSBL) to configure the clock tree above:

```
st,clksrc = <
    CLK_MPU_PLL1P
    CLK_AXI_PLL2P
    CLK_MCU_PLL3P
    CLK_PLL12_HSE
```



```

        CLK_PLL3_HSE
        CLK_PLL4_HSE
        CLK_RTC_LSE
        CLK_MC01_DISABLED
        CLK_MC02_DISABLED
    >;

    st,clkdiv = <
        1 /*MPU*/
        0 /*AXI*/
        0 /*MCU*/
        1 /*APB1*/
        1 /*APB2*/
        1 /*APB3*/
        1 /*APB4*/
        2 /*APB5*/
        23 /*RTC*/
        0 /*MC01*/
        0 /*MC02*/
    >;

    st,pkcs = <
        CLK_CKPER_HSE
        CLK_FMC_ACLK
        CLK_QSPI_ACLK
        CLK_ETH_DISABLED
        CLK_SDMMC12_PLL4P
        CLK_DSI_DSIPLL
        CLK_STGEN_HSE
        CLK_USBPHY_HSE
        CLK_SPI2S1_PLL3Q
        CLK_SPI2S23_PLL3Q
        CLK_SPI45_HSI
        CLK_SPI6_HSI
        CLK_I2C46_HSI
        CLK_SDMMC3_PLL4P
        CLK_USB0_USBPHY
        CLK_ADC_CKPER
        CLK_CEC_LSE
        CLK_I2C12_HSI
        CLK_I2C35_HSI
        CLK_UART1_HSI
        CLK_UART24_HSI
        CLK_UART35_HSI
        CLK_UART6_HSI
        CLK_UART78_HSI
        CLK_SPDIF_PLL4P
        CLK_FDCAN_PLL4R
        CLK_SAI1_PLL3Q
        CLK_SAI2_PLL3Q
        CLK_SAI3_PLL3Q
        CLK_SAI4_PLL3Q
        CLK_RNG1_LSI
        CLK_RNG2_LSI
        CLK_LPTIM1_PCLK1
        CLK_LPTIM23_PCLK3
        CLK_LPTIM45_LSE
    >;

    /* VCO = 1066.0 MHz => P = 266 (AXI), Q = 533 (GPU), R = 533 (DDR) */
    pll2: st,pll@1 {
        cfg = < 2 65 1 0 0 PQR(1,1,1) >;
        frac = < 0x1400 >;
        u-boot,dm-pre-reloc;
    };

    /* VCO = 417.8 MHz => P = 209, Q = 25, R = 11 */

```



```

pll3: st,pll@2 {
    cfg = < 1 33 1 16 36 PQR(1,1,1) >;
    frac = < 0x1a04 >;
    u-boot,dm-pre-reloc;
};

/* VCO = 594.0 MHz => P = 99, Q = 74, R = 74 */
pll4: st,pll@3 {
    cfg = < 3 98 5 7 7 PQR(1,1,1) >;
    u-boot,dm-pre-reloc;
};

```

Reset and Clock Control

Low Speed Internal oscillator (STM32 clock source)

Low Speed External oscillator (STM32 clock source)

High Speed Internal oscillator (STM32 clock source) or High Speed Synchronous Serial Interface (MIPI® Alliance standard)

High Speed External oscillator (STM32 clock source)

Multi Speed Internal oscillator (STM32 clock source)

Cortex®

First Stage Boot Loader

Linux® is a registered trademark of Linus Torvalds.

Das U-Boot -- the Universal Boot Loader (see [U-Boot_overview](#))

Secondary Program Loader, *Also known as **U-Boot SPL***

Read Only Memory

Central processing unit

Dynamic voltage and frequency scaling

Independent Watchdog

Digital-to-analog converter (Electronic circuit that converts a binary number into a continuously varying value.)

Device Tree Source (in software context) or Digital Temperature Sensor (in peripheral context)

Real Time Clock

Tamper

Consumer Electronics Control (HDMI standard)

Power Management Integrated Circuit

Display Serial Interface (MIPI® Alliance standard)

Analog-to-digital converter. The process of converting a sampled analog signal to a digital code that represents the amplitude of the original signal sample.

USB Host (STM32 specific)

Boot and Security and OTP control

Extended TrustZone Protection Controller



TrustZone® address space Controller for DDR

Arm® and TrustZone® are registered trademarks of Arm Limited (or its subsidiaries) in the US and/or elsewhere.

debug and test protocol, named from the Joint Test Action Group that developed it

System Trace Module

Graphics Processing Units

Doubledata rate (memory domain)

System Configuration

voltage reference buffer (STM32 specific)

Hardware Debug Port

Digital Camera Memory Interface

Inter-Processor Communication Controller

External Interrupt

Digital Filter for Sigma-Delta Modulator

Transmit

former spelling for e•MMC ('e' in italic)

Receive

LCD TFT Display Controller (STM32 specific)

System Time Generator

USB On-The-Go (Capability/type of USB port, acting primarily as USB device, to also act as USB host. Also known as USB OTG.)

input/output

Ethernet

Microprocessor Unit

Microcontroller Unit (MCUs have internal flash memory and are intended to operate with a minimum amount of external support ICs. They commonly are a self-contained, system-on-chip (SoC) designs.)

Discovery kit

BlueTooth

High-Definition Multimedia Interface (HDMI standard)

Stable: 01.04.2021 - 11:49 / Revision: 29.03.2021 - 08:47

All the peripherals receive one or several clocks that are generated via [RCC internal peripheral](#). RCC relies on several clocks sources (LSI, LSE, HSI, HSE, CSI) and four PLL in order to provide adequate input frequencies to all the peripherals. The clock tree covers all the system clock distribution aspects, from the clock sources to the consumer peripherals (internal and external), except clock gating management that is locally controlled by each peripheral driver.

Contents

1 Overview	121
2 How to build a clock tree?	122



3 ST boards clock tree	123
3.1 STM32MP157x-EV1 case	123
3.1.1 Clock tree	123
3.1.1.1 Device tree	125
3.2 STM32MP157x-DK2 case	127
3.2.1 Clock tree	127
3.2.2 Device tree	129



1 Overview

The clock tree is managed via RCC internal peripheral hardware block and it is configured at different steps from the Cortex-A7:

- When the device is **reset**, all RCC registers take their reset values: the four PLL are disabled and most of the clock source selectors are pointing to the HSI.
- The ROM Code configures the minimum clock tree needed to boot on the selected boot device.
- The **FSBL** (Cf. [Boot chain overview](#)) completely configures the clock tree as expected, all the way up to Linux, thanks to the configuration given in the [device tree](#).
 - U-Boot SPL (basic boot chain) binding is available in [doc/device-tree-bindings/clock/st,stm32mp1.txt](#)
 - TF-A (trusted boot chain) binding follows the same syntax as U-Boot SPL
- **Linux** is responsible for the clock tree at runtime and it may partly modify it thanks to information taken from the [device tree](#):
 - Linux binding is available in [Documentation/devicetree/bindings/clock/clock-bindings.txt](#) (and surrounding files): 'fixed-clock' compatible, 'clocks' and 'assigned-clocks' properties are important concepts to understand the management of clocks providers/consumers.

Notice that [STM32Cube](#) (running on the Cortex-M4) does not control the clock tree setup so the configuration applied by the Cortex-A7 must provide suitable clocks to Cortex-M4, like it is explained in [resource manager](#) description. One exception to this rule is the [engineering boot](#), allowing to directly load (and debug) the Cortex-M4 whereas the Cortex-A7 execution is stalled in the ROM code.



2 How to build a clock tree?

Building a clock tree is quite complex as it needs to take into account the constraints set by each internal and external peripheral, including external clock sources.

We encourage the use of [STM32CubeMX](#) to build the clock tree, and avoid having to know all internal peripherals details: the tool allows to select the peripherals that will be present on the board, fix the clock sources frequencies and automatically find an optimized clock tree. It is then able to generate the device tree that is directly consumed by the boot chain and Linux.



3 ST boards clock tree



See [How to change the CPU frequency](#) article for more information about the CPU frequency setting (PLL1P), including dynamic voltage and frequency scaling (DVFS) configuration.

3.1 STM32MP157x-EV1 case

This chapter shows the boot time clock tree set by the FSBL on STM32MP157x-EV1 evaluation board. Linux eventual runtime modifications are not covered here.

3.1.1 Clock tree

The following table shows what STM32MP157x-EV1 clock tree looks like, as a result of the boot chain execution with the device tree built with STM32CubeMX.

Component Comment	Parent	Frequency	Used?	
LSI	N.A.	0.032000 MHz	yes	Mandatory for IWDG, DAC
DAC	LSI	0.032000 MHz	yes	
RNG1	LSI	0.032000 MHz	yes	Input frequency should be
as low as possible				
RNG2	LSI	0.032000 MHz	yes	Input frequency should be
as low as possible				
IWDG1	LSI	0.032000 MHz	yes	
IWDG2	LSI	0.032000 MHz	yes	
LSE	N.A.	0.032768 MHz	yes	Mandatory for DTS
RTC	LSE	0.032768 MHz	yes	
TAMP	LSE	0.032768 MHz	yes	
CEC	LSE	0.032768 MHz	yes	
LPTIM4	LSE	0.032768 MHz	yes	
LPTIM5	LSE	0.032768 MHz	yes	
HSI	N.A.	64.000000 MHz	yes	
SPI4	HSI	64.000000 MHz	no	
SPI5	HSI	64.000000 MHz	no	
SPI6	HSI	64.000000 MHz	yes	
I2C4	HSI	64.000000 MHz	yes	PMIC
I2C6	HSI	64.000000 MHz	no	
I2C1	HSI	64.000000 MHz	no	
I2C2	HSI	64.000000 MHz	yes	Rpi and peripherals
I2C3	HSI	64.000000 MHz	no	
I2C5	HSI	64.000000 MHz	yes	Rpi
USART1	HSI	64.000000 MHz	yes	
USART2	HSI	64.000000 MHz	no	
USART3	HSI	64.000000 MHz	yes	Rpi
UART4	HSI	64.000000 MHz	yes	Linux console
UART5	HSI	64.000000 MHz	no	
UART6	HSI	64.000000 MHz	no	
UART7	HSI	64.000000 MHz	no	
UART8	HSI	64.000000 MHz	no	
MC01	HSI	64.000000 MHz	no	Available so can be used
HSE	N.A.	24.000000 MHz	yes	
DSIPLL	HSE	125.000000 MHz	no	DSI DPHY PLL
DSIBL	DSIPLL	125.000000 MHz	no	DSI lanebyte clock
RTCDIV	HSE	1.000000 MHz	yes	Only used when RTC source
is HSE				



ck_per	HSE	24.000000	MHz	yes	
ADC	ck_per	24.000000	MHz	yes	Can use internal divider to
be < 40MHz					
PLL1	HSE	xxx	MHz	yes	
PLL1P	PLL1	xxx	MHz	yes	
MPUDIV	PLL1P	xxx	MHz	yes	
Cortex-A7	PLL1P	xxx	MHz	yes	650 MHz or 800 MHz, see the
information box above					
MC02	Cortex-A7	xxx	MHz	no	Available so can be used
PLL2	HSE	1066.000000	MHz	yes	
PLL2P	PLL2	266.500000	MHz	yes	
AXI	PLL2P	266.500000	MHz	yes	< 266MHz
FMC	AXI	266.500000	MHz	yes	NAND flash
QSPI	AXI	266.500000	MHz	yes	NOR flash
SYSRAM	AXI	266.500000	MHz	yes	
ROM	AXI	266.500000	MHz	yes	
AHB5	AXI	266.500000	MHz	yes	< 266MHz
CRYP1	AHB5	266.500000	MHz	yes	
HASH1	AHB5	266.500000	MHz	yes	
GPIOZ	AHB5	266.500000	MHz	yes	
BKPSRAM	AHB5	266.500000	MHz	yes	
AHB6	AXI	266.500000	MHz	yes	< 266MHz
CRC1	AHB6	266.500000	MHz	yes	
MDMA	AHB6	266.500000	MHz	yes	
USBH	AHB6	266.500000	MHz	yes	USB Host
APB4	AHB6	133.250000	MHz	yes	
APB5	AHB6	66.625000	MHz	yes	
BSEC	APB5	66.625000	MHz	yes	< 67MHz
ETZPC	APB5	66.625000	MHz	yes	
TZC	APB5	66.625000	MHz	yes	
DBGAPB	AXI	133.250000	MHz	yes	JTAG & Coresight
DBGMCU	DBGAPB	66.625000	MHz	yes	
STM	DBGAPB	66.625000	MHz	yes	
PLL2Q	PLL2	533.000000	MHz	yes	
GPU	PLL2Q	533.000000	MHz	yes	< 533MHz
PLL2R	PLL2	533.000000	MHz	yes	
DDR	PLL2R	533.000000	MHz	yes	< 533MHz
PLL3	HSE	417.755859	MHz	yes	
PLL3P	PLL3	208.877930	MHz	yes	
MLAHB	PLL3P	208.877930	MHz	yes	< 209MHz
Cortex-M4	MLAHB	208.877930	MHz	yes	
SRAM1	MLAHB	208.877930	MHz	yes	
SRAM2	MLAHB	208.877930	MHz	yes	
SRAM3	MLAHB	208.877930	MHz	yes	
RETRAM	MLAHB	208.877930	MHz	yes	
AHB1	MLAHB	208.877930	MHz	yes	< 209MHz
AHB2	MLAHB	208.877930	MHz	yes	< 209MHz
DMA1	AHB2	208.877930	MHz	yes	
DMA2	AHB2	208.877930	MHz	yes	
DMAMUX	AHB2	208.877930	MHz	yes	
APB1	MLAHB	104.438965	MHz	yes	
LPTIM1	APB1	104.438965	MHz	yes	
WWDG	APB1	104.438965	MHz	yes	
APB2	MLAHB	104.438965	MHz	yes	
TIM2	MLAHB	208.877930	MHz	yes	TIM Group 1
TIM3	MLAHB	208.877930	MHz	yes	TIM Group 1
TIM4	MLAHB	208.877930	MHz	yes	TIM Group 1
TIM5	MLAHB	208.877930	MHz	yes	TIM Group 1
TIM6	MLAHB	208.877930	MHz	yes	TIM Group 1
TIM7	MLAHB	208.877930	MHz	yes	TIM Group 1
TIM12	MLAHB	208.877930	MHz	yes	TIM Group 1
TIM13	MLAHB	208.877930	MHz	yes	TIM Group 1
TIM14	MLAHB	208.877930	MHz	yes	TIM Group 1
TIM1	MLAHB	208.877930	MHz	yes	TIM Group 2
TIM8	MLAHB	208.877930	MHz	yes	TIM Group 2
TIM15	MLAHB	208.877930	MHz	yes	TIM Group 2
TIM16	MLAHB	208.877930	MHz	yes	TIM Group 2



TIM17	MLAHB	208.877930	MHz	yes	TIM Group 2
APB3	MLAHB	104.438965	MHz	yes	
LPTIM2	APB3	104.438965	MHz	yes	
LPTIM3	APB3	104.438965	MHz	yes	
SYSCFG	APB3	104.438965	MHz	yes	
VREFBUF	APB3	104.438965	MHz	yes	
DTS	APB3	104.438965	MHz	yes	
HDP	APB3	104.438965	MHz	yes	
AHB3	MLAHB	208.877930	MHz	yes	< 209MHz
CRC2	AHB3	208.877930	MHz	yes	
CRYP2	AHB3	208.877930	MHz	yes	
HASH2	AHB3	208.877930	MHz	yes	
DCMI	AHB3	208.877930	MHz	yes	Camera
IPCC	AHB3	208.877930	MHz	yes	Mailbox
AHB4	MLAHB	208.877930	MHz	yes	< 209MHz
PWR	AHB4	208.877930	MHz	yes	
RCC	AHB4	208.877930	MHz	yes	
GPIOA-K	AHB4	208.877930	MHz	yes	
EXTI	AHB4	208.877930	MHz	yes	
PLL3Q	PLL3	24.573874	MHz	yes	
SPI1	PLL3Q	24.573874	MHz	no	
SPI2	PLL3Q	24.573874	MHz	no	
SPI3	PLL3Q	24.573874	MHz	no	
DFSDM	PLL3Q	24.573874	MHz	yes	Digital micro
SAI1	PLL3Q	24.573874	MHz	no	
SAI2	PLL3Q	24.573874	MHz	yes	AudCodec 48kHz (use PLL3R
for 44.1kHz)					
SAI3	PLL3Q	24.573874	MHz	no	
SAI4	PLL3Q	24.573874	MHz	yes	SPDIF TX 48kHz (use PLL3R
for 44.1kHz)					
PLL3R	PLL3	11.290699	MHz	yes	
PLL4	HSE	594.000000	MHz	yes	
PLL4P	PLL4	99.000000	MHz	yes	
SDMMC1	PLL4P	99.000000	MHz	yes	µSD card
SDMMC2	PLL4P	99.000000	MHz	yes	eMMC
SDMMC3	PLL4P	99.000000	MHz	yes	
SPDIF	PLL4P	99.000000	MHz	yes	SPDIF RX
PLL4Q	PLL4	74.250000	MHz	yes	
LCD	PLL4Q	74.250000	MHz	yes	LTDC & DSI display pixel
clock					
PLL4R	PLL4	74.250000	MHz	yes	
FDCAN	PLL4R	74.250000	MHz	yes	Should be as high as
possible and < 100MHz					
STGEN	HSE	24.000000	MHz	yes	
USBPHYC	HSE	24.000000	MHz	yes	USB PHY Ctrl for USB Host
and OTG					
USBPLL	USBPHYC	48.000000	MHz	yes	
USB0	USBPLL	48.000000	MHz	yes	USB OTG
CSI	N.A.	4.000000	MHz	yes	Mandatory for IO
compensation					
ETH	N.A.	0.000000	MHz	no	ETH clocked by RGMII PHY

3.1.1.1 Device tree

Here are the corresponding device tree `rcm_clk` sub node properties consumed by the first stage boot loader (FSBL) to configure the clock tree above:

```
st,clksrc = <
    CLK_MPU_PLL1P
    CLK_AXI_PLL2P
    CLK_MCU_PLL3P
    CLK_PLL12_HSE
```



```

        CLK_PLL3_HSE
        CLK_PLL4_HSE
        CLK_RTC_LSE
        CLK_MC01_DISABLED
        CLK_MC02_DISABLED
    >;

    st,clkdiv = <
        1 /*MPU*/
        0 /*AXI*/
        0 /*MCU*/
        1 /*APB1*/
        1 /*APB2*/
        1 /*APB3*/
        1 /*APB4*/
        2 /*APB5*/
        23 /*RTC*/
        0 /*MC01*/
        0 /*MC02*/
    >;

    st,pkcs = <
        CLK_CKPER_HSE
        CLK_FMC_ACLK
        CLK_QSPI_ACLK
        CLK_ETH_DISABLED
        CLK_SDMMC12_PLL4P
        CLK_DSI_DSIPLL
        CLK_STGEN_HSE
        CLK_USBPHY_HSE
        CLK_SPI2S1_PLL3Q
        CLK_SPI2S23_PLL3Q
        CLK_SPI45_HSI
        CLK_SPI6_HSI
        CLK_I2C46_HSI
        CLK_SDMMC3_PLL4P
        CLK_USB0_USBPHY
        CLK_ADC_CKPER
        CLK_CEC_LSE
        CLK_I2C12_HSI
        CLK_I2C35_HSI
        CLK_UART1_HSI
        CLK_UART24_HSI
        CLK_UART35_HSI
        CLK_UART6_HSI
        CLK_UART78_HSI
        CLK_SPDIF_PLL4P
        CLK_FDCAN_PLL4R
        CLK_SAI1_PLL3Q
        CLK_SAI2_PLL3Q
        CLK_SAI3_PLL3Q
        CLK_SAI4_PLL3Q
        CLK_RNG1_LSI
        CLK_RNG2_LSI
        CLK_LPTIM1_PCLK1
        CLK_LPTIM23_PCLK3
        CLK_LPTIM45_LSE
    >;

    /* VCO = 1066.0 MHz => P = 266 (AXI), Q = 533 (GPU), R = 533 (DDR) */
    pll2: st,pll@1 {
        cfg = < 2 65 1 0 0 PQR(1,1,1) >;
        frac = < 0x1400 >;
        u-boot,dm-pre-reloc;
    };

    /* VCO = 417.8 MHz => P = 209, Q = 25, R = 11 */

```



```

pll3: st,pll@2 {
    cfg = < 1 33 1 16 36 PQR(1,1,1) >;
    frac = < 0x1a04 >;
    u-boot,dm-pre-reloc;
};

/* VCO = 594.0 MHz => P = 99, Q = 74, R = 74 */
pll4: st,pll@3 {
    cfg = < 3 98 5 7 7 PQR(1,1,1) >;
    u-boot,dm-pre-reloc;
};
    
```

3.2 STM32MP157x-DK2 case

This chapter shows the boot time clock tree set by the FSBL on STM32MP157x-DK2 DISCO board. Linux eventual runtime modifications are not covered here.

3.2.1 Clock tree

The following table shows what STM32MP157x-DK2 clock tree looks like, as a result of the boot chain execution with the device tree built with STM32CubeMX.

Component Comment	Parent	Frequency	Used?	
LSI	N.A.	0.032000 MHz	yes	Mandatory for IWDG, DAC
DAC	LSI	0.032000 MHz	no	
RNG1	LSI	0.032000 MHz	yes	Input frequency should be as low as possible
RNG2	LSI	0.032000 MHz	yes	Input frequency should be as low as possible
IWDG1	LSI	0.032000 MHz	yes	
IWDG2	LSI	0.032000 MHz	yes	
LSE	N.A.	0.032768 MHz	yes	Mandatory for DTS
RTC	LSE	0.032768 MHz	yes	
TAMP	LSE	0.032768 MHz	yes	
CEC	LSE	0.032768 MHz	yes	
LPTIM4	LSE	0.032768 MHz	yes	
LPTIM5	LSE	0.032768 MHz	yes	
HSI	N.A.	64.000000 MHz	yes	
SPI4	HSI	64.000000 MHz	yes	Arduino
SPI5	HSI	64.000000 MHz	yes	Rpi
SPI6	HSI	64.000000 MHz	no	
I2C4	HSI	64.000000 MHz	yes	PMIC
I2C6	HSI	64.000000 MHz	no	
I2C1	HSI	64.000000 MHz	yes	Rpi and peripherals
I2C2	HSI	64.000000 MHz	no	
I2C3	HSI	64.000000 MHz	no	
I2C5	HSI	64.000000 MHz	yes	Rpi and Arduino
USART1	HSI	64.000000 MHz	no	
USART2	HSI	64.000000 MHz	yes	Bluetooth
USART3	HSI	64.000000 MHz	yes	Rpi
UART4	HSI	64.000000 MHz	yes	Linux console
UART5	HSI	64.000000 MHz	no	
USART6	HSI	64.000000 MHz	no	
UART7	HSI	64.000000 MHz	yes	Arduino
UART8	HSI	64.000000 MHz	no	
MCO1	HSI	64.000000 MHz	no	Available so can be used
HSE	N.A.	24.000000 MHz	yes	
DSIPLL	HSE	125.000000 MHz	no	DSI DPHY PLL
DSIBL	DSIPLL	125.000000 MHz	no	DSI lanebyte clock



RTCDIV	HSE	1.000000 MHz	yes	Only used when RTC source
is HSE				
ck_per	HSE	24.000000 MHz	yes	
ADC	ck_per	24.000000 MHz	yes	Can use internal divider to
be < 40MHz				
PLL1	HSE	xxx MHz	yes	
PLL1P	PLL1	xxx MHz	yes	
MPUDIV	PLL1P	xxx MHz	yes	
Cortex-A7	PLL1P	xxx MHz	yes	650 MHz or 800 MHz, see the
information box above				
MC02	Cortex-A7	xxx MHz	no	Available so can be used
PLL2	HSE	1066.000000 MHz	yes	
PLL2P	PLL2	266.500000 MHz	yes	
AXI	PLL2P	266.500000 MHz	yes	< 266MHz
FMC	AXI	266.500000 MHz	no	
QSPI	AXI	266.500000 MHz	no	
SYSRAM	AXI	266.500000 MHz	yes	
ROM	AXI	266.500000 MHz	yes	
AHB5	AXI	266.500000 MHz	yes	< 266MHz
CRYP1	AHB5	266.500000 MHz	yes	
HASH1	AHB5	266.500000 MHz	yes	
GPIOZ	AHB5	266.500000 MHz	yes	
BKPSRAM	AHB5	266.500000 MHz	yes	
AHB6	AXI	266.500000 MHz	yes	< 266MHz
CRC1	AHB6	266.500000 MHz	yes	
MDMA	AHB6	266.500000 MHz	yes	
USBH	AHB6	266.500000 MHz	yes	USB Host
APB4	AHB6	133.250000 MHz	yes	
APB5	AHB6	66.625000 MHz	yes	
BSEC	APB5	66.625000 MHz	yes	< 67MHz
ETZPC	APB5	66.625000 MHz	yes	
TZC	APB5	66.625000 MHz	yes	
DBGAPB	AXI	133.250000 MHz	yes	JTAG & Coresight
DBGMCU	DBGAPB	66.625000 MHz	yes	
STM	DBGAPB	66.625000 MHz	no	
PLL2Q	PLL2	533.000000 MHz	yes	
GPU	PLL2Q	533.000000 MHz	yes	< 533MHz
PLL2R	PLL2	533.000000 MHz	yes	
DDR	PLL2R	533.000000 MHz	yes	< 533MHz
PLL3	HSE	417.755859 MHz	yes	
PLL3P	PLL3	208.877930 MHz	yes	
MLAHB	PLL3P	208.877930 MHz	yes	< 209MHz
Cortex-M4	MLAHB	208.877930 MHz	yes	
SRAM1	MLAHB	208.877930 MHz	yes	
SRAM2	MLAHB	208.877930 MHz	yes	
SRAM3	MLAHB	208.877930 MHz	yes	
RETRAM	MLAHB	208.877930 MHz	yes	
AHB1	MLAHB	208.877930 MHz	yes	< 209MHz
AHB2	MLAHB	208.877930 MHz	yes	< 209MHz
DMA1	AHB2	208.877930 MHz	yes	
DMA2	AHB2	208.877930 MHz	yes	
DMAMUX	AHB2	208.877930 MHz	yes	
APB1	MLAHB	104.438965 MHz	yes	
LPTIM1	APB1	104.438965 MHz	yes	
WWDG	APB1	104.438965 MHz	yes	
APB2	MLAHB	104.438965 MHz	yes	
TIM2	MLAHB	208.877930 MHz	yes	TIM Group 1
TIM3	MLAHB	208.877930 MHz	yes	TIM Group 1
TIM4	MLAHB	208.877930 MHz	yes	TIM Group 1
TIM5	MLAHB	208.877930 MHz	yes	TIM Group 1
TIM6	MLAHB	208.877930 MHz	yes	TIM Group 1
TIM7	MLAHB	208.877930 MHz	yes	TIM Group 1
TIM12	MLAHB	208.877930 MHz	yes	TIM Group 1
TIM13	MLAHB	208.877930 MHz	yes	TIM Group 1
TIM14	MLAHB	208.877930 MHz	yes	TIM Group 1
TIM1	MLAHB	208.877930 MHz	yes	TIM Group 2
TIM8	MLAHB	208.877930 MHz	yes	TIM Group 2



TIM15	MLAHB	208.877930	MHz	yes	TIM Group 2
TIM16	MLAHB	208.877930	MHz	yes	TIM Group 2
TIM17	MLAHB	208.877930	MHz	yes	TIM Group 2
APB3	MLAHB	104.438965	MHz	yes	
LPTIM2	APB3	104.438965	MHz	yes	
LPTIM3	APB3	104.438965	MHz	yes	
SYSCFG	APB3	104.438965	MHz	yes	
VREFBUF	APB3	104.438965	MHz	yes	
DTS	APB3	104.438965	MHz	yes	
HDP	APB3	104.438965	MHz	no	
AHB3	MLAHB	208.877930	MHz	yes	< 209MHz
CRC2	AHB3	208.877930	MHz	yes	
CRYP2	AHB3	208.877930	MHz	yes	
HASH2	AHB3	208.877930	MHz	yes	
DCMI	AHB3	208.877930	MHz	no	
IPCC	AHB3	208.877930	MHz	yes	Mailbox
AHB4	MLAHB	208.877930	MHz	yes	< 209MHz
PWR	AHB4	208.877930	MHz	yes	
RCC	AHB4	208.877930	MHz	yes	
GPIOA-K	AHB4	208.877930	MHz	yes	
EXTI	AHB4	208.877930	MHz	yes	
PLL3Q	PLL3	24.573874	MHz	yes	
SPI1	PLL3Q	24.573874	MHz	yes	SPI2S1 for BT PCM
SPI2	PLL3Q	24.573874	MHz	yes	SPI2S2 for HDMI
SPI3	PLL3Q	24.573874	MHz	no	
DFSDM	PLL3Q	24.573874	MHz	no	
SAI1	PLL3Q	24.573874	MHz	no	
SAI2	PLL3Q	24.573874	MHz	yes	AudCodec 48kHz (use PLL3R)
for 44.1kHz)					
SAI3	PLL3Q	24.573874	MHz	no	
SAI4	PLL3Q	24.573874	MHz	no	
PLL3R	PLL3	11.290699	MHz	yes	
PLL4	HSE	594.000000	MHz	yes	
PLL4P	PLL4	99.000000	MHz	yes	
SDMMC1	PLL4P	99.000000	MHz	yes	µSD card
SDMMC2	PLL4P	99.000000	MHz	yes	Wifi
SDMMC3	PLL4P	99.000000	MHz	yes	Rpi
SPDIF	PLL4P	99.000000	MHz	no	
PLL4Q	PLL4	74.250000	MHz	yes	
LCD	PLL4Q	74.250000	MHz	yes	LTDC & DSI display pixel
clock					
PLL4R	PLL4	74.250000	MHz	yes	
FDCAN	PLL4R	74.250000	MHz	no	
STGEN	HSE	24.000000	MHz	yes	
USBPHYC	HSE	24.000000	MHz	yes	USB PHY Ctrl for USB Host
and OTG					
USBPLL	USBPHYC	48.000000	MHz	yes	
USB0	USBPLL	48.000000	MHz	yes	USB OTG
CSI	N.A.	4.000000	MHz	yes	Mandatory for IO
compensation					
ETH	N.A.	0.000000	MHz	no	ETH clocked by RGMII PHY

3.2.2 Device tree

Here are the corresponding device tree `rcc_clk` sub node properties consumed by the first stage boot loader (FSBL) to configure the clock tree above:

```
st,clksrc = <
    CLK_MPU_PLL1P
    CLK_AXI_PLL2P
    CLK_MCU_PLL3P
    CLK_PLL12_HSE
```



```

        CLK_PLL3_HSE
        CLK_PLL4_HSE
        CLK_RTC_LSE
        CLK_MC01_DISABLED
        CLK_MC02_DISABLED
    >;

    st,clkdiv = <
        1 /*MPU*/
        0 /*AXI*/
        0 /*MCU*/
        1 /*APB1*/
        1 /*APB2*/
        1 /*APB3*/
        1 /*APB4*/
        2 /*APB5*/
        23 /*RTC*/
        0 /*MC01*/
        0 /*MC02*/
    >;

    st,pkcs = <
        CLK_CKPER_HSE
        CLK_FMC_ACLK
        CLK_QSPI_ACLK
        CLK_ETH_DISABLED
        CLK_SDMMC12_PLL4P
        CLK_DSI_DSIPLL
        CLK_STGEN_HSE
        CLK_USBPHY_HSE
        CLK_SPI2S1_PLL3Q
        CLK_SPI2S23_PLL3Q
        CLK_SPI45_HSI
        CLK_SPI6_HSI
        CLK_I2C46_HSI
        CLK_SDMMC3_PLL4P
        CLK_USB0_USBPHY
        CLK_ADC_CKPER
        CLK_CEC_LSE
        CLK_I2C12_HSI
        CLK_I2C35_HSI
        CLK_UART1_HSI
        CLK_UART24_HSI
        CLK_UART35_HSI
        CLK_UART6_HSI
        CLK_UART78_HSI
        CLK_SPDIF_PLL4P
        CLK_FDCAN_PLL4R
        CLK_SAI1_PLL3Q
        CLK_SAI2_PLL3Q
        CLK_SAI3_PLL3Q
        CLK_SAI4_PLL3Q
        CLK_RNG1_LSI
        CLK_RNG2_LSI
        CLK_LPTIM1_PCLK1
        CLK_LPTIM23_PCLK3
        CLK_LPTIM45_LSE
    >;

    /* VCO = 1066.0 MHz => P = 266 (AXI), Q = 533 (GPU), R = 533 (DDR) */
    pll2: st,pll@1 {
        cfg = < 2 65 1 0 0 PQR(1,1,1) >;
        frac = < 0x1400 >;
        u-boot,dm-pre-reloc;
    };

    /* VCO = 417.8 MHz => P = 209, Q = 25, R = 11 */

```



```

pll3: st,pll@2 {
    cfg = < 1 33 1 16 36 PQR(1,1,1) >;
    frac = < 0x1a04 >;
    u-boot,dm-pre-reloc;
};

/* VCO = 594.0 MHz => P = 99, Q = 74, R = 74 */
pll4: st,pll@3 {
    cfg = < 3 98 5 7 7 PQR(1,1,1) >;
    u-boot,dm-pre-reloc;
};

```

Reset and Clock Control

Low Speed Internal oscillator (STM32 clock source)

Low Speed External oscillator (STM32 clock source)

High Speed Internal oscillator (STM32 clock source) or High Speed Synchronous Serial Interface (MIPI® Alliance standard)

High Speed External oscillator (STM32 clock source)

Multi Speed Internal oscillator (STM32 clock source)

Cortex®

First Stage Boot Loader

Linux® is a registered trademark of Linus Torvalds.

Das U-Boot -- the Universal Boot Loader (see [U-Boot_overview](#))

Secondary Program Loader, *Also known as **U-Boot SPL***

Read Only Memory

Central processing unit

Dynamic voltage and frequency scaling

Independent Watchdog

Digital-to-analog converter (Electronic circuit that converts a binary number into a continuously varying value.)

Device Tree Source (in software context) or Digital Temperature Sensor (in peripheral context)

Real Time Clock

Tamper

Consumer Electronics Control (HDMI standard)

Power Management Integrated Circuit

Display Serial Interface (MIPI® Alliance standard)

Analog-to-digital converter. The process of converting a sampled analog signal to a digital code that represents the amplitude of the original signal sample.

USB Host (STM32 specific)

Boot and Security and OTP control

Extended TrustZone Protection Controller



TrustZone® address space Controller for DDR

Arm® and TrustZone® are registered trademarks of Arm Limited (or its subsidiaries) in the US and/or elsewhere.

debug and test protocol, named from the Joint Test Action Group that developed it

System Trace Module

Graphics Processing Units

Doubledata rate (memory domain)

System Configuration

voltage reference buffer (STM32 specific)

Hardware Debug Port

Digital Camera Memory Interface

Inter-Processor Communication Controller

External Interrupt

Digital Filter for Sigma-Delta Modulator

Transmit

former spelling for e•MMC ('e' in italic)

Receive

LCD TFT Display Controller (STM32 specific)

System Time Generator

USB On-The-Go (Capability/type of USB port, acting primarily as USB device, to also act as USB host. Also known as USB OTG.)

input/output

Ethernet

Microprocessor Unit

Microcontroller Unit (MCUs have internal flash memory and are intended to operate with a minimum amount of external support ICs. They commonly are a self-contained, system-on-chip (SoC) designs.)

Discovery kit

BlueTooth

High-Definition Multimedia Interface (HDMI standard)

Stable: 05.01.2021 - 15:36 / Revision: 13.11.2020 - 10:23

All the peripherals receive one or several clocks that are generated via [RCC internal peripheral](#). RCC relies on several clocks sources (LSI, LSE, HSI, HSE, CSI) and four PLL in order to provide adequate input frequencies to all the peripherals. The clock tree covers all the system clock distribution aspects, from the clock sources to the consumer peripherals (internal and external), except clock gating management that is locally controlled by each peripheral driver.

Contents

1 Overview	134
2 How to build a clock tree?	135



3 ST boards clock tree	136
3.1 STM32MP157x-EV1 case	136
3.1.1 Clock tree	136
3.1.1.1 Device tree	138
3.2 STM32MP157x-DK2 case	140
3.2.1 Clock tree	140
3.2.2 Device tree	142



1 Overview

The clock tree is managed via RCC internal peripheral hardware block and it is configured at different steps from the Cortex-A7:

- When the device is **reset**, all RCC registers take their reset values: the four PLL are disabled and most of the clock source selectors are pointing to the HSI.
- The ROM Code configures the minimum clock tree needed to boot on the selected boot device.
- The **FSBL** (Cf. [Boot chain overview](#)) completely configures the clock tree as expected, all the way up to Linux, thanks to the configuration given in the [device tree](#).
 - U-Boot SPL (basic boot chain) binding is available in [doc/device-tree-bindings/clock/st,stm32mp1.txt](#)
 - TF-A (trusted boot chain) binding follows the same syntax as U-Boot SPL
- **Linux** is responsible for the clock tree at runtime and it may partly modify it thanks to information taken from the [device tree](#):
 - Linux binding is available in [Documentation/devicetree/bindings/clock/clock-bindings.txt](#) (and surrounding files): 'fixed-clock' compatible, 'clocks' and 'assigned-clocks' properties are important concepts to understand the management of clocks providers/consumers.

Notice that [STM32Cube](#) (running on the Cortex-M4) does not control the clock tree setup so the configuration applied by the Cortex-A7 must provide suitable clocks to Cortex-M4, like it is explained in [resource manager](#) description. One exception to this rule is the [engineering boot](#), allowing to directly load (and debug) the Cortex-M4 whereas the Cortex-A7 execution is stalled in the ROM code.



2 How to build a clock tree?

Building a clock tree is quite complex as it needs to take into account the constraints set by each internal and external peripheral, including external clock sources.

We encourage the use of [STM32CubeMX](#) to build the clock tree, and avoid having to know all internal peripherals details: the tool allows to select the peripherals that will be present on the board, fix the clock sources frequencies and automatically find an optimized clock tree. It is then able to generate the device tree that is directly consumed by the boot chain and Linux.



3 ST boards clock tree

 See [How to change the CPU frequency](#) article for more information about the CPU frequency setting (PLL1P), including dynamic voltage and frequency scaling (DVFS) configuration.

3.1 STM32MP157x-EV1 case

This chapter shows the boot time clock tree set by the FSBL on STM32MP157x-EV1 evaluation board. Linux eventual runtime modifications are not covered here.

3.1.1 Clock tree

The following table shows what STM32MP157x-EV1 clock tree looks like, as a result of the boot chain execution with the device tree built with STM32CubeMX.

Component Comment	Parent	Frequency	Used?	
LSI	N.A.	0.032000 MHz	yes	Mandatory for IWDG, DAC
DAC	LSI	0.032000 MHz	yes	
RNG1	LSI	0.032000 MHz	yes	Input frequency should be
as low as possible				
RNG2	LSI	0.032000 MHz	yes	Input frequency should be
as low as possible				
IWDG1	LSI	0.032000 MHz	yes	
IWDG2	LSI	0.032000 MHz	yes	
LSE	N.A.	0.032768 MHz	yes	Mandatory for DTS
RTC	LSE	0.032768 MHz	yes	
TAMP	LSE	0.032768 MHz	yes	
CEC	LSE	0.032768 MHz	yes	
LPTIM4	LSE	0.032768 MHz	yes	
LPTIM5	LSE	0.032768 MHz	yes	
HSI	N.A.	64.000000 MHz	yes	
SPI4	HSI	64.000000 MHz	no	
SPI5	HSI	64.000000 MHz	no	
SPI6	HSI	64.000000 MHz	yes	
I2C4	HSI	64.000000 MHz	yes	PMIC
I2C6	HSI	64.000000 MHz	no	
I2C1	HSI	64.000000 MHz	no	
I2C2	HSI	64.000000 MHz	yes	Rpi and peripherals
I2C3	HSI	64.000000 MHz	no	
I2C5	HSI	64.000000 MHz	yes	Rpi
USART1	HSI	64.000000 MHz	yes	
USART2	HSI	64.000000 MHz	no	
USART3	HSI	64.000000 MHz	yes	Rpi
UART4	HSI	64.000000 MHz	yes	Linux console
UART5	HSI	64.000000 MHz	no	
USART6	HSI	64.000000 MHz	no	
UART7	HSI	64.000000 MHz	no	
UART8	HSI	64.000000 MHz	no	
MC01	HSI	64.000000 MHz	no	Available so can be used
HSE	N.A.	24.000000 MHz	yes	
DSIPLL	HSE	125.000000 MHz	no	DSI DPHY PLL
DSIBL	DSIPLL	125.000000 MHz	no	DSI lanebyte clock
RTCDIV	HSE	1.000000 MHz	yes	Only used when RTC source
is HSE				



ck_per	HSE	24.000000	MHz	yes	
ADC	ck_per	24.000000	MHz	yes	Can use internal divider to
be < 40MHz					
PLL1	HSE	xxx	MHz	yes	
PLL1P	PLL1	xxx	MHz	yes	
MPUDIV	PLL1P	xxx	MHz	yes	
Cortex-A7	PLL1P	xxx	MHz	yes	650 MHz or 800 MHz, see the
information box above					
MC02	Cortex-A7	xxx	MHz	no	Available so can be used
PLL2	HSE	1066.000000	MHz	yes	
PLL2P	PLL2	266.500000	MHz	yes	
AXI	PLL2P	266.500000	MHz	yes	< 266MHz
FMC	AXI	266.500000	MHz	yes	NAND flash
QSPI	AXI	266.500000	MHz	yes	NOR flash
SYSRAM	AXI	266.500000	MHz	yes	
ROM	AXI	266.500000	MHz	yes	
AHB5	AXI	266.500000	MHz	yes	< 266MHz
CRYP1	AHB5	266.500000	MHz	yes	
HASH1	AHB5	266.500000	MHz	yes	
GPI0Z	AHB5	266.500000	MHz	yes	
BKPSRAM	AHB5	266.500000	MHz	yes	
AHB6	AXI	266.500000	MHz	yes	< 266MHz
CRC1	AHB6	266.500000	MHz	yes	
MDMA	AHB6	266.500000	MHz	yes	
USBH	AHB6	266.500000	MHz	yes	USB Host
APB4	AHB6	133.250000	MHz	yes	
APB5	AHB6	66.625000	MHz	yes	
BSEC	APB5	66.625000	MHz	yes	< 67MHz
ETZPC	APB5	66.625000	MHz	yes	
TZC	APB5	66.625000	MHz	yes	
DBGAPB	AXI	133.250000	MHz	yes	JTAG & Coresight
DBGMCU	DBGAPB	66.625000	MHz	yes	
STM	DBGAPB	66.625000	MHz	yes	
PLL2Q	PLL2	533.000000	MHz	yes	
GPU	PLL2Q	533.000000	MHz	yes	< 533MHz
PLL2R	PLL2	533.000000	MHz	yes	
DDR	PLL2R	533.000000	MHz	yes	< 533MHz
PLL3	HSE	417.755859	MHz	yes	
PLL3P	PLL3	208.877930	MHz	yes	
MLAHB	PLL3P	208.877930	MHz	yes	< 209MHz
Cortex-M4	MLAHB	208.877930	MHz	yes	
SRAM1	MLAHB	208.877930	MHz	yes	
SRAM2	MLAHB	208.877930	MHz	yes	
SRAM3	MLAHB	208.877930	MHz	yes	
RETRAM	MLAHB	208.877930	MHz	yes	
AHB1	MLAHB	208.877930	MHz	yes	< 209MHz
AHB2	MLAHB	208.877930	MHz	yes	< 209MHz
DMA1	AHB2	208.877930	MHz	yes	
DMA2	AHB2	208.877930	MHz	yes	
DMAMUX	AHB2	208.877930	MHz	yes	
APB1	MLAHB	104.438965	MHz	yes	
LPTIM1	APB1	104.438965	MHz	yes	
WWDG	APB1	104.438965	MHz	yes	
APB2	MLAHB	104.438965	MHz	yes	
TIM2	MLAHB	208.877930	MHz	yes	TIM Group 1
TIM3	MLAHB	208.877930	MHz	yes	TIM Group 1
TIM4	MLAHB	208.877930	MHz	yes	TIM Group 1
TIM5	MLAHB	208.877930	MHz	yes	TIM Group 1
TIM6	MLAHB	208.877930	MHz	yes	TIM Group 1
TIM7	MLAHB	208.877930	MHz	yes	TIM Group 1
TIM12	MLAHB	208.877930	MHz	yes	TIM Group 1
TIM13	MLAHB	208.877930	MHz	yes	TIM Group 1
TIM14	MLAHB	208.877930	MHz	yes	TIM Group 1
TIM1	MLAHB	208.877930	MHz	yes	TIM Group 2
TIM8	MLAHB	208.877930	MHz	yes	TIM Group 2
TIM15	MLAHB	208.877930	MHz	yes	TIM Group 2
TIM16	MLAHB	208.877930	MHz	yes	TIM Group 2



TIM17	MLAHB	208.877930	MHz	yes	TIM Group 2
APB3	MLAHB	104.438965	MHz	yes	
LPTIM2	APB3	104.438965	MHz	yes	
LPTIM3	APB3	104.438965	MHz	yes	
SYSCFG	APB3	104.438965	MHz	yes	
VREFBUF	APB3	104.438965	MHz	yes	
DTS	APB3	104.438965	MHz	yes	
HDP	APB3	104.438965	MHz	yes	
AHB3	MLAHB	208.877930	MHz	yes	< 209MHz
CRC2	AHB3	208.877930	MHz	yes	
CRYP2	AHB3	208.877930	MHz	yes	
HASH2	AHB3	208.877930	MHz	yes	
DCMI	AHB3	208.877930	MHz	yes	Camera
IPCC	AHB3	208.877930	MHz	yes	Mailbox
AHB4	MLAHB	208.877930	MHz	yes	< 209MHz
PWR	AHB4	208.877930	MHz	yes	
RCC	AHB4	208.877930	MHz	yes	
GPIOA-K	AHB4	208.877930	MHz	yes	
EXTI	AHB4	208.877930	MHz	yes	
PLL3Q	PLL3	24.573874	MHz	yes	
SPI1	PLL3Q	24.573874	MHz	no	
SPI2	PLL3Q	24.573874	MHz	no	
SPI3	PLL3Q	24.573874	MHz	no	
DFSDM	PLL3Q	24.573874	MHz	yes	Digital micro
SAI1	PLL3Q	24.573874	MHz	no	
SAI2	PLL3Q	24.573874	MHz	yes	AudCodec 48kHz (use PLL3R)
for 44.1kHz)					
SAI3	PLL3Q	24.573874	MHz	no	
SAI4	PLL3Q	24.573874	MHz	yes	SPDIF TX 48kHz (use PLL3R)
for 44.1kHz)					
PLL3R	PLL3	11.290699	MHz	yes	
PLL4	HSE	594.000000	MHz	yes	
PLL4P	PLL4	99.000000	MHz	yes	
SDMMC1	PLL4P	99.000000	MHz	yes	µSD card
SDMMC2	PLL4P	99.000000	MHz	yes	eMMC
SDMMC3	PLL4P	99.000000	MHz	yes	
SPDIF	PLL4P	99.000000	MHz	yes	SPDIF RX
PLL4Q	PLL4	74.250000	MHz	yes	
LCD	PLL4Q	74.250000	MHz	yes	LTDC & DSI display pixel
clock					
PLL4R	PLL4	74.250000	MHz	yes	
FDCAN	PLL4R	74.250000	MHz	yes	Should be as high as
possible and < 100MHz					
STGEN	HSE	24.000000	MHz	yes	
USBPHYC	HSE	24.000000	MHz	yes	USB PHY Ctrl for USB Host
and OTG					
USBPLL	USBPHYC	48.000000	MHz	yes	
USB0	USBPLL	48.000000	MHz	yes	USB OTG
CSI	N.A.	4.000000	MHz	yes	Mandatory for IO
compensation					
ETH	N.A.	0.000000	MHz	no	ETH clocked by RGMII PHY

3.1.1.1 Device tree

Here are the corresponding device tree `rcm_clk` sub node properties consumed by the first stage boot loader (FSBL) to configure the clock tree above:

```
st,clksrc = <
    CLK_MPU_PLL1P
    CLK_AXI_PLL2P
    CLK_MCU_PLL3P
    CLK_PLL12_HSE
```



```

        CLK_PLL3_HSE
        CLK_PLL4_HSE
        CLK_RTC_LSE
        CLK_MC01_DISABLED
        CLK_MC02_DISABLED
    >;

    st,clkdiv = <
        1 /*MPU*/
        0 /*AXI*/
        0 /*MCU*/
        1 /*APB1*/
        1 /*APB2*/
        1 /*APB3*/
        1 /*APB4*/
        2 /*APB5*/
        23 /*RTC*/
        0 /*MC01*/
        0 /*MC02*/
    >;

    st,pkcs = <
        CLK_CKPER_HSE
        CLK_FMC_ACLK
        CLK_QSPI_ACLK
        CLK_ETH_DISABLED
        CLK_SDMMC12_PLL4P
        CLK_DSI_DSIPLL
        CLK_STGEN_HSE
        CLK_USBPHY_HSE
        CLK_SPI2S1_PLL3Q
        CLK_SPI2S23_PLL3Q
        CLK_SPI45_HSI
        CLK_SPI6_HSI
        CLK_I2C46_HSI
        CLK_SDMMC3_PLL4P
        CLK_USB0_USBPHY
        CLK_ADC_CKPER
        CLK_CEC_LSE
        CLK_I2C12_HSI
        CLK_I2C35_HSI
        CLK_UART1_HSI
        CLK_UART24_HSI
        CLK_UART35_HSI
        CLK_UART6_HSI
        CLK_UART78_HSI
        CLK_SPDIF_PLL4P
        CLK_FDCAN_PLL4R
        CLK_SAI1_PLL3Q
        CLK_SAI2_PLL3Q
        CLK_SAI3_PLL3Q
        CLK_SAI4_PLL3Q
        CLK_RNG1_LSI
        CLK_RNG2_LSI
        CLK_LPTIM1_PCLK1
        CLK_LPTIM23_PCLK3
        CLK_LPTIM45_LSE
    >;

    /* VCO = 1066.0 MHz => P = 266 (AXI), Q = 533 (GPU), R = 533 (DDR) */
    pll2: st,pll@1 {
        cfg = < 2 65 1 0 0 PQR(1,1,1) >;
        frac = < 0x1400 >;
        u-boot,dm-pre-reloc;
    };

    /* VCO = 417.8 MHz => P = 209, Q = 25, R = 11 */

```



```

pll3: st,pll@2 {
    cfg = < 1 33 1 16 36 PQR(1,1,1) >;
    frac = < 0x1a04 >;
    u-boot,dm-pre-reloc;
};

/* VCO = 594.0 MHz => P = 99, Q = 74, R = 74 */
pll4: st,pll@3 {
    cfg = < 3 98 5 7 7 PQR(1,1,1) >;
    u-boot,dm-pre-reloc;
};
    
```

3.2 STM32MP157x-DK2 case

This chapter shows the boot time clock tree set by the FSBL on STM32MP157x-DK2 DISCO board. Linux eventual runtime modifications are not covered here.

3.2.1 Clock tree

The following table shows what STM32MP157x-DK2 clock tree looks like, as a result of the boot chain execution with the device tree built with STM32CubeMX.

Component Comment	Parent	Frequency	Used?	
LSI	N.A.	0.032000 MHz	yes	Mandatory for IWDG, DAC
DAC	LSI	0.032000 MHz	no	
RNG1	LSI	0.032000 MHz	yes	Input frequency should be as low as possible
RNG2	LSI	0.032000 MHz	yes	Input frequency should be as low as possible
IWDG1	LSI	0.032000 MHz	yes	
IWDG2	LSI	0.032000 MHz	yes	
LSE	N.A.	0.032768 MHz	yes	Mandatory for DTS
RTC	LSE	0.032768 MHz	yes	
TAMP	LSE	0.032768 MHz	yes	
CEC	LSE	0.032768 MHz	yes	
LPTIM4	LSE	0.032768 MHz	yes	
LPTIM5	LSE	0.032768 MHz	yes	
HSI	N.A.	64.000000 MHz	yes	
SPI4	HSI	64.000000 MHz	yes	Arduino
SPI5	HSI	64.000000 MHz	yes	Rpi
SPI6	HSI	64.000000 MHz	no	
I2C4	HSI	64.000000 MHz	yes	PMIC
I2C6	HSI	64.000000 MHz	no	
I2C1	HSI	64.000000 MHz	yes	Rpi and peripherals
I2C2	HSI	64.000000 MHz	no	
I2C3	HSI	64.000000 MHz	no	
I2C5	HSI	64.000000 MHz	yes	Rpi and Arduino
USART1	HSI	64.000000 MHz	no	
USART2	HSI	64.000000 MHz	yes	Bluetooth
USART3	HSI	64.000000 MHz	yes	Rpi
UART4	HSI	64.000000 MHz	yes	Linux console
UART5	HSI	64.000000 MHz	no	
USART6	HSI	64.000000 MHz	no	
UART7	HSI	64.000000 MHz	yes	Arduino
UART8	HSI	64.000000 MHz	no	
MCO1	HSI	64.000000 MHz	no	Available so can be used
HSE	N.A.	24.000000 MHz	yes	
DSIPLL	HSE	125.000000 MHz	no	DSI DPHY PLL
DSIBL	DSIPLL	125.000000 MHz	no	DSI lanebyte clock



RTCDIV	HSE	1.000000 MHz	yes	Only used when RTC source
is HSE				
ck_per	HSE	24.000000 MHz	yes	
ADC	ck_per	24.000000 MHz	yes	Can use internal divider to
be < 40MHz				
PLL1	HSE	xxx MHz	yes	
PLL1P	PLL1	xxx MHz	yes	
MPUDIV	PLL1P	xxx MHz	yes	
Cortex-A7	PLL1P	xxx MHz	yes	650 MHz or 800 MHz, see the
information box above				
MC02	Cortex-A7	xxx MHz	no	Available so can be used
PLL2	HSE	1066.000000 MHz	yes	
PLL2P	PLL2	266.500000 MHz	yes	
AXI	PLL2P	266.500000 MHz	yes	< 266MHz
FMC	AXI	266.500000 MHz	no	
QSPI	AXI	266.500000 MHz	no	
SYSRAM	AXI	266.500000 MHz	yes	
ROM	AXI	266.500000 MHz	yes	
AHB5	AXI	266.500000 MHz	yes	< 266MHz
CRYP1	AHB5	266.500000 MHz	yes	
HASH1	AHB5	266.500000 MHz	yes	
GPI0Z	AHB5	266.500000 MHz	yes	
BKPSRAM	AHB5	266.500000 MHz	yes	
AHB6	AXI	266.500000 MHz	yes	< 266MHz
CRC1	AHB6	266.500000 MHz	yes	
MDMA	AHB6	266.500000 MHz	yes	
USBH	AHB6	266.500000 MHz	yes	USB Host
APB4	AHB6	133.250000 MHz	yes	
APB5	AHB6	66.625000 MHz	yes	
BSEC	APB5	66.625000 MHz	yes	< 67MHz
ETZPC	APB5	66.625000 MHz	yes	
TZC	APB5	66.625000 MHz	yes	
DBGAPB	AXI	133.250000 MHz	yes	JTAG & Coresight
DBGMCU	DBGAPB	66.625000 MHz	yes	
STM	DBGAPB	66.625000 MHz	no	
PLL2Q	PLL2	533.000000 MHz	yes	
GPU	PLL2Q	533.000000 MHz	yes	< 533MHz
PLL2R	PLL2	533.000000 MHz	yes	
DDR	PLL2R	533.000000 MHz	yes	< 533MHz
PLL3	HSE	417.755859 MHz	yes	
PLL3P	PLL3	208.877930 MHz	yes	
MLAHB	PLL3P	208.877930 MHz	yes	< 209MHz
Cortex-M4	MLAHB	208.877930 MHz	yes	
SRAM1	MLAHB	208.877930 MHz	yes	
SRAM2	MLAHB	208.877930 MHz	yes	
SRAM3	MLAHB	208.877930 MHz	yes	
RETRAM	MLAHB	208.877930 MHz	yes	
AHB1	MLAHB	208.877930 MHz	yes	< 209MHz
AHB2	MLAHB	208.877930 MHz	yes	< 209MHz
DMA1	AHB2	208.877930 MHz	yes	
DMA2	AHB2	208.877930 MHz	yes	
DMAMUX	AHB2	208.877930 MHz	yes	
APB1	MLAHB	104.438965 MHz	yes	
LPTIM1	APB1	104.438965 MHz	yes	
WWDG	APB1	104.438965 MHz	yes	
APB2	MLAHB	104.438965 MHz	yes	
TIM2	MLAHB	208.877930 MHz	yes	TIM Group 1
TIM3	MLAHB	208.877930 MHz	yes	TIM Group 1
TIM4	MLAHB	208.877930 MHz	yes	TIM Group 1
TIM5	MLAHB	208.877930 MHz	yes	TIM Group 1
TIM6	MLAHB	208.877930 MHz	yes	TIM Group 1
TIM7	MLAHB	208.877930 MHz	yes	TIM Group 1
TIM12	MLAHB	208.877930 MHz	yes	TIM Group 1
TIM13	MLAHB	208.877930 MHz	yes	TIM Group 1
TIM14	MLAHB	208.877930 MHz	yes	TIM Group 1
TIM1	MLAHB	208.877930 MHz	yes	TIM Group 2
TIM8	MLAHB	208.877930 MHz	yes	TIM Group 2



TIM15	MLAHB	208.877930	MHz	yes	TIM Group 2
TIM16	MLAHB	208.877930	MHz	yes	TIM Group 2
TIM17	MLAHB	208.877930	MHz	yes	TIM Group 2
APB3	MLAHB	104.438965	MHz	yes	
LPTIM2	APB3	104.438965	MHz	yes	
LPTIM3	APB3	104.438965	MHz	yes	
SYSCFG	APB3	104.438965	MHz	yes	
VREFBUF	APB3	104.438965	MHz	yes	
DTS	APB3	104.438965	MHz	yes	
HDP	APB3	104.438965	MHz	no	
AHB3	MLAHB	208.877930	MHz	yes	< 209MHz
CRC2	AHB3	208.877930	MHz	yes	
CRYP2	AHB3	208.877930	MHz	yes	
HASH2	AHB3	208.877930	MHz	yes	
DCMI	AHB3	208.877930	MHz	no	
IPCC	AHB3	208.877930	MHz	yes	Mailbox
AHB4	MLAHB	208.877930	MHz	yes	< 209MHz
PWR	AHB4	208.877930	MHz	yes	
RCC	AHB4	208.877930	MHz	yes	
GPIOA-K	AHB4	208.877930	MHz	yes	
EXTI	AHB4	208.877930	MHz	yes	
PLL3Q	PLL3	24.573874	MHz	yes	
SPI1	PLL3Q	24.573874	MHz	yes	SPI2S1 for BT PCM
SPI2	PLL3Q	24.573874	MHz	yes	SPI2S2 for HDMI
SPI3	PLL3Q	24.573874	MHz	no	
DFSDM	PLL3Q	24.573874	MHz	no	
SAI1	PLL3Q	24.573874	MHz	no	
SAI2	PLL3Q	24.573874	MHz	yes	AudCodec 48kHz (use PLL3R)
for 44.1kHz)					
SAI3	PLL3Q	24.573874	MHz	no	
SAI4	PLL3Q	24.573874	MHz	no	
PLL3R	PLL3	11.290699	MHz	yes	
PLL4	HSE	594.000000	MHz	yes	
PLL4P	PLL4	99.000000	MHz	yes	
SDMMC1	PLL4P	99.000000	MHz	yes	µSD card
SDMMC2	PLL4P	99.000000	MHz	yes	Wifi
SDMMC3	PLL4P	99.000000	MHz	yes	Rpi
SPDIF	PLL4P	99.000000	MHz	no	
PLL4Q	PLL4	74.250000	MHz	yes	
LCD	PLL4Q	74.250000	MHz	yes	LTDC & DSI display pixel
clock					
PLL4R	PLL4	74.250000	MHz	yes	
FDCAN	PLL4R	74.250000	MHz	no	
STGEN	HSE	24.000000	MHz	yes	
USBPHYC	HSE	24.000000	MHz	yes	USB PHY Ctrl for USB Host
and OTG					
USBPLL	USBPHYC	48.000000	MHz	yes	
USB0	USBPLL	48.000000	MHz	yes	USB OTG
CSI	N.A.	4.000000	MHz	yes	Mandatory for IO
compensation					
ETH	N.A.	0.000000	MHz	no	ETH clocked by RGMII PHY

3.2.2 Device tree

Here are the corresponding device tree `rcc_clk` sub node properties consumed by the first stage boot loader (FSBL) to configure the clock tree above:

```
st,clksrc = <
    CLK_MPU_PLL1P
    CLK_AXI_PLL2P
    CLK_MCU_PLL3P
    CLK_PLL12_HSE
```



```

        CLK_PLL3_HSE
        CLK_PLL4_HSE
        CLK_RTC_LSE
        CLK_MC01_DISABLED
        CLK_MC02_DISABLED
    >;

    st,clkdiv = <
        1 /*MPU*/
        0 /*AXI*/
        0 /*MCU*/
        1 /*APB1*/
        1 /*APB2*/
        1 /*APB3*/
        1 /*APB4*/
        2 /*APB5*/
        23 /*RTC*/
        0 /*MC01*/
        0 /*MC02*/
    >;

    st,pkcs = <
        CLK_CKPER_HSE
        CLK_FMC_ACLK
        CLK_QSPI_ACLK
        CLK_ETH_DISABLED
        CLK_SDMMC12_PLL4P
        CLK_DSI_DSIPLL
        CLK_STGEN_HSE
        CLK_USBPHY_HSE
        CLK_SPI2S1_PLL3Q
        CLK_SPI2S23_PLL3Q
        CLK_SPI45_HSI
        CLK_SPI6_HSI
        CLK_I2C46_HSI
        CLK_SDMMC3_PLL4P
        CLK_USB0_USBPHY
        CLK_ADC_CKPER
        CLK_CEC_LSE
        CLK_I2C12_HSI
        CLK_I2C35_HSI
        CLK_UART1_HSI
        CLK_UART24_HSI
        CLK_UART35_HSI
        CLK_UART6_HSI
        CLK_UART78_HSI
        CLK_SPDIF_PLL4P
        CLK_FDCAN_PLL4R
        CLK_SAI1_PLL3Q
        CLK_SAI2_PLL3Q
        CLK_SAI3_PLL3Q
        CLK_SAI4_PLL3Q
        CLK_RNG1_LSI
        CLK_RNG2_LSI
        CLK_LPTIM1_PCLK1
        CLK_LPTIM23_PCLK3
        CLK_LPTIM45_LSE
    >;

    /* VCO = 1066.0 MHz => P = 266 (AXI), Q = 533 (GPU), R = 533 (DDR) */
    pll2: st,pll@1 {
        cfg = < 2 65 1 0 0 PQR(1,1,1) >;
        frac = < 0x1400 >;
        u-boot,dm-pre-reloc;
    };

    /* VCO = 417.8 MHz => P = 209, Q = 25, R = 11 */

```



```

pll3: st,pll@2 {
    cfg = < 1 33 1 16 36 PQR(1,1,1) >;
    frac = < 0x1a04 >;
    u-boot,dm-pre-reloc;
};

/* VCO = 594.0 MHz => P = 99, Q = 74, R = 74 */
pll4: st,pll@3 {
    cfg = < 3 98 5 7 7 PQR(1,1,1) >;
    u-boot,dm-pre-reloc;
};

```

Reset and Clock Control

Low Speed Internal oscillator (STM32 clock source)

Low Speed External oscillator (STM32 clock source)

High Speed Internal oscillator (STM32 clock source) or High Speed Synchronous Serial Interface (MIPI® Alliance standard)

High Speed External oscillator (STM32 clock source)

Multi Speed Internal oscillator (STM32 clock source)

Cortex®

First Stage Boot Loader

Linux® is a registered trademark of Linus Torvalds.

Das U-Boot -- the Universal Boot Loader (see [U-Boot_overview](#))

Secondary Program Loader, *Also known as **U-Boot SPL***

Read Only Memory

Central processing unit

Dynamic voltage and frequency scaling

Independent Watchdog

Digital-to-analog converter (Electronic circuit that converts a binary number into a continuously varying value.)

Device Tree Source (in software context) or Digital Temperature Sensor (in peripheral context)

Real Time Clock

Tamper

Consumer Electronics Control (HDMI standard)

Power Management Integrated Circuit

Display Serial Interface (MIPI® Alliance standard)

Analog-to-digital converter. The process of converting a sampled analog signal to a digital code that represents the amplitude of the original signal sample.

USB Host (STM32 specific)

Boot and Security and OTP control

Extended TrustZone Protection Controller



TrustZone® address space Controller for DDR

Arm® and TrustZone® are registered trademarks of Arm Limited (or its subsidiaries) in the US and/or elsewhere.

debug and test protocol, named from the Joint Test Action Group that developed it

System Trace Module

Graphics Processing Units

Doubledata rate (memory domain)

System Configuration

voltage reference buffer (STM32 specific)

Hardware Debug Port

Digital Camera Memory Interface

Inter-Processor Communication Controller

External Interrupt

Digital Filter for Sigma-Delta Modulator

Transmit

former spelling for e•MMC ('e' in italic)

Receive

LCD TFT Display Controller (STM32 specific)

System Time Generator

USB On-The-Go (Capability/type of USB port, acting primarily as USB device, to also act as USB host. Also known as USB OTG.)

input/output

Ethernet

Microprocessor Unit

Microcontroller Unit (MCUs have internal flash memory and are intended to operate with a minimum amount of external support ICs. They commonly are a self-contained, system-on-chip (SoC) designs.)

Discovery kit

BlueTooth

High-Definition Multimedia Interface (HDMI standard)

Stable: 22.04.2021 - 11:23 / Revision: 09.04.2021 - 13:17

All the peripherals receive one or several clocks that are generated via [RCC internal peripheral](#). RCC relies on several clocks sources (LSI, LSE, HSI, HSE, CSI) and four PLL in order to provide adequate input frequencies to all the peripherals. The clock tree covers all the system clock distribution aspects, from the clock sources to the consumer peripherals (internal and external), except clock gating management that is locally controlled by each peripheral driver.

Contents

1 Overview	147
2 How to build a clock tree?	148



3 ST boards clock tree	149
3.1 STM32MP157x-EV1 case	149
3.1.1 Clock tree	149
3.1.1.1 Device tree	151
3.2 STM32MP157x-DK2 case	153
3.2.1 Clock tree	153
3.2.2 Device tree	155



1 Overview

The clock tree is managed via RCC internal peripheral hardware block and it is configured at different steps from the Cortex-A7:

- When the device is **reset**, all RCC registers take their reset values: the four PLL are disabled and most of the clock source selectors are pointing to the HSI.
- The ROM Code configures the minimum clock tree needed to boot on the selected boot device.
- The **FSBL** (Cf. [Boot chain overview](#)) completely configures the clock tree as expected, all the way up to Linux, thanks to the configuration given in the [device tree](#).
 - U-Boot SPL (basic boot chain) binding is available in [doc/device-tree-bindings/clock/st,stm32mp1.txt](#)
 - TF-A (trusted boot chain) binding follows the same syntax as U-Boot SPL
- **Linux** is responsible for the clock tree at runtime and it may partly modify it thanks to information taken from the [device tree](#):
 - Linux binding is available in [Documentation/devicetree/bindings/clock/clock-bindings.txt](#) (and surrounding files): 'fixed-clock' compatible, 'clocks' and 'assigned-clocks' properties are important concepts to understand the management of clocks providers/consumers.

Notice that [STM32Cube](#) (running on the Cortex-M4) does not control the clock tree setup so the configuration applied by the Cortex-A7 must provide suitable clocks to Cortex-M4, like it is explained in [resource manager](#) description. One exception to this rule is the [engineering boot](#), allowing to directly load (and debug) the Cortex-M4 whereas the Cortex-A7 execution is stalled in the ROM code.



2 How to build a clock tree?

Building a clock tree is quite complex as it needs to take into account the constraints set by each internal and external peripheral, including external clock sources.

We encourage the use of [STM32CubeMX](#) to build the clock tree, and avoid having to know all internal peripherals details: the tool allows to select the peripherals that will be present on the board, fix the clock sources frequencies and automatically find an optimized clock tree. It is then able to generate the device tree that is directly consumed by the boot chain and Linux.



3 ST boards clock tree

 See [How to change the CPU frequency](#) article for more information about the CPU frequency setting (PLL1P), including dynamic voltage and frequency scaling (DVFS) configuration.

3.1 STM32MP157x-EV1 case

This chapter shows the boot time clock tree set by the FSBL on STM32MP157x-EV1 evaluation board. Linux eventual runtime modifications are not covered here.

3.1.1 Clock tree

The following table shows what STM32MP157x-EV1 clock tree looks like, as a result of the boot chain execution with the device tree built with STM32CubeMX.

Component Comment	Parent	Frequency	Used?	
LSI	N.A.	0.032000 MHz	yes	Mandatory for IWDG, DAC
DAC	LSI	0.032000 MHz	yes	
RNG1	LSI	0.032000 MHz	yes	Input frequency should be
as low as possible				
RNG2	LSI	0.032000 MHz	yes	Input frequency should be
as low as possible				
IWDG1	LSI	0.032000 MHz	yes	
IWDG2	LSI	0.032000 MHz	yes	
LSE	N.A.	0.032768 MHz	yes	Mandatory for DTS
RTC	LSE	0.032768 MHz	yes	
TAMP	LSE	0.032768 MHz	yes	
CEC	LSE	0.032768 MHz	yes	
LPTIM4	LSE	0.032768 MHz	yes	
LPTIM5	LSE	0.032768 MHz	yes	
HSI	N.A.	64.000000 MHz	yes	
SPI4	HSI	64.000000 MHz	no	
SPI5	HSI	64.000000 MHz	no	
SPI6	HSI	64.000000 MHz	yes	
I2C4	HSI	64.000000 MHz	yes	PMIC
I2C6	HSI	64.000000 MHz	no	
I2C1	HSI	64.000000 MHz	no	
I2C2	HSI	64.000000 MHz	yes	Rpi and peripherals
I2C3	HSI	64.000000 MHz	no	
I2C5	HSI	64.000000 MHz	yes	Rpi
USART1	HSI	64.000000 MHz	yes	
USART2	HSI	64.000000 MHz	no	
USART3	HSI	64.000000 MHz	yes	Rpi
UART4	HSI	64.000000 MHz	yes	Linux console
UART5	HSI	64.000000 MHz	no	
USART6	HSI	64.000000 MHz	no	
UART7	HSI	64.000000 MHz	no	
UART8	HSI	64.000000 MHz	no	
MC01	HSI	64.000000 MHz	no	Available so can be used
HSE	N.A.	24.000000 MHz	yes	
DSIPLL	HSE	125.000000 MHz	no	DSI DPHY PLL
DSIBL	DSIPLL	125.000000 MHz	no	DSI lanebyte clock
RTCDIV	HSE	1.000000 MHz	yes	Only used when RTC source
is HSE				



ck_per	HSE	24.000000	MHz	yes	
ADC	ck_per	24.000000	MHz	yes	Can use internal divider to
be < 40MHz					
PLL1	HSE	xxx	MHz	yes	
PLL1P	PLL1	xxx	MHz	yes	
MPUDIV	PLL1P	xxx	MHz	yes	
Cortex-A7	PLL1P	xxx	MHz	yes	650 MHz or 800 MHz, see the
information box above					
MC02	Cortex-A7	xxx	MHz	no	Available so can be used
PLL2	HSE	1066.000000	MHz	yes	
PLL2P	PLL2	266.500000	MHz	yes	
AXI	PLL2P	266.500000	MHz	yes	< 266MHz
FMC	AXI	266.500000	MHz	yes	NAND flash
QSPI	AXI	266.500000	MHz	yes	NOR flash
SYSRAM	AXI	266.500000	MHz	yes	
ROM	AXI	266.500000	MHz	yes	
AHB5	AXI	266.500000	MHz	yes	< 266MHz
CRYP1	AHB5	266.500000	MHz	yes	
HASH1	AHB5	266.500000	MHz	yes	
GPI0Z	AHB5	266.500000	MHz	yes	
BKPSRAM	AHB5	266.500000	MHz	yes	
AHB6	AXI	266.500000	MHz	yes	< 266MHz
CRC1	AHB6	266.500000	MHz	yes	
MDMA	AHB6	266.500000	MHz	yes	
USBH	AHB6	266.500000	MHz	yes	USB Host
APB4	AHB6	133.250000	MHz	yes	
APB5	AHB6	66.625000	MHz	yes	
BSEC	APB5	66.625000	MHz	yes	< 67MHz
ETZPC	APB5	66.625000	MHz	yes	
TZC	APB5	66.625000	MHz	yes	
DBGAPB	AXI	133.250000	MHz	yes	JTAG & Coresight
DBGMCU	DBGAPB	66.625000	MHz	yes	
STM	DBGAPB	66.625000	MHz	yes	
PLL2Q	PLL2	533.000000	MHz	yes	
GPU	PLL2Q	533.000000	MHz	yes	< 533MHz
PLL2R	PLL2	533.000000	MHz	yes	
DDR	PLL2R	533.000000	MHz	yes	< 533MHz
PLL3	HSE	417.755859	MHz	yes	
PLL3P	PLL3	208.877930	MHz	yes	
MLAHB	PLL3P	208.877930	MHz	yes	< 209MHz
Cortex-M4	MLAHB	208.877930	MHz	yes	
SRAM1	MLAHB	208.877930	MHz	yes	
SRAM2	MLAHB	208.877930	MHz	yes	
SRAM3	MLAHB	208.877930	MHz	yes	
RETRAM	MLAHB	208.877930	MHz	yes	
AHB1	MLAHB	208.877930	MHz	yes	< 209MHz
AHB2	MLAHB	208.877930	MHz	yes	< 209MHz
DMA1	AHB2	208.877930	MHz	yes	
DMA2	AHB2	208.877930	MHz	yes	
DMAMUX	AHB2	208.877930	MHz	yes	
APB1	MLAHB	104.438965	MHz	yes	
LPTIM1	APB1	104.438965	MHz	yes	
WWDG	APB1	104.438965	MHz	yes	
APB2	MLAHB	104.438965	MHz	yes	
TIM2	MLAHB	208.877930	MHz	yes	TIM Group 1
TIM3	MLAHB	208.877930	MHz	yes	TIM Group 1
TIM4	MLAHB	208.877930	MHz	yes	TIM Group 1
TIM5	MLAHB	208.877930	MHz	yes	TIM Group 1
TIM6	MLAHB	208.877930	MHz	yes	TIM Group 1
TIM7	MLAHB	208.877930	MHz	yes	TIM Group 1
TIM12	MLAHB	208.877930	MHz	yes	TIM Group 1
TIM13	MLAHB	208.877930	MHz	yes	TIM Group 1
TIM14	MLAHB	208.877930	MHz	yes	TIM Group 1
TIM1	MLAHB	208.877930	MHz	yes	TIM Group 2
TIM8	MLAHB	208.877930	MHz	yes	TIM Group 2
TIM15	MLAHB	208.877930	MHz	yes	TIM Group 2
TIM16	MLAHB	208.877930	MHz	yes	TIM Group 2



TIM17	MLAHB	208.877930	MHz	yes	TIM Group 2
APB3	MLAHB	104.438965	MHz	yes	
LPTIM2	APB3	104.438965	MHz	yes	
LPTIM3	APB3	104.438965	MHz	yes	
SYSCFG	APB3	104.438965	MHz	yes	
VREFBUF	APB3	104.438965	MHz	yes	
DTS	APB3	104.438965	MHz	yes	
HDP	APB3	104.438965	MHz	yes	
AHB3	MLAHB	208.877930	MHz	yes	< 209MHz
CRC2	AHB3	208.877930	MHz	yes	
CRYP2	AHB3	208.877930	MHz	yes	
HASH2	AHB3	208.877930	MHz	yes	
DCMI	AHB3	208.877930	MHz	yes	Camera
IPCC	AHB3	208.877930	MHz	yes	Mailbox
AHB4	MLAHB	208.877930	MHz	yes	< 209MHz
PWR	AHB4	208.877930	MHz	yes	
RCC	AHB4	208.877930	MHz	yes	
GPIOA-K	AHB4	208.877930	MHz	yes	
EXTI	AHB4	208.877930	MHz	yes	
PLL3Q	PLL3	24.573874	MHz	yes	
SPI1	PLL3Q	24.573874	MHz	no	
SPI2	PLL3Q	24.573874	MHz	no	
SPI3	PLL3Q	24.573874	MHz	no	
DFSDM	PLL3Q	24.573874	MHz	yes	Digital micro
SAI1	PLL3Q	24.573874	MHz	no	
SAI2	PLL3Q	24.573874	MHz	yes	AudCodec 48kHz (use PLL3R
for 44.1kHz)					
SAI3	PLL3Q	24.573874	MHz	no	
SAI4	PLL3Q	24.573874	MHz	yes	SPDIF TX 48kHz (use PLL3R
for 44.1kHz)					
PLL3R	PLL3	11.290699	MHz	yes	
PLL4	HSE	594.000000	MHz	yes	
PLL4P	PLL4	99.000000	MHz	yes	
SDMMC1	PLL4P	99.000000	MHz	yes	µSD card
SDMMC2	PLL4P	99.000000	MHz	yes	eMMC
SDMMC3	PLL4P	99.000000	MHz	yes	
SPDIF	PLL4P	99.000000	MHz	yes	SPDIF RX
PLL4Q	PLL4	74.250000	MHz	yes	
LCD	PLL4Q	74.250000	MHz	yes	LTDC & DSI display pixel
clock					
PLL4R	PLL4	74.250000	MHz	yes	
FDCAN	PLL4R	74.250000	MHz	yes	Should be as high as
possible and < 100MHz					
STGEN	HSE	24.000000	MHz	yes	
USBPHYC	HSE	24.000000	MHz	yes	USB PHY Ctrl for USB Host
and OTG					
USBPLL	USBPHYC	48.000000	MHz	yes	
USB0	USBPLL	48.000000	MHz	yes	USB OTG
CSI	N.A.	4.000000	MHz	yes	Mandatory for IO
compensation					
ETH	N.A.	0.000000	MHz	no	ETH clocked by RGMII PHY

3.1.1.1 Device tree

Here are the corresponding device tree `rcc_clk` sub node properties consumed by the first stage boot loader (FSBL) to configure the clock tree above:

```
st,clksrc = <
    CLK_MPU_PLL1P
    CLK_AXI_PLL2P
    CLK_MCU_PLL3P
    CLK_PLL12_HSE
```



```

        CLK_PLL3_HSE
        CLK_PLL4_HSE
        CLK_RTC_LSE
        CLK_MC01_DISABLED
        CLK_MC02_DISABLED
    >;

    st,clkdiv = <
        1 /*MPU*/
        0 /*AXI*/
        0 /*MCU*/
        1 /*APB1*/
        1 /*APB2*/
        1 /*APB3*/
        1 /*APB4*/
        2 /*APB5*/
        23 /*RTC*/
        0 /*MC01*/
        0 /*MC02*/
    >;

    st,pkcs = <
        CLK_CKPER_HSE
        CLK_FMC_ACLK
        CLK_QSPI_ACLK
        CLK_ETH_DISABLED
        CLK_SDMMC12_PLL4P
        CLK_DSI_DSIPLL
        CLK_STGEN_HSE
        CLK_USBPHY_HSE
        CLK_SPI2S1_PLL3Q
        CLK_SPI2S23_PLL3Q
        CLK_SPI45_HSI
        CLK_SPI6_HSI
        CLK_I2C46_HSI
        CLK_SDMMC3_PLL4P
        CLK_USB0_USBPHY
        CLK_ADC_CKPER
        CLK_CEC_LSE
        CLK_I2C12_HSI
        CLK_I2C35_HSI
        CLK_UART1_HSI
        CLK_UART24_HSI
        CLK_UART35_HSI
        CLK_UART6_HSI
        CLK_UART78_HSI
        CLK_SPDIF_PLL4P
        CLK_FDCAN_PLL4R
        CLK_SAI1_PLL3Q
        CLK_SAI2_PLL3Q
        CLK_SAI3_PLL3Q
        CLK_SAI4_PLL3Q
        CLK_RNG1_LSI
        CLK_RNG2_LSI
        CLK_LPTIM1_PCLK1
        CLK_LPTIM23_PCLK3
        CLK_LPTIM45_LSE
    >;

    /* VCO = 1066.0 MHz => P = 266 (AXI), Q = 533 (GPU), R = 533 (DDR) */
    pll2: st,pll@1 {
        cfg = < 2 65 1 0 0 PQR(1,1,1) >;
        frac = < 0x1400 >;
        u-boot,dm-pre-reloc;
    };

    /* VCO = 417.8 MHz => P = 209, Q = 25, R = 11 */

```




```

pll3: st,pll@2 {
    cfg = < 1 33 1 16 36 PQR(1,1,1) >;
    frac = < 0x1a04 >;
    u-boot,dm-pre-reloc;
};

/* VCO = 594.0 MHz => P = 99, Q = 74, R = 74 */
pll4: st,pll@3 {
    cfg = < 3 98 5 7 7 PQR(1,1,1) >;
    u-boot,dm-pre-reloc;
};
    
```

3.2 STM32MP157x-DK2 case

This chapter shows the boot time clock tree set by the FSBL on STM32MP157x-DK2 DISCO board. Linux eventual runtime modifications are not covered here.

3.2.1 Clock tree

The following table shows what STM32MP157x-DK2 clock tree looks like, as a result of the boot chain execution with the device tree built with STM32CubeMX.

Component Comment	Parent	Frequency	Used?	
LSI	N.A.	0.032000 MHz	yes	Mandatory for IWDG, DAC
DAC	LSI	0.032000 MHz	no	
RNG1	LSI	0.032000 MHz	yes	Input frequency should be as low as possible
RNG2	LSI	0.032000 MHz	yes	Input frequency should be as low as possible
IWDG1	LSI	0.032000 MHz	yes	
IWDG2	LSI	0.032000 MHz	yes	
LSE	N.A.	0.032768 MHz	yes	Mandatory for DTS
RTC	LSE	0.032768 MHz	yes	
TAMP	LSE	0.032768 MHz	yes	
CEC	LSE	0.032768 MHz	yes	
LPTIM4	LSE	0.032768 MHz	yes	
LPTIM5	LSE	0.032768 MHz	yes	
HSI	N.A.	64.000000 MHz	yes	
SPI4	HSI	64.000000 MHz	yes	Arduino
SPI5	HSI	64.000000 MHz	yes	Rpi
SPI6	HSI	64.000000 MHz	no	
I2C4	HSI	64.000000 MHz	yes	PMIC
I2C6	HSI	64.000000 MHz	no	
I2C1	HSI	64.000000 MHz	yes	Rpi and peripherals
I2C2	HSI	64.000000 MHz	no	
I2C3	HSI	64.000000 MHz	no	
I2C5	HSI	64.000000 MHz	yes	Rpi and Arduino
USART1	HSI	64.000000 MHz	no	
USART2	HSI	64.000000 MHz	yes	Bluetooth
USART3	HSI	64.000000 MHz	yes	Rpi
UART4	HSI	64.000000 MHz	yes	Linux console
UART5	HSI	64.000000 MHz	no	
USART6	HSI	64.000000 MHz	no	
UART7	HSI	64.000000 MHz	yes	Arduino
UART8	HSI	64.000000 MHz	no	
MCO1	HSI	64.000000 MHz	no	Available so can be used
HSE	N.A.	24.000000 MHz	yes	
DSIPLL	HSE	125.000000 MHz	no	DSI DPHY PLL
DSIBL	DSIPLL	125.000000 MHz	no	DSI lanebyte clock



RTCDIV	HSE	1.000000 MHz	yes	Only used when RTC source
is HSE				
ck_per	HSE	24.000000 MHz	yes	
ADC	ck_per	24.000000 MHz	yes	Can use internal divider to
be < 40MHz				
PLL1	HSE	xxx MHz	yes	
PLL1P	PLL1	xxx MHz	yes	
MPUDIV	PLL1P	xxx MHz	yes	
Cortex-A7	PLL1P	xxx MHz	yes	650 MHz or 800 MHz, see the
information box above				
MC02	Cortex-A7	xxx MHz	no	Available so can be used
PLL2	HSE	1066.000000 MHz	yes	
PLL2P	PLL2	266.500000 MHz	yes	
AXI	PLL2P	266.500000 MHz	yes	< 266MHz
FMC	AXI	266.500000 MHz	no	
QSPI	AXI	266.500000 MHz	no	
SYSRAM	AXI	266.500000 MHz	yes	
ROM	AXI	266.500000 MHz	yes	
AHB5	AXI	266.500000 MHz	yes	< 266MHz
CRYPT1	AHB5	266.500000 MHz	yes	
HASH1	AHB5	266.500000 MHz	yes	
GPIOZ	AHB5	266.500000 MHz	yes	
BKPSRAM	AHB5	266.500000 MHz	yes	
AHB6	AXI	266.500000 MHz	yes	< 266MHz
CRC1	AHB6	266.500000 MHz	yes	
MDMA	AHB6	266.500000 MHz	yes	
USBH	AHB6	266.500000 MHz	yes	USB Host
APB4	AHB6	133.250000 MHz	yes	
APB5	AHB6	66.625000 MHz	yes	
BSEC	APB5	66.625000 MHz	yes	< 67MHz
ETZPC	APB5	66.625000 MHz	yes	
TZC	APB5	66.625000 MHz	yes	
DBGAPB	AXI	133.250000 MHz	yes	JTAG & Coresight
DBGMCU	DBGAPB	66.625000 MHz	yes	
STM	DBGAPB	66.625000 MHz	no	
PLL2Q	PLL2	533.000000 MHz	yes	
GPU	PLL2Q	533.000000 MHz	yes	< 533MHz
PLL2R	PLL2	533.000000 MHz	yes	
DDR	PLL2R	533.000000 MHz	yes	< 533MHz
PLL3	HSE	417.755859 MHz	yes	
PLL3P	PLL3	208.877930 MHz	yes	
MLAHB	PLL3P	208.877930 MHz	yes	< 209MHz
Cortex-M4	MLAHB	208.877930 MHz	yes	
SRAM1	MLAHB	208.877930 MHz	yes	
SRAM2	MLAHB	208.877930 MHz	yes	
SRAM3	MLAHB	208.877930 MHz	yes	
RETRAM	MLAHB	208.877930 MHz	yes	
AHB1	MLAHB	208.877930 MHz	yes	< 209MHz
AHB2	MLAHB	208.877930 MHz	yes	< 209MHz
DMA1	AHB2	208.877930 MHz	yes	
DMA2	AHB2	208.877930 MHz	yes	
DMAMUX	AHB2	208.877930 MHz	yes	
APB1	MLAHB	104.438965 MHz	yes	
LPTIM1	APB1	104.438965 MHz	yes	
WWDG	APB1	104.438965 MHz	yes	
APB2	MLAHB	104.438965 MHz	yes	
TIM2	MLAHB	208.877930 MHz	yes	TIM Group 1
TIM3	MLAHB	208.877930 MHz	yes	TIM Group 1
TIM4	MLAHB	208.877930 MHz	yes	TIM Group 1
TIM5	MLAHB	208.877930 MHz	yes	TIM Group 1
TIM6	MLAHB	208.877930 MHz	yes	TIM Group 1
TIM7	MLAHB	208.877930 MHz	yes	TIM Group 1
TIM12	MLAHB	208.877930 MHz	yes	TIM Group 1
TIM13	MLAHB	208.877930 MHz	yes	TIM Group 1
TIM14	MLAHB	208.877930 MHz	yes	TIM Group 1
TIM1	MLAHB	208.877930 MHz	yes	TIM Group 2
TIM8	MLAHB	208.877930 MHz	yes	TIM Group 2



TIM15	MLAHB	208.877930	MHz	yes	TIM Group 2
TIM16	MLAHB	208.877930	MHz	yes	TIM Group 2
TIM17	MLAHB	208.877930	MHz	yes	TIM Group 2
APB3	MLAHB	104.438965	MHz	yes	
LPTIM2	APB3	104.438965	MHz	yes	
LPTIM3	APB3	104.438965	MHz	yes	
SYSCFG	APB3	104.438965	MHz	yes	
VREFBUF	APB3	104.438965	MHz	yes	
DTS	APB3	104.438965	MHz	yes	
HDP	APB3	104.438965	MHz	no	
AHB3	MLAHB	208.877930	MHz	yes	< 209MHz
CRC2	AHB3	208.877930	MHz	yes	
CRYP2	AHB3	208.877930	MHz	yes	
HASH2	AHB3	208.877930	MHz	yes	
DCMI	AHB3	208.877930	MHz	no	
IPCC	AHB3	208.877930	MHz	yes	Mailbox
AHB4	MLAHB	208.877930	MHz	yes	< 209MHz
PWR	AHB4	208.877930	MHz	yes	
RCC	AHB4	208.877930	MHz	yes	
GPIOA-K	AHB4	208.877930	MHz	yes	
EXTI	AHB4	208.877930	MHz	yes	
PLL3Q	PLL3	24.573874	MHz	yes	
SPI1	PLL3Q	24.573874	MHz	yes	SPI2S1 for BT PCM
SPI2	PLL3Q	24.573874	MHz	yes	SPI2S2 for HDMI
SPI3	PLL3Q	24.573874	MHz	no	
DFSDM	PLL3Q	24.573874	MHz	no	
SAI1	PLL3Q	24.573874	MHz	no	
SAI2	PLL3Q	24.573874	MHz	yes	AudCodec 48kHz (use PLL3R)
for 44.1kHz)					
SAI3	PLL3Q	24.573874	MHz	no	
SAI4	PLL3Q	24.573874	MHz	no	
PLL3R	PLL3	11.290699	MHz	yes	
PLL4	HSE	594.000000	MHz	yes	
PLL4P	PLL4	99.000000	MHz	yes	
SDMMC1	PLL4P	99.000000	MHz	yes	µSD card
SDMMC2	PLL4P	99.000000	MHz	yes	Wifi
SDMMC3	PLL4P	99.000000	MHz	yes	Rpi
SPDIF	PLL4P	99.000000	MHz	no	
PLL4Q	PLL4	74.250000	MHz	yes	
LCD	PLL4Q	74.250000	MHz	yes	LTDC & DSI display pixel
clock					
PLL4R	PLL4	74.250000	MHz	yes	
FDCAN	PLL4R	74.250000	MHz	no	
STGEN	HSE	24.000000	MHz	yes	
USBPHYC	HSE	24.000000	MHz	yes	USB PHY Ctrl for USB Host
and OTG					
USBPLL	USBPHYC	48.000000	MHz	yes	
USB0	USBPLL	48.000000	MHz	yes	USB OTG
CSI	N.A.	4.000000	MHz	yes	Mandatory for IO
compensation					
ETH	N.A.	0.000000	MHz	no	ETH clocked by RGMII PHY

3.2.2 Device tree

Here are the corresponding device tree `rcc_clk` sub node properties consumed by the first stage boot loader (FSBL) to configure the clock tree above:

```
st,clksrc = <
    CLK_MPU_PLL1P
    CLK_AXI_PLL2P
    CLK_MCU_PLL3P
    CLK_PLL12_HSE
```



```

        CLK_PLL3_HSE
        CLK_PLL4_HSE
        CLK_RTC_LSE
        CLK_MC01_DISABLED
        CLK_MC02_DISABLED
    >;

    st,clkdiv = <
        1 /*MPU*/
        0 /*AXI*/
        0 /*MCU*/
        1 /*APB1*/
        1 /*APB2*/
        1 /*APB3*/
        1 /*APB4*/
        2 /*APB5*/
        23 /*RTC*/
        0 /*MC01*/
        0 /*MC02*/
    >;

    st,pkcs = <
        CLK_CKPER_HSE
        CLK_FMC_ACLK
        CLK_QSPI_ACLK
        CLK_ETH_DISABLED
        CLK_SDMMC12_PLL4P
        CLK_DSI_DSIPLL
        CLK_STGEN_HSE
        CLK_USBPHY_HSE
        CLK_SPI2S1_PLL3Q
        CLK_SPI2S23_PLL3Q
        CLK_SPI45_HSI
        CLK_SPI6_HSI
        CLK_I2C46_HSI
        CLK_SDMMC3_PLL4P
        CLK_USB0_USBPHY
        CLK_ADC_CKPER
        CLK_CEC_LSE
        CLK_I2C12_HSI
        CLK_I2C35_HSI
        CLK_UART1_HSI
        CLK_UART24_HSI
        CLK_UART35_HSI
        CLK_UART6_HSI
        CLK_UART78_HSI
        CLK_SPDIF_PLL4P
        CLK_FDCAN_PLL4R
        CLK_SAI1_PLL3Q
        CLK_SAI2_PLL3Q
        CLK_SAI3_PLL3Q
        CLK_SAI4_PLL3Q
        CLK_RNG1_LSI
        CLK_RNG2_LSI
        CLK_LPTIM1_PCLK1
        CLK_LPTIM23_PCLK3
        CLK_LPTIM45_LSE
    >;

    /* VCO = 1066.0 MHz => P = 266 (AXI), Q = 533 (GPU), R = 533 (DDR) */
    pll2: st,pll@1 {
        cfg = < 2 65 1 0 0 PQR(1,1,1) >;
        frac = < 0x1400 >;
        u-boot,dm-pre-reloc;
    };

    /* VCO = 417.8 MHz => P = 209, Q = 25, R = 11 */

```



```

pll3: st,pll@2 {
    cfg = < 1 33 1 16 36 PQR(1,1,1) >;
    frac = < 0x1a04 >;
    u-boot,dm-pre-reloc;
};

/* VCO = 594.0 MHz => P = 99, Q = 74, R = 74 */
pll4: st,pll@3 {
    cfg = < 3 98 5 7 7 PQR(1,1,1) >;
    u-boot,dm-pre-reloc;
};

```

Reset and Clock Control

Low Speed Internal oscillator (STM32 clock source)

Low Speed External oscillator (STM32 clock source)

High Speed Internal oscillator (STM32 clock source) or High Speed Synchronous Serial Interface (MIPI® Alliance standard)

High Speed External oscillator (STM32 clock source)

Multi Speed Internal oscillator (STM32 clock source)

Cortex®

First Stage Boot Loader

Linux® is a registered trademark of Linus Torvalds.

Das U-Boot -- the Universal Boot Loader (see [U-Boot_overview](#))

Secondary Program Loader, *Also known as **U-Boot SPL***

Read Only Memory

Central processing unit

Dynamic voltage and frequency scaling

Independent Watchdog

Digital-to-analog converter (Electronic circuit that converts a binary number into a continuously varying value.)

Device Tree Source (in software context) or Digital Temperature Sensor (in peripheral context)

Real Time Clock

Tamper

Consumer Electronics Control (HDMI standard)

Power Management Integrated Circuit

Display Serial Interface (MIPI® Alliance standard)

Analog-to-digital converter. The process of converting a sampled analog signal to a digital code that represents the amplitude of the original signal sample.

USB Host (STM32 specific)

Boot and Security and OTP control

Extended TrustZone Protection Controller



TrustZone[®] address space Controller for DDR

Arm[®] and TrustZone[®] are registered trademarks of Arm Limited (or its subsidiaries) in the US and/or elsewhere.

debug and test protocol, named from the Joint Test Action Group that developed it

System Trace Module

Graphics Processing Units

Doubledata rate (memory domain)

System Configuration

voltage reference buffer (STM32 specific)

Hardware Debug Port

Digital Camera Memory Interface

Inter-Processor Communication Controller

External Interrupt

Digital Filter for Sigma-Delta Modulator

Transmit

former spelling for e•MMC ('e' in italic)

Receive

LCD TFT Display Controller (STM32 specific)

System Time Generator

USB On-The-Go (Capability/type of USB port, acting primarily as USB device, to also act as USB host. Also known as USB OTG.)

input/output

Ethernet

Microprocessor Unit

Microcontroller Unit (MCUs have internal flash memory and are intended to operate with a minimum amount of external support ICs. They commonly are a self-contained, system-on-chip (SoC) designs.)

Discovery kit

BlueTooth

High-Definition Multimedia Interface (HDMI standard)

Stable: 01.03.2021 - 10:58 / Revision: 01.03.2021 - 10:55

All the peripherals receive one or several clocks that are generated via [RCC internal peripheral](#). RCC relies on several clocks sources (LSI, LSE, HSI, HSE, CSI) and four PLL in order to provide adequate input frequencies to all the peripherals. The clock tree covers all the system clock distribution aspects, from the clock sources to the consumer peripherals (internal and external), except clock gating management that is locally controlled by each peripheral driver.

Contents

1 Overview	160
2 How to build a clock tree?	161



3 ST boards clock tree	162
3.1 STM32MP157x-EV1 case	162
3.1.1 Clock tree	162
3.1.1.1 Device tree	164
3.2 STM32MP157x-DK2 case	166
3.2.1 Clock tree	166
3.2.2 Device tree	168



1 Overview

The clock tree is managed via RCC internal peripheral hardware block and it is configured at different steps from the Cortex-A7:

- When the device is **reset**, all RCC registers take their reset values: the four PLL are disabled and most of the clock source selectors are pointing to the HSI.
- The ROM Code configures the minimum clock tree needed to boot on the selected boot device.
- The **FSBL** (Cf. [Boot chain overview](#)) completely configures the clock tree as expected, all the way up to Linux, thanks to the configuration given in the [device tree](#).
 - U-Boot SPL (basic boot chain) binding is available in [doc/device-tree-bindings/clock/st,stm32mp1.txt](#)
 - TF-A (trusted boot chain) binding follows the same syntax as U-Boot SPL
- **Linux** is responsible for the clock tree at runtime and it may partly modify it thanks to information taken from the [device tree](#):
 - Linux binding is available in [Documentation/devicetree/bindings/clock/clock-bindings.txt](#) (and surrounding files): 'fixed-clock' compatible, 'clocks' and 'assigned-clocks' properties are important concepts to understand the management of clocks providers/consumers.

Notice that [STM32Cube](#) (running on the Cortex-M4) does not control the clock tree setup so the configuration applied by the Cortex-A7 must provide suitable clocks to Cortex-M4, like it is explained in [resource manager](#) description. One exception to this rule is the [engineering boot](#), allowing to directly load (and debug) the Cortex-M4 whereas the Cortex-A7 execution is stalled in the ROM code.



2 How to build a clock tree?

Building a clock tree is quite complex as it needs to take into account the constraints set by each internal and external peripheral, including external clock sources.

We encourage the use of [STM32CubeMX](#) to build the clock tree, and avoid having to know all internal peripherals details: the tool allows to select the peripherals that will be present on the board, fix the clock sources frequencies and automatically find an optimized clock tree. It is then able to generate the device tree that is directly consumed by the boot chain and Linux.



3 ST boards clock tree



See [How to change the CPU frequency](#) article for more information about the CPU frequency setting (PLL1P), including dynamic voltage and frequency scaling (DVFS) configuration.

3.1 STM32MP157x-EV1 case

This chapter shows the boot time clock tree set by the FSBL on STM32MP157x-EV1 evaluation board. Linux eventual runtime modifications are not covered here.

3.1.1 Clock tree

The following table shows what STM32MP157x-EV1 clock tree looks like, as a result of the boot chain execution with the device tree built with STM32CubeMX.

Component Comment	Parent	Frequency	Used?	
LSI	N.A.	0.032000 MHz	yes	Mandatory for IWDG, DAC
DAC	LSI	0.032000 MHz	yes	
RNG1	LSI	0.032000 MHz	yes	Input frequency should be
as low as possible				
RNG2	LSI	0.032000 MHz	yes	Input frequency should be
as low as possible				
IWDG1	LSI	0.032000 MHz	yes	
IWDG2	LSI	0.032000 MHz	yes	
LSE	N.A.	0.032768 MHz	yes	Mandatory for DTS
RTC	LSE	0.032768 MHz	yes	
TAMP	LSE	0.032768 MHz	yes	
CEC	LSE	0.032768 MHz	yes	
LPTIM4	LSE	0.032768 MHz	yes	
LPTIM5	LSE	0.032768 MHz	yes	
HSI	N.A.	64.000000 MHz	yes	
SPI4	HSI	64.000000 MHz	no	
SPI5	HSI	64.000000 MHz	no	
SPI6	HSI	64.000000 MHz	yes	
I2C4	HSI	64.000000 MHz	yes	PMIC
I2C6	HSI	64.000000 MHz	no	
I2C1	HSI	64.000000 MHz	no	
I2C2	HSI	64.000000 MHz	yes	Rpi and peripherals
I2C3	HSI	64.000000 MHz	no	
I2C5	HSI	64.000000 MHz	yes	Rpi
USART1	HSI	64.000000 MHz	yes	
USART2	HSI	64.000000 MHz	no	
USART3	HSI	64.000000 MHz	yes	Rpi
UART4	HSI	64.000000 MHz	yes	Linux console
UART5	HSI	64.000000 MHz	no	
USART6	HSI	64.000000 MHz	no	
UART7	HSI	64.000000 MHz	no	
UART8	HSI	64.000000 MHz	no	
MC01	HSI	64.000000 MHz	no	Available so can be used
HSE	N.A.	24.000000 MHz	yes	
DSIPLL	HSE	125.000000 MHz	no	DSI DPHY PLL
DSIBL	DSIPLL	125.000000 MHz	no	DSI lanebyte clock
RTCDIV	HSE	1.000000 MHz	yes	Only used when RTC source
is HSE				



ck_per	HSE	24.000000	MHz	yes	
ADC	ck_per	24.000000	MHz	yes	Can use internal divider to
be < 40MHz					
PLL1	HSE	xxx	MHz	yes	
PLL1P	PLL1	xxx	MHz	yes	
MPUDIV	PLL1P	xxx	MHz	yes	
Cortex-A7	PLL1P	xxx	MHz	yes	650 MHz or 800 MHz, see the
information box above					
MC02	Cortex-A7	xxx	MHz	no	Available so can be used
PLL2	HSE	1066.000000	MHz	yes	
PLL2P	PLL2	266.500000	MHz	yes	
AXI	PLL2P	266.500000	MHz	yes	< 266MHz
FMC	AXI	266.500000	MHz	yes	NAND flash
QSPI	AXI	266.500000	MHz	yes	NOR flash
SYSRAM	AXI	266.500000	MHz	yes	
ROM	AXI	266.500000	MHz	yes	
AHB5	AXI	266.500000	MHz	yes	< 266MHz
CRYP1	AHB5	266.500000	MHz	yes	
HASH1	AHB5	266.500000	MHz	yes	
GPI0Z	AHB5	266.500000	MHz	yes	
BKPSRAM	AHB5	266.500000	MHz	yes	
AHB6	AXI	266.500000	MHz	yes	< 266MHz
CRC1	AHB6	266.500000	MHz	yes	
MDMA	AHB6	266.500000	MHz	yes	
USBH	AHB6	266.500000	MHz	yes	USB Host
APB4	AHB6	133.250000	MHz	yes	
APB5	AHB6	66.625000	MHz	yes	
BSEC	APB5	66.625000	MHz	yes	< 67MHz
ETZPC	APB5	66.625000	MHz	yes	
TZC	APB5	66.625000	MHz	yes	
DBGAPB	AXI	133.250000	MHz	yes	JTAG & Coresight
DBGMCU	DBGAPB	66.625000	MHz	yes	
STM	DBGAPB	66.625000	MHz	yes	
PLL2Q	PLL2	533.000000	MHz	yes	
GPU	PLL2Q	533.000000	MHz	yes	< 533MHz
PLL2R	PLL2	533.000000	MHz	yes	
DDR	PLL2R	533.000000	MHz	yes	< 533MHz
PLL3	HSE	417.755859	MHz	yes	
PLL3P	PLL3	208.877930	MHz	yes	
MLAHB	PLL3P	208.877930	MHz	yes	< 209MHz
Cortex-M4	MLAHB	208.877930	MHz	yes	
SRAM1	MLAHB	208.877930	MHz	yes	
SRAM2	MLAHB	208.877930	MHz	yes	
SRAM3	MLAHB	208.877930	MHz	yes	
RETRAM	MLAHB	208.877930	MHz	yes	
AHB1	MLAHB	208.877930	MHz	yes	< 209MHz
AHB2	MLAHB	208.877930	MHz	yes	< 209MHz
DMA1	AHB2	208.877930	MHz	yes	
DMA2	AHB2	208.877930	MHz	yes	
DMAMUX	AHB2	208.877930	MHz	yes	
APB1	MLAHB	104.438965	MHz	yes	
LPTIM1	APB1	104.438965	MHz	yes	
WWDG	APB1	104.438965	MHz	yes	
APB2	MLAHB	104.438965	MHz	yes	
TIM2	MLAHB	208.877930	MHz	yes	TIM Group 1
TIM3	MLAHB	208.877930	MHz	yes	TIM Group 1
TIM4	MLAHB	208.877930	MHz	yes	TIM Group 1
TIM5	MLAHB	208.877930	MHz	yes	TIM Group 1
TIM6	MLAHB	208.877930	MHz	yes	TIM Group 1
TIM7	MLAHB	208.877930	MHz	yes	TIM Group 1
TIM12	MLAHB	208.877930	MHz	yes	TIM Group 1
TIM13	MLAHB	208.877930	MHz	yes	TIM Group 1
TIM14	MLAHB	208.877930	MHz	yes	TIM Group 1
TIM1	MLAHB	208.877930	MHz	yes	TIM Group 2
TIM8	MLAHB	208.877930	MHz	yes	TIM Group 2
TIM15	MLAHB	208.877930	MHz	yes	TIM Group 2
TIM16	MLAHB	208.877930	MHz	yes	TIM Group 2



TIM17	MLAHB	208.877930	MHz	yes	TIM Group 2
APB3	MLAHB	104.438965	MHz	yes	
LPTIM2	APB3	104.438965	MHz	yes	
LPTIM3	APB3	104.438965	MHz	yes	
SYSCFG	APB3	104.438965	MHz	yes	
VREFBUF	APB3	104.438965	MHz	yes	
DTS	APB3	104.438965	MHz	yes	
HDP	APB3	104.438965	MHz	yes	
AHB3	MLAHB	208.877930	MHz	yes	< 209MHz
CRC2	AHB3	208.877930	MHz	yes	
CRYP2	AHB3	208.877930	MHz	yes	
HASH2	AHB3	208.877930	MHz	yes	
DCMI	AHB3	208.877930	MHz	yes	Camera
IPCC	AHB3	208.877930	MHz	yes	Mailbox
AHB4	MLAHB	208.877930	MHz	yes	< 209MHz
PWR	AHB4	208.877930	MHz	yes	
RCC	AHB4	208.877930	MHz	yes	
GPIOA-K	AHB4	208.877930	MHz	yes	
EXTI	AHB4	208.877930	MHz	yes	
PLL3Q	PLL3	24.573874	MHz	yes	
SPI1	PLL3Q	24.573874	MHz	no	
SPI2	PLL3Q	24.573874	MHz	no	
SPI3	PLL3Q	24.573874	MHz	no	
DFSDM	PLL3Q	24.573874	MHz	yes	Digital micro
SAI1	PLL3Q	24.573874	MHz	no	
SAI2	PLL3Q	24.573874	MHz	yes	AudCodec 48kHz (use PLL3R)
for 44.1kHz)					
SAI3	PLL3Q	24.573874	MHz	no	
SAI4	PLL3Q	24.573874	MHz	yes	SPDIF TX 48kHz (use PLL3R)
for 44.1kHz)					
PLL3R	PLL3	11.290699	MHz	yes	
PLL4	HSE	594.000000	MHz	yes	
PLL4P	PLL4	99.000000	MHz	yes	
SDMMC1	PLL4P	99.000000	MHz	yes	µSD card
SDMMC2	PLL4P	99.000000	MHz	yes	eMMC
SDMMC3	PLL4P	99.000000	MHz	yes	
SPDIF	PLL4P	99.000000	MHz	yes	SPDIF RX
PLL4Q	PLL4	74.250000	MHz	yes	
LCD	PLL4Q	74.250000	MHz	yes	LTDC & DSI display pixel
clock					
PLL4R	PLL4	74.250000	MHz	yes	
FDCAN	PLL4R	74.250000	MHz	yes	Should be as high as
possible and < 100MHz					
STGEN	HSE	24.000000	MHz	yes	
USBPHYC	HSE	24.000000	MHz	yes	USB PHY Ctrl for USB Host
and OTG					
USBPLL	USBPHYC	48.000000	MHz	yes	
USB0	USBPLL	48.000000	MHz	yes	USB OTG
CSI	N.A.	4.000000	MHz	yes	Mandatory for IO
compensation					
ETH	N.A.	0.000000	MHz	no	ETH clocked by RGMII PHY

3.1.1.1 Device tree

Here are the corresponding device tree `rcc_clk` sub node properties consumed by the first stage boot loader (FSBL) to configure the clock tree above:

```
st,clksrc = <
    CLK_MPU_PLL1P
    CLK_AXI_PLL2P
    CLK_MCU_PLL3P
    CLK_PLL12_HSE
```



```

        CLK_PLL3_HSE
        CLK_PLL4_HSE
        CLK_RTC_LSE
        CLK_MC01_DISABLED
        CLK_MC02_DISABLED
    >;

    st,clkdiv = <
        1 /*MPU*/
        0 /*AXI*/
        0 /*MCU*/
        1 /*APB1*/
        1 /*APB2*/
        1 /*APB3*/
        1 /*APB4*/
        2 /*APB5*/
        23 /*RTC*/
        0 /*MC01*/
        0 /*MC02*/
    >;

    st,pkcs = <
        CLK_CKPER_HSE
        CLK_FMC_ACLK
        CLK_QSPI_ACLK
        CLK_ETH_DISABLED
        CLK_SDMMC12_PLL4P
        CLK_DSI_DSIPLL
        CLK_STGEN_HSE
        CLK_USBPHY_HSE
        CLK_SPI2S1_PLL3Q
        CLK_SPI2S23_PLL3Q
        CLK_SPI45_HSI
        CLK_SPI6_HSI
        CLK_I2C46_HSI
        CLK_SDMMC3_PLL4P
        CLK_USB0_USBPHY
        CLK_ADC_CKPER
        CLK_CEC_LSE
        CLK_I2C12_HSI
        CLK_I2C35_HSI
        CLK_UART1_HSI
        CLK_UART24_HSI
        CLK_UART35_HSI
        CLK_UART6_HSI
        CLK_UART78_HSI
        CLK_SPDIF_PLL4P
        CLK_FDCAN_PLL4R
        CLK_SAI1_PLL3Q
        CLK_SAI2_PLL3Q
        CLK_SAI3_PLL3Q
        CLK_SAI4_PLL3Q
        CLK_RNG1_LSI
        CLK_RNG2_LSI
        CLK_LPTIM1_PCLK1
        CLK_LPTIM23_PCLK3
        CLK_LPTIM45_LSE
    >;

    /* VCO = 1066.0 MHz => P = 266 (AXI), Q = 533 (GPU), R = 533 (DDR) */
    pll2: st,pll@1 {
        cfg = < 2 65 1 0 0 PQR(1,1,1) >;
        frac = < 0x1400 >;
        u-boot,dm-pre-reloc;
    };

    /* VCO = 417.8 MHz => P = 209, Q = 25, R = 11 */

```



```

pll3: st,pll@2 {
    cfg = < 1 33 1 16 36 PQR(1,1,1) >;
    frac = < 0x1a04 >;
    u-boot,dm-pre-reloc;
};

/* VCO = 594.0 MHz => P = 99, Q = 74, R = 74 */
pll4: st,pll@3 {
    cfg = < 3 98 5 7 7 PQR(1,1,1) >;
    u-boot,dm-pre-reloc;
};
    
```

3.2 STM32MP157x-DK2 case

This chapter shows the boot time clock tree set by the FSBL on STM32MP157x-DK2 DISCO board. Linux eventual runtime modifications are not covered here.

3.2.1 Clock tree

The following table shows what STM32MP157x-DK2 clock tree looks like, as a result of the boot chain execution with the device tree built with STM32CubeMX.

Component Comment	Parent	Frequency	Used?	
LSI	N.A.	0.032000 MHz	yes	Mandatory for IWDG, DAC
DAC	LSI	0.032000 MHz	no	
RNG1	LSI	0.032000 MHz	yes	Input frequency should be as low as possible
RNG2	LSI	0.032000 MHz	yes	Input frequency should be as low as possible
IWDG1	LSI	0.032000 MHz	yes	
IWDG2	LSI	0.032000 MHz	yes	
LSE	N.A.	0.032768 MHz	yes	Mandatory for DTS
RTC	LSE	0.032768 MHz	yes	
TAMP	LSE	0.032768 MHz	yes	
CEC	LSE	0.032768 MHz	yes	
LPTIM4	LSE	0.032768 MHz	yes	
LPTIM5	LSE	0.032768 MHz	yes	
HSI	N.A.	64.000000 MHz	yes	
SPI4	HSI	64.000000 MHz	yes	Arduino
SPI5	HSI	64.000000 MHz	yes	Rpi
SPI6	HSI	64.000000 MHz	no	
I2C4	HSI	64.000000 MHz	yes	PMIC
I2C6	HSI	64.000000 MHz	no	
I2C1	HSI	64.000000 MHz	yes	Rpi and peripherals
I2C2	HSI	64.000000 MHz	no	
I2C3	HSI	64.000000 MHz	no	
I2C5	HSI	64.000000 MHz	yes	Rpi and Arduino
USART1	HSI	64.000000 MHz	no	
USART2	HSI	64.000000 MHz	yes	Bluetooth
USART3	HSI	64.000000 MHz	yes	Rpi
UART4	HSI	64.000000 MHz	yes	Linux console
UART5	HSI	64.000000 MHz	no	
USART6	HSI	64.000000 MHz	no	
UART7	HSI	64.000000 MHz	yes	Arduino
UART8	HSI	64.000000 MHz	no	
MCO1	HSI	64.000000 MHz	no	Available so can be used
HSE	N.A.	24.000000 MHz	yes	
DSIPLL	HSE	125.000000 MHz	no	DSI DPHY PLL
DSIBL	DSIPLL	125.000000 MHz	no	DSI lanebyte clock



RTCDIV	HSE	1.000000 MHz	yes	Only used when RTC source
is HSE				
ck_per	HSE	24.000000 MHz	yes	
ADC	ck_per	24.000000 MHz	yes	Can use internal divider to
be < 40MHz				
PLL1	HSE	xxx MHz	yes	
PLL1P	PLL1	xxx MHz	yes	
MPUDIV	PLL1P	xxx MHz	yes	
Cortex-A7	PLL1P	xxx MHz	yes	650 MHz or 800 MHz, see the
information box above				
MC02	Cortex-A7	xxx MHz	no	Available so can be used
PLL2	HSE	1066.000000 MHz	yes	
PLL2P	PLL2	266.500000 MHz	yes	
AXI	PLL2P	266.500000 MHz	yes	< 266MHz
FMC	AXI	266.500000 MHz	no	
QSPI	AXI	266.500000 MHz	no	
SYSRAM	AXI	266.500000 MHz	yes	
ROM	AXI	266.500000 MHz	yes	
AHB5	AXI	266.500000 MHz	yes	< 266MHz
CRYP1	AHB5	266.500000 MHz	yes	
HASH1	AHB5	266.500000 MHz	yes	
GPIOZ	AHB5	266.500000 MHz	yes	
BKPSRAM	AHB5	266.500000 MHz	yes	
AHB6	AXI	266.500000 MHz	yes	< 266MHz
CRC1	AHB6	266.500000 MHz	yes	
MDMA	AHB6	266.500000 MHz	yes	
USBH	AHB6	266.500000 MHz	yes	USB Host
APB4	AHB6	133.250000 MHz	yes	
APB5	AHB6	66.625000 MHz	yes	
BSEC	APB5	66.625000 MHz	yes	< 67MHz
ETZPC	APB5	66.625000 MHz	yes	
TZC	APB5	66.625000 MHz	yes	
DBGAPB	AXI	133.250000 MHz	yes	JTAG & Coresight
DBGMCU	DBGAPB	66.625000 MHz	yes	
STM	DBGAPB	66.625000 MHz	no	
PLL2Q	PLL2	533.000000 MHz	yes	
GPU	PLL2Q	533.000000 MHz	yes	< 533MHz
PLL2R	PLL2	533.000000 MHz	yes	
DDR	PLL2R	533.000000 MHz	yes	< 533MHz
PLL3	HSE	417.755859 MHz	yes	
PLL3P	PLL3	208.877930 MHz	yes	
MLAHB	PLL3P	208.877930 MHz	yes	< 209MHz
Cortex-M4	MLAHB	208.877930 MHz	yes	
SRAM1	MLAHB	208.877930 MHz	yes	
SRAM2	MLAHB	208.877930 MHz	yes	
SRAM3	MLAHB	208.877930 MHz	yes	
RETRAM	MLAHB	208.877930 MHz	yes	
AHB1	MLAHB	208.877930 MHz	yes	< 209MHz
AHB2	MLAHB	208.877930 MHz	yes	< 209MHz
DMA1	AHB2	208.877930 MHz	yes	
DMA2	AHB2	208.877930 MHz	yes	
DMAMUX	AHB2	208.877930 MHz	yes	
APB1	MLAHB	104.438965 MHz	yes	
LPTIM1	APB1	104.438965 MHz	yes	
WWDG	APB1	104.438965 MHz	yes	
APB2	MLAHB	104.438965 MHz	yes	
TIM2	MLAHB	208.877930 MHz	yes	TIM Group 1
TIM3	MLAHB	208.877930 MHz	yes	TIM Group 1
TIM4	MLAHB	208.877930 MHz	yes	TIM Group 1
TIM5	MLAHB	208.877930 MHz	yes	TIM Group 1
TIM6	MLAHB	208.877930 MHz	yes	TIM Group 1
TIM7	MLAHB	208.877930 MHz	yes	TIM Group 1
TIM12	MLAHB	208.877930 MHz	yes	TIM Group 1
TIM13	MLAHB	208.877930 MHz	yes	TIM Group 1
TIM14	MLAHB	208.877930 MHz	yes	TIM Group 1
TIM1	MLAHB	208.877930 MHz	yes	TIM Group 2
TIM8	MLAHB	208.877930 MHz	yes	TIM Group 2



TIM15	MLAHB	208.877930	MHz	yes	TIM Group 2
TIM16	MLAHB	208.877930	MHz	yes	TIM Group 2
TIM17	MLAHB	208.877930	MHz	yes	TIM Group 2
APB3	MLAHB	104.438965	MHz	yes	
LPTIM2	APB3	104.438965	MHz	yes	
LPTIM3	APB3	104.438965	MHz	yes	
SYSCFG	APB3	104.438965	MHz	yes	
VREFBUF	APB3	104.438965	MHz	yes	
DTS	APB3	104.438965	MHz	yes	
HDP	APB3	104.438965	MHz	no	
AHB3	MLAHB	208.877930	MHz	yes	< 209MHz
CRC2	AHB3	208.877930	MHz	yes	
CRYP2	AHB3	208.877930	MHz	yes	
HASH2	AHB3	208.877930	MHz	yes	
DCMI	AHB3	208.877930	MHz	no	
IPCC	AHB3	208.877930	MHz	yes	Mailbox
AHB4	MLAHB	208.877930	MHz	yes	< 209MHz
PWR	AHB4	208.877930	MHz	yes	
RCC	AHB4	208.877930	MHz	yes	
GPIOA-K	AHB4	208.877930	MHz	yes	
EXTI	AHB4	208.877930	MHz	yes	
PLL3Q	PLL3	24.573874	MHz	yes	
SPI1	PLL3Q	24.573874	MHz	yes	SPI2S1 for BT PCM
SPI2	PLL3Q	24.573874	MHz	yes	SPI2S2 for HDMI
SPI3	PLL3Q	24.573874	MHz	no	
DFSDM	PLL3Q	24.573874	MHz	no	
SAI1	PLL3Q	24.573874	MHz	no	
SAI2	PLL3Q	24.573874	MHz	yes	AudCodec 48kHz (use PLL3R)
for 44.1kHz)					
SAI3	PLL3Q	24.573874	MHz	no	
SAI4	PLL3Q	24.573874	MHz	no	
PLL3R	PLL3	11.290699	MHz	yes	
PLL4	HSE	594.000000	MHz	yes	
PLL4P	PLL4	99.000000	MHz	yes	
SDMMC1	PLL4P	99.000000	MHz	yes	µSD card
SDMMC2	PLL4P	99.000000	MHz	yes	Wifi
SDMMC3	PLL4P	99.000000	MHz	yes	Rpi
SPDIF	PLL4P	99.000000	MHz	no	
PLL4Q	PLL4	74.250000	MHz	yes	
LCD	PLL4Q	74.250000	MHz	yes	LTDC & DSI display pixel
clock					
PLL4R	PLL4	74.250000	MHz	yes	
FDCAN	PLL4R	74.250000	MHz	no	
STGEN	HSE	24.000000	MHz	yes	
USBPHYC	HSE	24.000000	MHz	yes	USB PHY Ctrl for USB Host
and OTG					
USBPLL	USBPHYC	48.000000	MHz	yes	
USB0	USBPLL	48.000000	MHz	yes	USB OTG
CSI	N.A.	4.000000	MHz	yes	Mandatory for IO
compensation					
ETH	N.A.	0.000000	MHz	no	ETH clocked by RGMII PHY

3.2.2 Device tree

Here are the corresponding device tree `rcc_clk` sub node properties consumed by the first stage boot loader (FSBL) to configure the clock tree above:

```
st,clksrc = <
    CLK_MPU_PLL1P
    CLK_AXI_PLL2P
    CLK_MCU_PLL3P
    CLK_PLL12_HSE
```




```

        CLK_PLL3_HSE
        CLK_PLL4_HSE
        CLK_RTC_LSE
        CLK_MC01_DISABLED
        CLK_MC02_DISABLED
    >;

    st,clkdiv = <
        1 /*MPU*/
        0 /*AXI*/
        0 /*MCU*/
        1 /*APB1*/
        1 /*APB2*/
        1 /*APB3*/
        1 /*APB4*/
        2 /*APB5*/
        23 /*RTC*/
        0 /*MC01*/
        0 /*MC02*/
    >;

    st,pkcs = <
        CLK_CKPER_HSE
        CLK_FMC_ACLK
        CLK_QSPI_ACLK
        CLK_ETH_DISABLED
        CLK_SDMMC12_PLL4P
        CLK_DSI_DSIPLL
        CLK_STGEN_HSE
        CLK_USBPHY_HSE
        CLK_SPI2S1_PLL3Q
        CLK_SPI2S23_PLL3Q
        CLK_SPI45_HSI
        CLK_SPI6_HSI
        CLK_I2C46_HSI
        CLK_SDMMC3_PLL4P
        CLK_USB0_USBPHY
        CLK_ADC_CKPER
        CLK_CEC_LSE
        CLK_I2C12_HSI
        CLK_I2C35_HSI
        CLK_UART1_HSI
        CLK_UART24_HSI
        CLK_UART35_HSI
        CLK_UART6_HSI
        CLK_UART78_HSI
        CLK_SPDIF_PLL4P
        CLK_FDCAN_PLL4R
        CLK_SAI1_PLL3Q
        CLK_SAI2_PLL3Q
        CLK_SAI3_PLL3Q
        CLK_SAI4_PLL3Q
        CLK_RNG1_LSI
        CLK_RNG2_LSI
        CLK_LPTIM1_PCLK1
        CLK_LPTIM23_PCLK3
        CLK_LPTIM45_LSE
    >;

    /* VCO = 1066.0 MHz => P = 266 (AXI), Q = 533 (GPU), R = 533 (DDR) */
    pll2: st,pll@1 {
        cfg = < 2 65 1 0 0 PQR(1,1,1) >;
        frac = < 0x1400 >;
        u-boot,dm-pre-reloc;
    };

    /* VCO = 417.8 MHz => P = 209, Q = 25, R = 11 */

```



```

pll3: st,pll@2 {
    cfg = < 1 33 1 16 36 PQR(1,1,1) >;
    frac = < 0x1a04 >;
    u-boot,dm-pre-reloc;
};

/* VCO = 594.0 MHz => P = 99, Q = 74, R = 74 */
pll4: st,pll@3 {
    cfg = < 3 98 5 7 7 PQR(1,1,1) >;
    u-boot,dm-pre-reloc;
};

```

Reset and Clock Control

Low Speed Internal oscillator (STM32 clock source)

Low Speed External oscillator (STM32 clock source)

High Speed Internal oscillator (STM32 clock source) or High Speed Synchronous Serial Interface (MIPI® Alliance standard)

High Speed External oscillator (STM32 clock source)

Multi Speed Internal oscillator (STM32 clock source)

Cortex®

First Stage Boot Loader

Linux® is a registered trademark of Linus Torvalds.

Das U-Boot -- the Universal Boot Loader (see [U-Boot_overview](#))

Secondary Program Loader, *Also known as **U-Boot SPL***

Read Only Memory

Central processing unit

Dynamic voltage and frequency scaling

Independent Watchdog

Digital-to-analog converter (Electronic circuit that converts a binary number into a continuously varying value.)

Device Tree Source (in software context) or Digital Temperature Sensor (in peripheral context)

Real Time Clock

Tamper

Consumer Electronics Control (HDMI standard)

Power Management Integrated Circuit

Display Serial Interface (MIPI® Alliance standard)

Analog-to-digital converter. The process of converting a sampled analog signal to a digital code that represents the amplitude of the original signal sample.

USB Host (STM32 specific)

Boot and Security and OTP control

Extended TrustZone Protection Controller



TrustZone® address space Controller for DDR

Arm® and TrustZone® are registered trademarks of Arm Limited (or its subsidiaries) in the US and/or elsewhere.

debug and test protocol, named from the Joint Test Action Group that developed it

System Trace Module

Graphics Processing Units

Doubledata rate (memory domain)

System Configuration

voltage reference buffer (STM32 specific)

Hardware Debug Port

Digital Camera Memory Interface

Inter-Processor Communication Controller

External Interrupt

Digital Filter for Sigma-Delta Modulator

Transmit

former spelling for e•MMC ('e' in italic)

Receive

LCD TFT Display Controller (STM32 specific)

System Time Generator

USB On-The-Go (Capability/type of USB port, acting primarily as USB device, to also act as USB host. Also known as USB OTG.)

input/output

Ethernet

Microprocessor Unit

Microcontroller Unit (MCUs have internal flash memory and are intended to operate with a minimum amount of external support ICs. They commonly are a self-contained, system-on-chip (SoC) designs.)

Discovery kit

BlueTooth

High-Definition Multimedia Interface (HDMI standard)