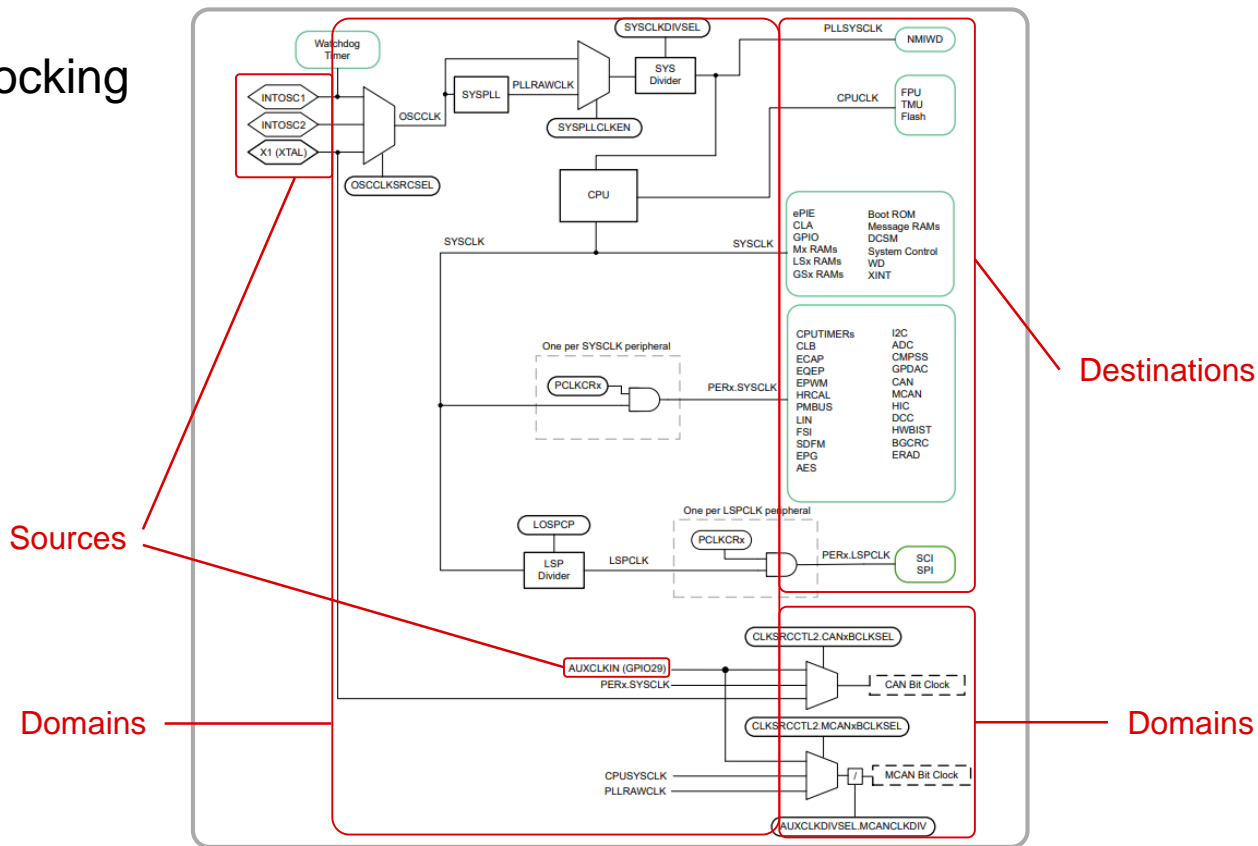


C2000 SysConfig – ClockTree Tool

GUI-Based Clocking for C2000 MCUs

Clock Tree Overview

F28003x Clocking Diagram



Clock Tree Tool Overview

The screenshot shows the SysConfig application window with the following components:

- Menu Bar:** FILE, ABOUT, RESTART
- Search Bar:** Type Filter Text...
- Left Panel (ClockTree Domain View):**
 - COMPLETE CLOCKTREE (1)
 - ClockTree
 - CLOCK DOMAIN VIEW (9)
 - CANA Clock
 - CLB Clock
 - CPUCLK
 - Low Speed Clock
 - MCAN Clock
 - NMI Watchdog
 - Peripheral Clock
 - SYSCLK
 - Watchdog
- Main Area (Complete ClockTree Overview):** A detailed clock tree diagram showing various clock sources and their connections to different components.
- Bottom Left (Clock Settings):**

AUXCLK	
Description	Auxillary Clock
XTAL Frequency (in MHz)	20
Min XTAL Frequency supported (in M...	10
Max XTAL Frequency supported (in M...	60
- Bottom Right (Views):** A status message: "There are no problems in the current design."

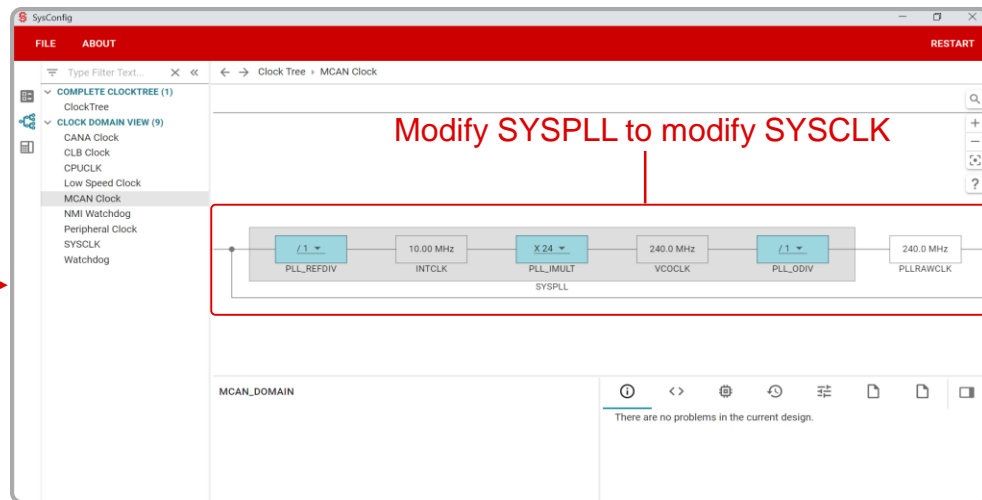
ClockTree Domain View

Complete ClockTree Overview

Clock Settings

Views

Clock Domain View



Configuring Clock Tree Settings

The screenshot displays a configuration interface for clock tree settings. A dropdown menu is open for the PLL_REFDIV parameter, showing options from /1 to /13. A red box highlights this menu, with a red line pointing to the label 'Value Dropdown Menu'. Below the menu, a table summarizes the SYSPLL configuration. A red box highlights the 'SYSPLL Reference Clock Divide' section of the table, with a red line pointing to the label 'Divider Values'.

Parameter	Value
INTCLK	10.00 MHz
PLL_IMULT	X 24
SYSPLL	
PLL_REFDIV	
Description	SYSPLL Reference Clock Divide
Input Clock	10 MHz
Divide Value	/ 1
INTCLK	10.00 MHz

Value Dropdown Menu

Divider Values

SysConfig Error

Domains effected by the error

Clocking Violation Error

Configured VCOCCLK = 48 MHz. Minimum frequency supported = 220 MHz

PLL_REFDIV	
Description	SYSPLL Reference Clock Divide
Input Clock	10 MHz
Divide Value	/ 5
INTCLK	2.000 MHz

ERRORS 1	
Location	Details
VCOCCLK	Configured VCOCCLK = 48 MHz. Minimum frequency supported = 220 MHz

Modify Multiplier Value

The screenshot displays a PLL configuration interface. A dropdown menu is open, showing multiplier values from X 115 to X 127. The PLL is currently configured with a VCOCLK of 48.00 MHz and a PLL_ODIV of /1. An error/warnings window is open at the bottom, indicating that the configured VCOCLK (48 MHz) is not supported (supported range is up to 220 MHz).

Value Dropdown Menu

48.00 MHz
VCOCLK

/1
PLL_ODIV

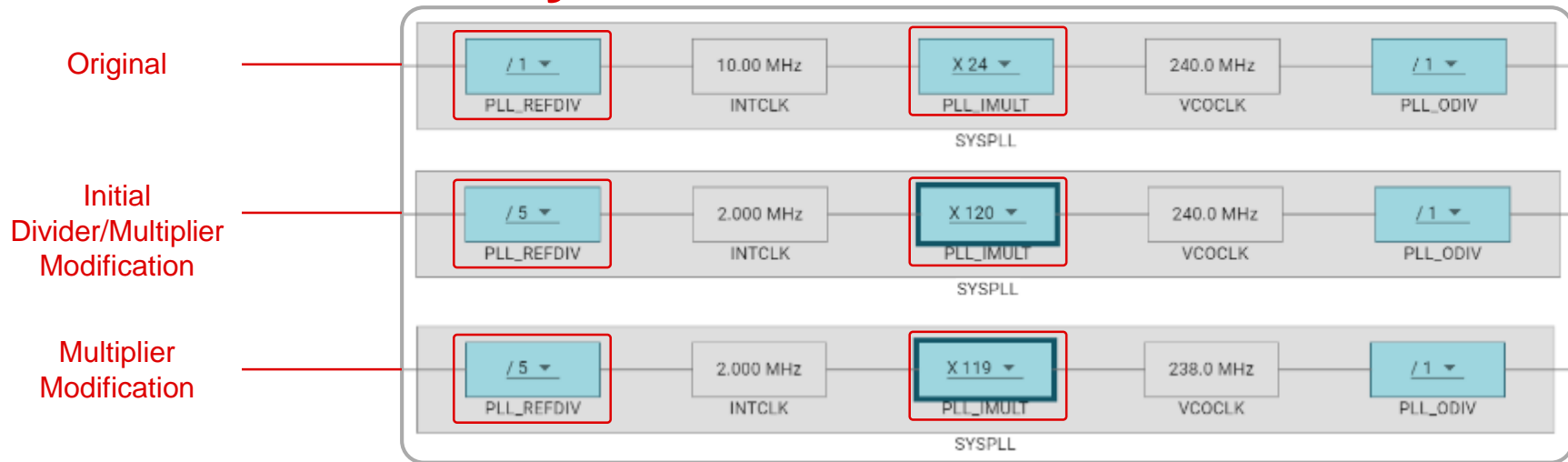
multiplier

ERRORS 1 WARNINGS 0 SUPPRESSED 0

Location	Details
VCOCLK	Configured VCOCLK = 48 MHz supported = 220 MHz

Error/Warnings Window

Modification History

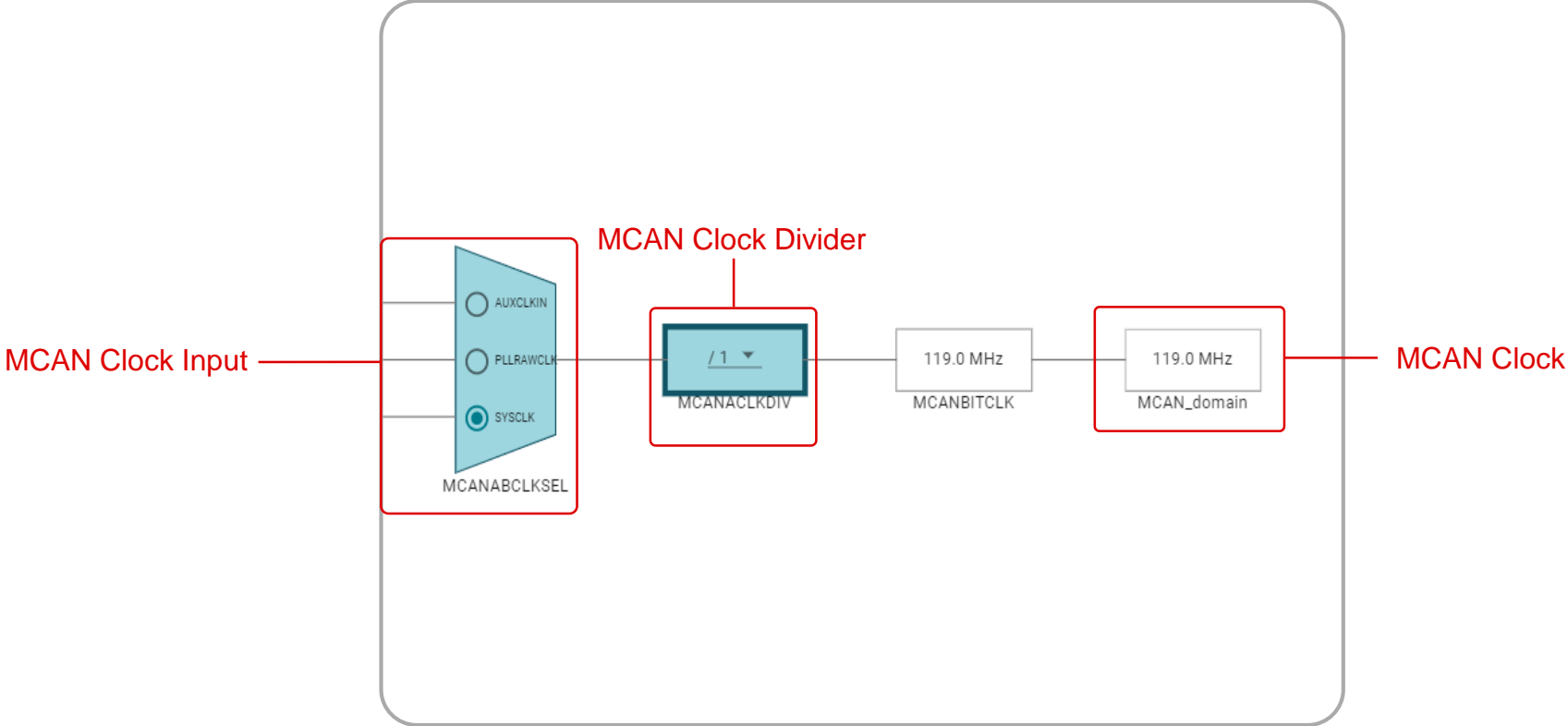


The screenshot shows a modification history panel with a list of changes:

- Changed "Multiply Value" on PLL_IMULT (Latest)
- Changed "Divide Value" on PLL_REFDIV (4)
- Changed "Multiply Value" on PLL_IMULT (3)
- Changed "Divide Value" on PLL_REFDIV (2)
- Initial state (1)

History

Effect on the **MCAN Clock**



Software Impact

clocktree.h

```
clocktree.h
43 43
44 44 //*****
45 45 //
46 46 // Summary of SYSPLL related clock configuration
47 47 //
48 48 //*****
49 49 //
50 50 // Input Clock to SYSPLL (OSCLK) = 10 MHz (INTOSC2 provides OSCLK)
51 51 //
52 52 //#### SYSPLL ENABLED ####
53 53 //
54 54 //-- PLLRAMCLK = 240 MHz (Output of SYSPLL if enabled)
55 55 //-- PLLSYSCLK = 120 MHz
56 56 //-- CPUCCLK = 120 MHz
57 57 //-- SYSCLK = 120 MHz
58 58 //-- LSPCLK = 60 MHz
59 59
60 60 //*****
61 61 //
62 62 // Macro definitions used in device.c (SYSPLL / LSPCLK)
63 63 //
64 64 //*****
65 65 //
66 66 // Input Clock to SYSPLL (OSCLK) = INTOSC2 = 10 MHz
67 67 //
68 68 #define DEVICE_OSCSRC_FREQ 10000000U
69 69 //
70 70 // Define to pass to SysCtl_setClock(). Will configure the clock as follows:
71 71 // SYSPLL ENABLED
72 72 //-- SYSCLK = 120 MHz = 10 MHz (OSCLK) * 120 (IMULT) / (5 (REFDIV) * 1 (ODIV) * 2 (SYSCLKDIVSEL))
73 73 #define DEVICE_SYSCLK_FREQ ((DEVICE_OSCSRC_FREQ * 120) / (5 * 1 * 2))
74 74 //
75 75 //-- SYSCLK = 119 MHz = 10 MHz (OSCLK) * 119 (IMULT) / (5 (REFDIV) * 1 (ODIV) * 2 (SYSCLKDIVSEL))
76 76 #define DEVICE_SYSCLK_FREQ ((DEVICE_OSCSRC_FREQ * 119) / (5 * 1 * 2))
77 77 //
78 78 //
79 79 //
80 80 //
```

Clock Configuration

device.c

```
device.c
69 69 void Device_init(void)
70 70 {
71 71 //
72 72 // Disable the watchdog
73 73 //
74 74 SysCtl_disableWatchdog();
75 75 #ifdef CHDTOOL
76 76 CHD_init();
77 77 #endif
78 78 //
79 79 #ifdef _FLASH
80 80 #ifdef CHDTOOL
81 81 //
82 82 // Copy time critical code and flash setup code to RAM. This includes the
83 83 // following functions: InitFlash();
84 84 //
85 85 // The RamfuncsLoadStart, RamfuncsLoadSize, and RamfuncsRunStart symbols
86 86 // are created by the linker. Refer to the device .cmd file.
87 87 //
88 88 memcpy(&RamfuncsRunStart, &RamfuncsLoadStart, (size_t)&RamfuncsLoadSize);
89 89 #endif
90 90 //
91 91 // Call Flash Initialization to setup flash waitstates. This function must
92 92 // reside in RAM.
93 93 //
94 94 Flash_initModule(FLASH0CTRL_BASE, FLASH0ECC_BASE, DEVICE_FLASH_WAITSTATES);
95 95 #endif
96 96 //
97 97 //
98 98 // Set up PLL control and clock dividers
99 99 //
100 100 SysCtl_setClock(DEVICE_SETCLK_CFG);
101 101 SysCtl_setLowSpeedClock(DEVICE_LSPCLK_CFG);
102 102 //
103 103 //
104 104 //
105 105 // These asserts will check that the #defines for the clock rates in
106 106 // device.h match the actual rates that have been configured. If they do
107 107 // not match, check that the calculations of DEVICE_SYSCLK_FREQ and
108 108 // DEVICE_LSPCLK_FREQ are accurate. Some examples will not perform as
109 109 // expected if these are not correct.
110 110 //
111 111 CAN_selectClockSource(CANA_BASE, CAN_CLOCK_SOURCE_SYS);
112 112 SysCtl_setMCANCLK(SYSCTL_MCANCLK_DIV1);
113 113 ASSERT(SysCtl_getClock(DEVICE_OSCSRC_FREQ) == DEVICE_SYSCLK_FREQ);
114 114
```

Device Clock Setup

Resources

- [C2000 SysConfig Video Series](#)
- Benefits of C2000 SysConfig:
 - [Speed Up Development With C2000™ Real-Time MCUs Using SysConfig](#)
- Application Report:
 - [C2000 SysConfig](#)
- Getting started with software
 - [Software Guide](#)