



Mir Jeffres, Karan Das

ABSTRACT

The TAx5x1x family of devices has a flexible PLL and clock tree that can support a wide range of input and output clocking frequencies. For standard audio rates of WCLK, BCLK, and CCLK, the device can automatically configure the clock tree for performance, but if more control or non-standard audio rates are desired, the user can customize the clocking configuration as needed. The manual clocking mode involves manually setting all clock dividers and PLL parameters and can be used for both target (all ASI clocks provided externally) and controller mode (CCLK or BCLK provided, BCLK and/or WCLK generated by device). This app note serves to make the calculations of PLL parameters and clock dividers more straightforward.

This application note focuses on custom mode, and more about the automatic modes can be found in [Clocking Configuration of Device and Flexible Clocking For TAx5x1x Family](#) as well as [Clock Error Configuration, Detection, and Modes Supported in TAx5x1x Family](#).

The clocking configuration described in this application note is available for the following devices:

TAC5111, TAC5111-Q1, TAC5112, TAC5112-Q1, TAC5211, TAC5212, TAC5212-Q1, TAC5301-Q1, TAC5311-Q1, TAC5312-Q1, TAC5411-Q1, TAC5412-Q1, TAD5112, TAD5112-Q1, TAA5212, TAD5212, TAD5212-Q1, TAA5412-Q1

Table of Contents

1 Introduction	2
2 Clock Tree	3
3 Clocking Limits	4
3.1 Internal Frequency Ranges.....	4
3.2 Clock Divider Parameter Ranges.....	4
3.3 Clock Enable Registers.....	5
4 Manual Clock Tree Configuration Registers	6
4.1 Required Register Settings.....	6
4.2 Register Settings in Manual Clock Tree Configuration.....	6
4.3 Sample Rate and MOD_CLK Frequency Bins.....	7
4.4 Power Consumption Considerations.....	8
5 Calculating PLL Multiplier and Dividers	9
6 Example 1: Custom Target Mode, Non-Audio Rates	11
6.1 Example Script.....	13
7 Example 2: Custom Controller Mode, Non-Audio Rates	14
7.1 Example Script.....	16
8 Summary	17
9 References	17

Trademarks

All trademarks are the property of their respective owners.

1 Introduction

The TAx5x1x devices support a few automatic modes of PLL configuration, where the internal clocks get configured based on input clocks (CCLK, BCLK, FSYNC). While the automatic modes for this family of devices can cover most typical use cases, there are times when manual clocking is needed or preferred. Some example use cases of manual clocking include:

1. When in controller mode and CCLK has no integer relation with FSYNC frequency, and is not one of the allowed CCLK frequencies for auto CCLK fixed mode found in Table 3-7 of [Clocking Configuration of Device and Flexible Clocking For TAx5x1x Family](#).
 - a. For example, 15MHz CCLK and 48kHz FSYNC.
2. Non-standard sample rate with specific clocking needs. Automatic mode sample rate detection will work only for specific ranges of WCLK frequency, and semi-automatic mode will work for any supported sample rate, but you lose knowledge of the internal clocks.
 - a. For example, if a user wants to use a sample rate of 50kHz (is supported in semi-automatic mode, not automatic mode) and specify what frequency of PDM clock, CLKOUT, or verify the PLL is turned on or off.
3. Sample rate conversion at non-standard sample rates, including PASI and SASI at different rates, or one in controller and one in target mode. See [TAx5x1x Synchronous Sample Rate Conversion](#) for more information.
4. Any other situation that requires intimate knowledge of internal clocking schemes on the TAx5x1x. Knowing internal clocks is beneficial to generate a general CLKOUT, PDM clock, or to be able to recreate settings for uniform power consumption between chips.

2 Clock Tree

The TA5x1x devices support a wide range of options for generating internal clocks for the ADC and DAC as well as the ASI clocks such as BCLK and FSYNC for both PASI and SASI. They also support generating the clocks used to interface with other parts of an embedded system, such as PDM clock for digital microphones and an adjustable CLKOUT which can be used for debugging or driving another device. Figure 2-1 shows the clock tree for the internal clocks and dividers. The source for all internal clocks can come from an applied CCLK or BCLK, or one of the applied clocks can be the source to the PLL which can then drive all internal clock dividers. Note that the BOOST, SAR, MOD, and DEM clocks are solely internal clocks, while the others can be routed out of a respective digital output pin if desired. More about calculating the required internal clocks and dividers for a custom manual clocking configuration follows in this application note.

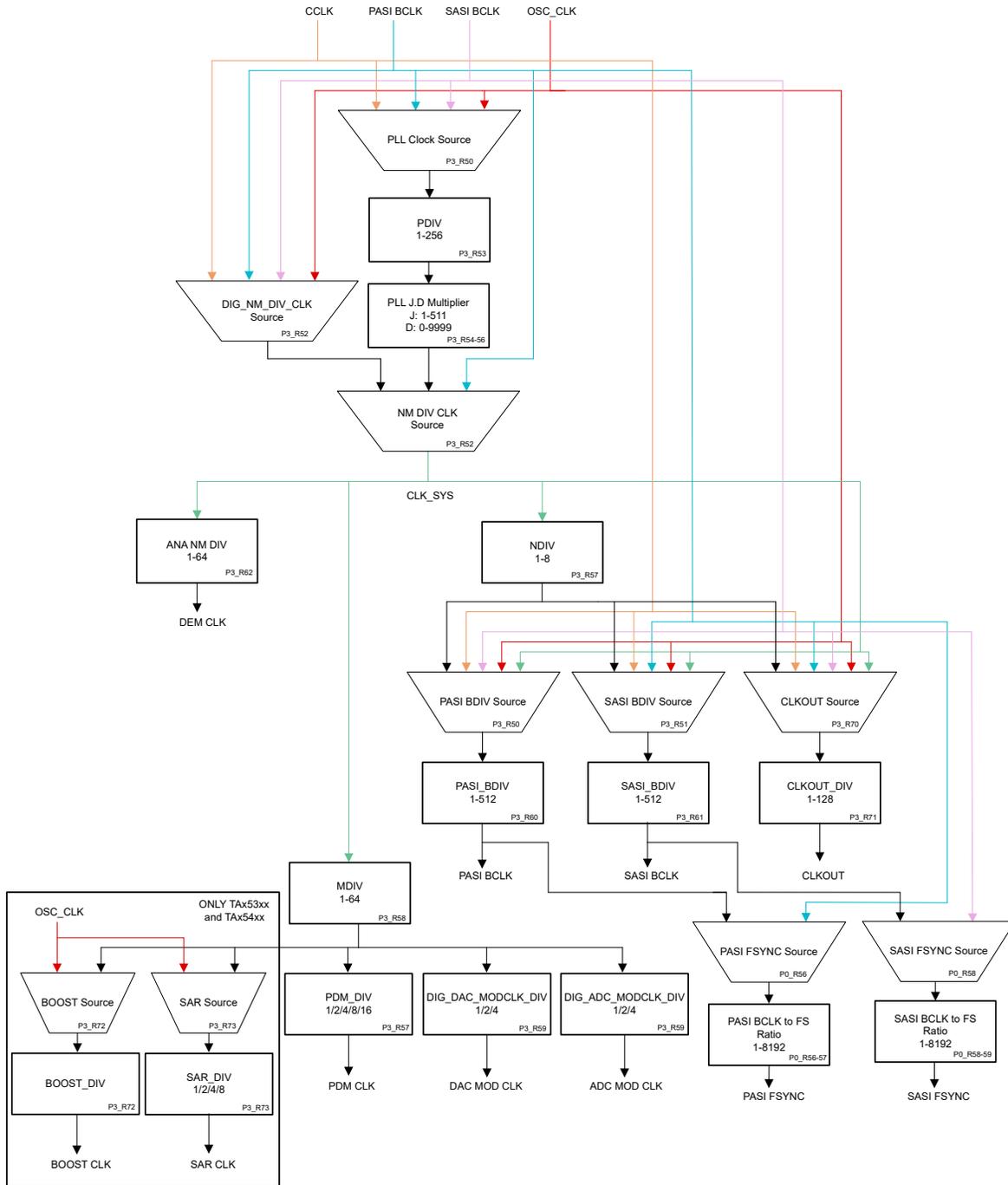


Figure 2-1. Clock Tree

3 Clocking Limits

Note these limits to the inputs to the clocking system. Clocks provided outside of this range can cause unexpected behavior.

Table 3-1. Input Frequency Ranges

Clock Input	Frequency Range (+/- 5% variance allowed as a multiple of FSYNC)
WCLK/FSYNC	3kHz-768kHz
BCLK	256kHz-24.576MHz
CCLK	Up to 50MHz

3.1 Internal Frequency Ranges

Table 3-2 lists the absolute minimum and maximum frequency ranges of the inputs and outputs of the PLL. Table 3-3 lists the internal clock frequency ranges. Note the expected values for internal clocks can vary by +/- 5% due to input clock variation from FSYNC frequency bins, as shown in Table 4-3.

Table 3-2. PLL Input/Output Clock Frequency Ranges

PLL Input/Output Clock	Frequency Range
PLL input, integer mode (D=0)	256kHz-20MHz
PLL input, fractional mode (D!=0)	10MHz-20MHz
CLK_SYS (PLL output)	67MHz-116MHz

Table 3-3. Internal Clock Frequency Ranges

Internal Clock	Frequency Range (+/- 5% Variance Allowed as a Multiple of FSYNC)
PDM_CLK	0.768/1.536/3.072/6.144 MHz
ADC_MOD_CLK/DAC_MOD_CLK	1.536/3.072MHz
ADC_DEM_CLK/DAC_DEM_CLK	2x or 4x MOD_CLK, 3.072/6.144/12.288MHz
OSC_CLK ¹	12.288MHz (+/- 10% due to device variation)
BOOST_CLK	3.072/6.144MHz
SAR_CLK	1.536/3.072/6.144/12.288MHz

3.2 Clock Divider Parameter Ranges

Table 3-4. Clock Divider Parameter Ranges

Parameter	Range	Register (all page 3)
J	1-511	0x36, 0x38
D	0-9999	0x36, 0x37
PDIV	1-256	0x35
NDIV	1-8	0x39
PDM_DIV	1, 2, 4, 8, 16	0x39
MDIV	1-64	0x3A
DIG_ADC_MODCLK_DIV	1, 2, 4	0x3A
DIG_DAC_MODCLK_DIV	1, 2, 4	0x3B
PASI BDIV	1-512	0x3B, 0x3C
SASI BDIV	1-512	0x3B, 0x3D
ANA_NM_DIV	1-64	0x3E
CLKOUT_DIV	1-128	0x47
BOOST_DIV	1, 2, 4, 8	0x48 (TAx53xx, TAx54xx only)

¹ For low power ultrasound and acoustic activity detections, internal oscillator clock OSC_CLK is available as a clock source, but not recommended to be used for playback and record applications due to variations between devices and with temperature/voltage.

Table 3-4. Clock Divider Parameter Ranges (continued)

Parameter	Range	Register (all page 3)
SAR_DIV	1, 2, 4, 8	0x49 (TAx53xx, TAx54xx only)

3.3 Clock Enable Registers

Table 3-5. Clock Enable Registers

Parameter	Register (page 3)	Bit
NDIV_EN	0x44	2
MDIV_EN	0x44	1
PDM_DIV_EN	0x44	0
DIG_ADC_MODCLK_DIV_EN	0x45	6
DIG_DAC_MODCLK_DIV_EN	0x45	4
PASI_BDIV_EN	0x45	3
SASI_BDIV_EN	0x45	2
PASI_FSYNC_DIV_EN	0x45	1
SASI_FSYNC_DIV_EN	0x45	0
CLKOUT_DIV_EN	0x47	7
BST_CLK_MANUAL_EN	0x48	2
SAR_CLK_MANUAL_EN	0x49	2

4 Manual Clock Tree Configuration Registers

4.1 Required Register Settings

To enable manual clock configuration on the TAx5x1x devices, these registers must be set. Then, all other desired clock divider registers can be set.

Table 4-1. Required Register Settings to Enable Operation in Manual Clock Config

I2C Bits	Register Setting
CUSTOM_CLK_CFG (B0_P0)	0x32[0] must be 1b
PASI_SAMP_RATE (B0_P0)	0x32[7:2] set according to Table 4-3
SASI_SAMP_RATE (B0_P0)	0x32[7:2] set according to Table 4-3 if secondary ASI is desired
CLK_CFG30 (B0_P3)	0x44[1] set to 1 to enable MDIV, other dividers enable if used
CLK_CFG31 (B0_P3)	0x45[7:4] set to 1 to enable internal clocks for ADC/DAC operation, other dividers enable if used

4.2 Register Settings in Manual Clock Tree Configuration

[Table 4-2](#) details all the registers that can be used in manual clock configuration. Not all need to be set every time, this is a general guide of all the registers that can be used that were not mentioned in [Table 3-4](#) or [Table 3-5](#). Please refer to the datasheet for full register descriptions.

Table 4-2. Register Settings in Manual Clock Tree Configuration

Register Name	Parameters	Page	Register
CLK_CFG0	PASI sample rate, enable custom clock config	0	0x32
CLK_CFG1	SASI sample rate, if desired	0	0x33
CLK_CFG2	Enable/disable PLL, set PLL fractional mode, PLL input clock source select MUX	0	0x34
CNT_CLK_CFG0	Controller mode: Select PDM_CLK, CCLK and FSYNC ratio select MSB (auto MCLK fixed mode)	0	0x35
CNT_CLK_CFG1	Controller mode: CCLK and FSYNC ratio LSB (for auto MCLK fixed mode)	0	0x36
CNT_CLK_CFG2	Controller mode: Select CCLK input frequency (for auto MCLK fixed mode), select PASI or SASI in target or controller mode	0	0x37
CNT_CLK_CFG3	Controller mode: Select PASI internal or external BCLK in controller mode, select PASI BCLK to FSYNC ratio MSB	0	0x38
CNT_CLK_CFG4	Controller mode: Select PASI BCLK to FSYNC ratio LSB	0	0x39
CNT_CLK_CFG5	Controller mode: Select SASI internal or external BCLK in controller mode, select SASI BCLK to FSYN ratio MSB	0	0x3a
CNT_CLK_CFG6	Controller mode: Select SASI BCLK to FSYNC ratio LSB	0	0x3b
PWR_TUNE_CFG0	Select ADC MOD CLK 3.072MHz or 1.536MHz frequency ranges	0	0x4e
PWR_TUNE_CFG1	Select DAC MOD CLK 3.072MHz or 1.536MHz frequency ranges	0	0x4f
CLK_CFG12	PLL_PDIV input clock selection mux, PASI BCLK divider clock source mux	3	0x32
CLK_CFG13	SASI BCLK divider clock source mux, if desired	3	0x33
CLK_CFG14	DIG_NM_DIV_CLK and NM_DIV_CLK source select mux	3	0x34
CLK_CFG15	PLL PDIV value, 1-256	3	0x35
CLK_CFG16	PLL_DIV divide by 2, J and D MSB	3	0x36
CLK_CFG17	PLL D LSB	3	0x37
CLK_CFG18	PLL J LSB	3	0x38
CLK_CFG19	NDIV, PDM_DIV select	3	0x39
CLK_CFG20	MDIV, DIG_ADC_MODCLK_DIV select	3	0x3a
CLK_CFG21	DIG_DAC_MODCLK_DIV select, PASI and SASI BCLK divider MSB	3	0x3b
CLK_CFG22	PASI BCLK divider LSB	3	0x3c
CLK_CFG23	SASI BCLK divider LSB	3	0x3d
CLK_CFG24	Analog N-M DIV value	3	0x3e

Table 4-2. Register Settings in Manual Clock Tree Configuration (continued)

Register Name	Parameters	Page	Register
CLK_CFG30	enable NDIV, MDIV, and PDM DIV	3	0x44
CLK_CFG31	enable ADC MOD CLK, DAC MOD CLK, PASI BDIV, SASI BDIV, PASI FSYNC DIV, SASI FSYNC DIV dividers	3	0x45
CLKOUT_CFG1	CLKOUT divider source selection mux	3	0x46
CLKOUT_CFG2	CLKOUT divider value and enable	3	0x47
BSTCLK_CFG1	Boost clock frequency, divider enable, and divider value (TAX53xx and TAX54xx only)	3	0x48
SARCLK_CFG1	SAR clock frequency, divider enable, and divider value (TAX53xx and TAX54xx only)	3	0x49

4.3 Sample Rate and MOD_CLK Frequency Bins

The frequency of MOD_CLK changes with the sample rate, also called FS. The TAX5x1x devices have many defined sample rate frequency bins, with the center frequency of the bin being the expected frequency that would allow the MOD_CLK to be either 3.072MHz or 1.536MHz. The sample rate relates to the MOD_CLK by an integer multiple called the oversampling rate (OSR).

Table 4-3 shows all the possible bins in the device, along with the OSR. The Bin ID is used in register P0_R50 and P0_R51 when specifying the sample rate, with the decimal value of the bin in bits 7-2. These are the same bins used for semi-automatic mode. In manual mode, pay special attention to the OSR associated with each bin. Sample rates can be any value between the minimum and maximum FS shown for each bin, and some frequencies may fall into multiple bins. For these scenarios, choose the bin with the typical value that is closest to the desired sample rate for best performance.

There are two options for the internal MOD_CLK, which affects the oversampling rate of the delta-sigma ADC/DAC. More about the considerations of the options is in Section 4.4.

Table 4-3. Mod Clock and Sample Rate Frequency Bins

MOD_CLK=3.072MHz Mode					MOD_CLK=1.536MHz Mode				
Bin ID	OSR	Min FS	Typical FS	Max FS	Bin ID	OSR	Min FS	Typical FS	Max FS
1	4	670320	768000	806400	1	2	670320	768000	806400
2	5	536256	614400	645120	Not supported				
3	6	446880	512000	537600	3	3	446880	512000	537600
4	7	383040	438857.143	460800	Not supported				
5	8	335160	384000	403200	5	4	335160	384000	403200
6	9	297920	341333.333	358400	Not supported				
7	10	268128	307200	322560	7	5	268128	307200	322560
8	12	223440	256000	268800	8	6	223440	256000	268800
9	14	191520	219428.571	230400	9	7	191520	219428.571	230400
10	16	167580	192000	201600	10	8	167580	192000	201600
11	18	148960	170666.667	179200	11	9	148960	170666.667	179200
12	20	134064	153600	161280	12	10	134064	153600	161280
13	24	111720	128000	134400	13	12	111720	128000	134400
14	28	95760	109714.286	115200	14	14	95760	109714.286	115200
15	32	83790	96000	100800	15	16	83790	96000	100800
16	36	74480	85333.333	89600	16	18	74480	85333.333	89600
17	40	67032	76800	80640	17	20	67032	76800	80640
18	48	55860	64000	67200	18	24	55860	64000	67200
19	56	47880	54857.1429	57600	19	28	47880	54857.1429	57600
20	64	41895	48000	50400	20	32	41895	48000	50400
21	72	37240	42666.667	44800	21	36	37240	42666.667	44800
22	80	33516	38400	40320	22	40	33516	38400	40320

Table 4-3. Mod Clock and Sample Rate Frequency Bins (continued)

MOD_CLK=3.072MHz Mode					MOD_CLK=1.536MHz Mode				
23	96	27930	32000	33600	23	48	27930	32000	33600
24	112	23940	27428.5714	28800	24	56	23940	27428.5714	28800
25	128	20947.5	24000	25200	25	64	20947.5	24000	25200
26	144	18620	21333.333	22400	26	72	18620	21333.333	22400
27	160	16758	19200	20160	27	80	16758	19200	20160
28	192	13965	16000	16800	28	96	13965	16000	16800
29	224	11970	13714.2857	14400	29	112	11970	13714.2857	14400
30	256	10473.75	12000	12600	30	128	10473.75	12000	12600
31	288	9310	10666.667	11200	31	144	9310	10666.667	11200
32	320	8379	9600	10080	32	160	8379	9600	10080
33	384	6982.5	8000	8400	33	192	6982.5	8000	8400
34	448	5985	6857.14286	7200	34	224	5985	6857.14286	7200
35	512	5236.875	6000	6300	35	256	5236.875	6000	6300
					36	288	4655	5333.333	5600
					37	320	4189.5	4800	5040
					38	384	3491.25	4000	4200
					39	448	2992.5	3428.5714	3600
					40	512	2618.438	3000	3150

4.4 Power Consumption Considerations

When selecting internal clocks, note the relationships between speed, power consumption, and processing cycles. As the internal clocks run faster, the power consumption increases. However, the amount of cycles per sample increases, allowing internal processing to increase such as using more biquad filters, more channels, or lower group delay decimation/interpolation filters. The internal clocks to pay attention to for these considerations include the input to the PLL, since PLL in integer mode uses less power than in fractional mode, the PLL multiplier creating CLK_SYS, and the relationship between MOD_CLK and DEM_CLK. By default, DEM_CLK will be 4x the MOD_CLK, but it can be forced to be 2x in the PWR_TUNE configuration registers. This will reduce the current consumption slightly, as shown in the application notes [TAC5x1x Power Consumption Matrix Across Various Usage Scenarios](#), [TAA52xx Power Consumption Matrix Across Various Usage Scenarios](#), and [TAD52xx Power Consumption Matrix Across Various Usage Scenarios](#). However, this is not recommended to be changed except in extreme power-saving configurations. The MOD_CLK has two speed options to select from, and the same considerations can be taken for this. A higher internal MOD_CLK is generally preferred, since this allows the most internal cycles per sample which is best for allowing maximal DSP processing, but sometimes the lower MOD_CLK rate is needed for especially low sample rates or for low power applications, or if the internal clocking math requires this lower rate. This math is explained in [Section 5](#).

5 Calculating PLL Multiplier and Dividers

Once the internal modulator clock (MOD_CLK) is selected with OSR and FS as described in [Section 4.3](#), the clock dividers and PLL multiplier can be calculated. In the clock tree (see [Figure 2-1](#)), there is a relationship between the input clock that gets multiplied by the PLL to create CLK_SYS which is the highest frequency internal clock, and the eventual outputs from the tree of DEM_CLK, MOD_CLK, BCLK, and FSYNC. Note that DEM_CLK, MOD_CLK, BOOST_CLK, and SAR_CLK are not actual outputs, but they are internal clocks that are used in operation of the device. Clocks that can be accessible by the user outside of the chip are PDM_CLK, which is related to the MOD_CLK, CLKOUT, and BCLK/FSYNC.

To calculate the PLL multiplier and CLK_SYS, we first select the input to the PLL: PASI BCLK, SASI BCLK, or CCLK. While OSC_CLK is an option, this is not recommended since it can vary due to temperature, voltage, and device. Then, make sure the input to the PDIV is 50MHz or below, and the input to the PLL block is 20MHz or below, as specified in [Table 3-3](#). In PLL integer mode (D=0000), PLL input can be from 256kHz-20MHz, and in fractional mode (D!=0000), PLL input is limited to 10MHz-20MHz. If the CCLK or BCLK that is the input to the PLL is too high, the PDIV divider can be used to divide the input down to an acceptable frequency for the input to the PLL.

The PLL multiplies the input by J.D, which is configured with page 3, registers 54-56. The output of the PLL must be an integer multiple of the MOD clocks, which is an integer multiple of the BCLK and therefore the sample rate. There are two main branches after the PLL multiplier to note in the clock tree which give these fundamental equations which we can use to determine J.D and the rest of the clock dividers:

$$\frac{\text{CLK_SYS}}{A} = \text{MOD_CLK} \quad (1)$$

$$\frac{\text{CLK_SYS}}{B} = \text{BCLK} \quad (2)$$

Calculating the CLK_SYS becomes the next step to determine all internal dividers. To do this, we can find the integer relationship between BCLK and MOD_CLK:

$$\frac{A}{B} = \frac{\text{BCLK}}{\text{MOD_CLK}} \quad (3)$$

Next, the clock that is the output of the PLL (also called CLK_SYS) can be calculated. CLK_SYS is limited to between 67-116MHz. In [Equation 4](#), X can be any integer that lets $67\text{MHz} \leq \text{CLK_SYS} \leq 116\text{MHz}$.

$$\frac{67\text{MHz}}{A \times \text{MOD_CLK}} \leq X \leq \frac{116\text{MHz}}{A \times \text{MOD_CLK}} \quad (4)$$

$$\text{CLK_SYS} = A \times \text{MOD_CLK} \times X \quad (5)$$

Finally, J.D can be calculated, finding the multiplier needed to transform the input to the PLL to the output CLK_SYS.

$$J.D = \frac{\text{CLK_SYS}}{\text{PLL_IN}} = \frac{\text{CLK_SYS}}{\text{PLL Clock Source}/P} = \frac{P \times \text{CLK_SYS}}{\text{PLL Clock Source}} \quad (6)$$

J is limited from 1-511, and D is limited from 0000-9999. D always has 4 digits even if not all decimal places are needed to represent the value, so for example a multiple of 6.4 is expressed as 6.4000.

Note that PLL Clock Source can be CCLK or BCLK of either ASI desired. CLK_SYS has an integer relationship with the MOD clock and BCLK but there can be several designs, so X can be chosen to allow for ideal J.D multipliers and later clock dividers. Similarly, PLL_IN can be optimized with different divisions of the P divider, but there is not always one unique design.

Remember if the PLL is operating in fractional mode, if D!=0, then the input range of frequencies is smaller, as shown in [Table 4](#) (for integer mode, PLL input can be 256kHz-20MHz, for fractional mode, PLL input can be 10MHz-20MHz). Keep this in mind when choosing X and the input divider PDIV.

Once J.D and the correct frequency of CLK_SYS are calculated, the other dividers can be chosen.

First, select NM_DIV_VAL to let the DEM_CLK be 4x the MOD_CLK:

$$\text{DEM_CLK} = \frac{\text{CLK_SYS}}{\text{NM_DIV_VAL}} \quad (7)$$

$$\text{NM_DIV_VAL} = \frac{\text{CLK_SYS}}{\text{MOD_CLK} \times 4} \quad (8)$$

NM_DIV_VAL has a range of 1-64. See [Table 3-3](#) for all divider value limitations.

CLK_SYS gets routed to MDIV and then MOD_CLKs as well as PDM_CLK and optionally BOOST_CLK and SAR_CLK. Select these dividers similarly to DEM clocks. Remember the limitations of these clocks, with the variance allowing for OSR changes, as shown in [Table 3-3](#).

$$\text{MOD_CLK} = \frac{\text{CLK_SYS}}{\text{MDIV} \times \text{MODCLK_DIV}} \quad (9)$$

$$\text{PDM_CLK} = \frac{\text{CLK_SYS}}{\text{MDIV} \times \text{PDM_DIV}} \quad (10)$$

$$\text{SAR_CLK} = \frac{\text{CLK_SYS}}{\text{MDIV} \times \text{SAR_DIV}} \quad (11)$$

$$\text{BOOST_CLK} = \frac{\text{CLK_SYS}}{\text{MDIV} \times \text{BOOST_DIV}} \quad (12)$$

An important note about the MDIV branch of the clock tree is that the clock that comes out of MDIV must be maximized with respect to the modulator clocks. So, the MODCLK_DIV must be maximized for best performance. Since the MODCLK_DIV is limited to 1, 2, or 4, any ratio of [Equation 13](#) that is divisible by 2 or 4 must put the 2 or 4 in the MODCLK_DIV, not leaving the MDIV to be the large multiple while the MODCLK_DIV=1.

$$\text{MDIV} \times \text{MODCLK_DIV} = \frac{\text{CLK_SYS}}{\text{MOD_CLK}} \quad (13)$$

CLK_SYS is also routed to NDIV, which can be selected as the input to PASI_BDIV, SASI_BDIV, and CLKOUT_DIV which lead to the outputs of BCLK, FSYNC, and CLKOUT. BDIV is used to generate BCLK when in controller mode and BCLK was not originally provided to the device. BDIV divider values can be calculated with the same variables used in calculating J.D, with X from [Equation 4](#) and B from calculating A/B in [Equation 3](#):

$$\text{BDIV} = \frac{\text{CLK_SYS}}{\text{B} \times \text{BCLK} \times \text{X}} \quad (14)$$

Then, if the device is in controller mode, the FSYNC is generated with one more divider called PASI_FSYNC_DIV, which can be 1-256. This is the FS to BCLK ratio set with P0_R56_D[5:0] and P0_R57_D[7:0]. Typically this is 32, 48, or 64 but the ratio can be any value 1-8192. Higher ratios are good for using many TDM channels, these devices support up to 256 channels of 32 bit audio. If in target mode, this divider must not be used and the FSYNC is provided externally.

$$\text{FSYNC} = \frac{\text{BCLK}}{\text{PASI_FSYNC_DIV}} \quad (15)$$

6 Example 1: Custom Target Mode, Non-Audio Rates

For example, what if the user wanted to use the TAC5112 in custom clocking mode with a sample rate of 20kHz and BCLK to FSYNC ratio of 16, with the BCLK and WCLK provided externally. This gives WCLK=20k, BCLK = 320kHz. They also want to internally generate PDM_CLK=800kHz. With this example, it is possible to configure all clocks in auto mode, but for maximum configurability with PDM_CLK and knowledge of the internal clocks, use custom clocking mode.

As seen in [Table 4-3](#), this sample rate can fall in bin 26 or 27, with an OSR of 144 or 160 respectively. To calculate potential MOD_CLK, multiply the sample rate by the OSR:

$$\text{MOD_CLK} = \text{FS} \times \text{OSR} \quad (16)$$

$$20\text{kHz} \times 144 = 2.88\text{MHz}; 20\text{kHz} \times 160 = 3.2\text{MHz} \quad (17)$$

Select the 3.2MHz (bin 27) option as our MOD_CLK since it is within 5% of the expected 3.072MHz (~4.17% higher than expected), while 2.88MHz (bin 26) is 6.25% lower than 3.072MHz.

Next, calculate J.D. Keep in mind that with a PLL_IN clock of only 320kHz, the PLL is in integer mode only. So, D must equal 0.

To calculate J and the other dividers:

$$\frac{A}{B} = \frac{\text{BCLK}}{\text{MOD_CLK}} = \frac{320\text{kHz}}{3.2\text{MHz}} = \frac{1}{10} \quad (18)$$

Remember CLK_SYS must be between 67 and 116MHz. So, the options for X are as follows:

$$\text{CLK_SYS} = A \times \text{MOD_CLK} \times X \quad (19)$$

$$67\text{MHz} \leq 1 \times 3.2\text{MHz} \times X \leq 116\text{MHz} \quad (20)$$

$$21 \leq X \leq 36 \quad (21)$$

Then, calculate J.D, noting the PLL_IN for this example is BCLK with PDIV=1:

$$J.D = \frac{\text{CLK_SYS}}{\text{PLL_IN}} = \frac{1 \times 3.2\text{MHz} \times X}{320\text{kHz}} = 10 \times X \quad (22)$$

Select X=32 because it is on the higher end of the range (higher CLK_SYS allows for better audio quality) and can easily be divided down in the next steps. So, J=320 and D=0, and CLK_SYS = 3.2MHz × 32 = 102.4MHz.

Next, select NM_DIV_VAL. Calculating NM DIV VAL:

$$\text{NM_DIV_VAL} = \frac{\text{CLK_SYS}}{\text{DEM_CLK}} = \frac{102.4\text{MHz}}{3.2\text{MHz} \times 4} = 8 \quad (23)$$

Remember DEM_CLK by default is 4 times MOD_CLK. This gives us NM_DIV_VAL=8.

Next, select an MDIV and the associated MOD_CLK, BOOST, SAR, and PDM CLKs. Since we know MOD_CLK is 3.2MHz, we can work backwards.

$$\text{MOD_CLK} \times \text{DIG_DAC_MODCLK_DIV} \times \text{MDIV} = \text{CLK_SYS} \quad (24)$$

$$3.2\text{MHz} \times \text{DIG_DAC_MODCLK_DIV} \times \text{MDIV} = 102.4\text{MHz} \quad (25)$$

$$\text{DIG_DAC_MODCLK_DIV} \times \text{MDIV} = 32 \quad (26)$$

As described in [Section 5](#), maximize DIG_DAC_MODCLK_DIV and minimize MDIV, which maximizes the clock coming out of MDIV. Since 32 is divisible by 4, set DIG_DAC_MODCLK_DIV to the maximum of 4 and MDIV to 8.

Next, select PDM_DIV for PDM_CLK=800kHz:

$$\text{PDM_CLK} = \frac{\text{CLK_SYS}}{\text{MDIV} \times \text{PDM_DIV}} = \frac{102.4\text{MHz}}{8 \times \text{PDM_DIV}} = 800 \text{ kHz} \quad (27)$$

$$\text{PDM_DIV} = \frac{102.4\text{MHz}}{800\text{kHz} \times 8} = 16 \quad (28)$$

Note that when ADC MOD_CLK and PDM_CLK are different, we need to enable page 0, register 0x77 bit 3 as dynamic power-up and power-down with different ADC MOD clock and PDM clock configuration, or there can be no PDM_CLK output.

Next, if in controller mode and required the BDIV for BCLK and FSYNCR dividers, the user can select an NDIV. However in target mode when BCLK and FSYNCR are provided, there is no need to configure NDIV, so this can be left at NDIV=1. CLKOUT is also on the NDIV branch of the clock tree, so if a specific other output clock is desired, this can be configured further. In this example, set CLKOUT to be CLK_SYS/8 for an output of 12.8MHz, so CLKOUT_DIV=8.

6.1 Example Script

```
# Target mode, 16 bit 1-channel TDM
# BCLK in=320kHz, WCLK in=20kHz, PDMCLK out=800kHz, CLKOUT = 12.8MHz
# CLKOUT on GPIO1, PDMCLK on GPIO2

w a0 00 00 #Set page 0
w a0 01 01 #Software Reset
w a0 02 09 #wake up with AVDD > 2v and all VDDIO level
w a0 0a b1 #GPIO1 = CLKOUT
w a0 0b 41 #GPIO2 = PDMCLK output
w a0 1a 00 #PASI TDM, 16 bit format
w a0 1e 20 #PASI DOUT Ch1 on TDM slot 0
w a0 28 20 #PASI DIN Ch1 on TDM slot 0
w a0 32 6f #FS bin=27, 5% tolerance, custom clock configuration
w a0 34 00 #PLL enabled, fraction mode disabled (D=0000), BCLK is input clock source
w a0 37 20 #PASI/SASI in target configuration (default setting)

w a0 00 03 #page 3
w a0 32 00 #(default) PLL_PDIV_IN is PASI BCLK, PASI BCLK divider source is PLL output
w a0 34 10 #(default) NM div input clock is PLL output
w a0 35 01 #(default) P=1
w a0 36 80 #J MSB=1
w a0 37 00 #D=0000
w a0 38 40 #J LSB=0100 0000 (J=320)
w a0 39 30 #NDIV=1, PDM_DIV=16
w a0 3a 22 #DIG_ADC_MODCLK_DIV=4, mdiv=8
w a0 3b 20 #DIG_DAC_MODCLK_DIV=4, DAC MOD clock 2x enabled (3.072 mode)
w a0 3e 08 #ANA_NM_DIV=8
w a0 44 07 #NDIV, MDIV, PDM_DIV enabled
w a0 45 f0 #MODCLK DIV enabled
w a0 46 05 #CLKOUT source = DSP clock (output of NDIV)
w a0 47 88 #CLKOUT divider enabled, value=8

w a0 00 00 #page 0
w a0 50 00 #(default) ADC Ch1 diff input, 5Kohm, 2Vrms AC-coupled, audio bandwidth
w a0 65 20 #(default) OUT1P LINEOUT, 0dB, DAC ch1 audio bandwidth
w a0 76 88 #enable input channel 1 and output channel 1
w a0 77 08 #different ADC MOD CLK and PDM CLK in dynamic power up/down
w a0 78 e0 #Power up all DAC+ADC channels
```

7 Example 2: Custom Controller Mode, Non-Audio Rates

For the next example, configure the device to be in controller mode, so it takes in an external CCLK and generates BCLK and FSYNC. Select a CCLK that has a non-integer relationship with the associated BCLK to take advantage of the configurable PLL, and this is a situation that can only be managed with the manual clock config mode. For this example, there are these clocks for the end system:

CCLK= 50MHz, WCLK= 98kHz, and BCLK-FS ratio of 32 (BCLK= 3.136MHz)

This sample rate can fall in bins 14 or 15 by the ranges defined in [Table 4-3](#). To select which OSR/bin to use, find which bin has the closest difference between the WCLK × OSR and 3.072MHz (or the 1.536MHz variant). Another way to do this is to check the difference in sample rates from the center frequency of the bin.

$$\% \text{ difference} = \frac{\text{WCLK} \times \text{OSR} - 3.072\text{MHz}}{3.072\text{MHz}} \times 100 \quad (29)$$

$$\text{Bin 14: } \frac{98\text{kHz} \times 28 - 3.072 \text{ MHz}}{3.072 \text{ MHz}} \times 100 = 10.68\% \quad (30)$$

$$\text{Bin 15: } \frac{98\text{kHz} \times 32 - 3.072\text{MHz}}{3.072\text{MHz}} \times 100 = 2.08\% \quad (31)$$

Select bin 15 with OSR=32 because of the smaller deviation of MOD_CLK from 3.072MHz. So, this gives MOD_CLK = 98kHz × 32 = 3.136MHz.

Next, calculate J.D:

$$\frac{A}{B} = \frac{\text{BCLK}}{\text{MOD_CLK}} = \frac{3.136\text{MHz}}{3.136\text{MHz}} = \frac{1}{1} \quad (32)$$

Calculate possible CLK_SYS values:

$$\text{CLK_SYS} = A \times \text{MOD_CLK} \times X = 3.136\text{MHz} \times X \quad (33)$$

$$67\text{MHz} \leq 3.136\text{MHz} \times X \leq 116\text{MHz} \quad (34)$$

$$22 \leq X \leq 36 \quad (35)$$

Remember that the maximum input to the PLL is 20MHz, so we need to use the PDIV or PLL pre-divider to divide down 50MHz to something in the range of 10-20MHz for fractional PLL. Select PDIV P=4 for PLL_IN of 12.5MHz (PLL Clock Source of 50MHz) and use this value to calculate J.D:

$$J.D = \frac{\text{CLK_SYS}}{\text{PLL_IN}} = \frac{\text{CLK_SYS} \times P}{\text{PLL Clock Source}} = \frac{3.136\text{MHz} \times 4 \times X}{50\text{MHz}} \quad (36)$$

Select X=30:

$$J.D = \frac{\text{CLK_SYS}}{\text{PLL_IN}} = \frac{3.136\text{MHz} \times 4 \times X}{50\text{MHz}} = \frac{3.136\text{MHz} \times 4 \times 30}{50\text{MHz}} = 7.5264 \quad (37)$$

So,

$$\text{CLK_SYS} = 3.136\text{MHz} \times 30 = 94.08\text{MHz} \quad (38)$$

From this CLK_SYS, calculate all other dividers needed.

$$\text{NM_DIV_VAL} = \frac{\text{CLK_SYS}}{\text{DEM_CLK}} = \frac{94.08\text{MHz}}{3.136\text{MHz} \times 4} = 7.5 \quad (39)$$

However, this divider needs to be an integer. So, transition to an approximate 1.536MHz setup for the MOD_CLK. With this noted, divide MOD_CLK by 2 in calculating all internal dividers:

$$NM_DIV_VAL = \frac{CLK_SYS}{DEM_CLK} = \frac{94.08MHz}{\frac{3.136MHz}{2} \times 4} = 15 \quad (40)$$

So, select NM_DIV_VAL=15.

Next, look at the MDIV branch of the clock tree to generate the MOD_CLK.

$$MOD_CLK \times DIG_DAC_MODCLK_DIV \times MDIV = CLK_SYS \quad (41)$$

$$DIG_DAC_MODCLK_DIV \times MDIV = \frac{CLK_SYS}{MOD_CLK} = \frac{94.08MHz}{\frac{3.136MHz}{2}} = 60 \quad (42)$$

As described before, TI recommends maximizing the clock coming out of MDIV, so a user can maximize DIG_DAC_MODCLK_DIV between the options of 1, 2, or 4. Since 60 is divisible by 4, select DIG_DAC_MODCLK_DIV=DIG_ADC_MODCLK_DIV=4, and MDIV=15.

Finally, look at the NDIV branch to generate BCLK and FSYNC.

$$NDIV \times PASI_BDIV = \frac{CLK_SYS}{BCLK} = \frac{94.08MHz}{3.136MHz} = 30 \quad (43)$$

Since NDIV is limited to 1-8, select NDIV=1 and PASI_BDIV=30. Then, since the BCLK-FSYNC ratio is 32, set PASI_FSYNC_DIV to 32.

$$FSYNC = \frac{BCLK}{PASI_FSYNC_DIV} = \frac{3.136MHz}{32} = 98kHz \quad (44)$$

7.1 Example Script

```

# Controller mode, 16 bit 2 channel TDM, wide bandwidth mode, MOD_CLK=1.5MHz mode
# CCLK in=50MHz, WCLK out=98kHz, BCLK out=3.136MHz
# CCLK on GPIO1

w a0 00 00 #Set page 0
w a0 01 01 #Software Reset
w a0 02 09 #wake up with AVDD > 2v and all VDDIO level
w a0 0a 10 #configure GPIO1 as input
w a0 0f 20 #Set GPIO1=CCLK
w a0 1a 00 #PASI TDM, 16 bit format
w a0 1e 20 #PASI DOUT Ch1 on TDM slot 0
w a0 1f 21 #PASI DOUT Ch2 on TDM slot 1
w a0 28 20 #PASI DIN ch1 on TDM slot 0
w a0 29 21 #PASI DIN ch2 on TDM slot 1
w a0 32 3f #FS bin=15, 5% tolerance, custom clock configuration
w a0 34 48 #PLL enabled, fraction mode enabled, fixed CCLK is input clock source
w a0 37 30 #PASI in controller configuration
w a0 38 80 #use internal BCLK for FSYNC generation
w a0 39 20 #LSB for pasi BCLK to FSYNC ratio = 32d
w a0 4e 80 #ADC MOD_CLK = 1.5mhz
w a0 4f 80 #DAC MOD_CLK = 1.5mhz

w a0 00 03 #page 3
w a0 32 80 #PLL_PDIV_IN_CLK is CCLK, and PASI BCLK divider source is PLL output
w a0 34 10 #(default) NM div input clock is PLL output
w a0 35 04 #PDIV=4
w a0 36 14 #J MSB=0, D MSB=01 0100 (D=5264)
w a0 37 90 #D LSB= 1001 0000
w a0 38 07 #J LSB=00000111 (J=7)
w a0 39 20 #NDIV=1, PDM_DIV=1
w a0 3a 3e #MDIV=15, DIG_ADC_MODCLK_DIV=4
w a0 3b 28 #DIG_DAC_MODCLK_DIV=4, DAC MOD_CLK 2x disabled (1.536 mode)
w a0 3c 1e #PASI BCLK divider=30
w a0 3e 0f #ANA_NM_DIV=15
w a0 44 07 #NDIV, MDIV, and PDMDIV enabled
w a0 45 fa #MODCLK, PASI BDIV, and PASI FSYNC DIV enabled

w a0 00 00 #page 0
w a0 50 01 #ADC Ch1 diff input, 5Kohm, 2Vrms AC-coupled, wide band mode (over 96k)
w a0 55 01 #ADC Ch2 diff input, 5Kohm, 2Vrms ac-coupled, wide band mode (over 96k)
w a0 65 21 #OUT1P LINEOUT 0dB, DAC ch1 wide band mode (over 96k)
w a0 6c 21 #OUT2P LINEOUT 0dB, DAC ch2 wide band mode (over 96k)
w a0 76 cc #enable input and output channels 1 and 2
w a0 78 e0 #Power up all DAC+ADC channels
  
```

8 Summary

The clocking system of the TAx5x1x family is extremely customizable, and users can calculate the dividers required to use the device in non-standard clocking configurations. Manual PLL configuration can be useful for controller mode in non-standard CCLK or WCLK rates, or to generate specific outputs for PDM clock, BCLK, WCLK, or a general CLKOUT.

9 References

1. Texas Instruments, [TAC5212 High-performance stereo audio codec with 119dB dynamic range ADC and 120dB dynamic range DAC](#), datasheet.
2. Texas Instruments, [Clocking Configuration of Device and Flexible Clocking For TAx5x1x Family](#), application note.
3. Texas Instruments, [Clock Error Configuration, Detection, and Modes Supported in TAx5x1x Family](#), application note.
4. Texas Instruments, [TAX5X1X Synchronous Sample Rate Conversion](#), application note.
5. Texas Instruments, [TAC5XXX-Q1 Power Consumption Matrix Across Various Usage Scenarios](#), application note.
6. Texas Instruments, [TAA52xx Power Consumption Matrix Across Various Usage Scenarios](#), application note.
7. Texas Instruments, [TAD52xx Power Consumption Matrix Across Various Usage Scenarios](#), application note.

IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATASHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES "AS IS" AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS AND IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for skilled developers designing with TI products. You are solely responsible for (1) selecting the appropriate TI products for your application, (2) designing, validating and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, regulatory or other requirements.

These resources are subject to change without notice. TI grants you permission to use these resources only for development of an application that uses the TI products described in the resource. Other reproduction and display of these resources is prohibited. No license is granted to any other TI intellectual property right or to any third party intellectual property right. TI disclaims responsibility for, and you fully indemnify TI and its representatives against any claims, damages, costs, losses, and liabilities arising out of your use of these resources.

TI's products are provided subject to [TI's Terms of Sale](#), [TI's General Quality Guidelines](#), or other applicable terms available either on [ti.com](#) or provided in conjunction with such TI products. TI's provision of these resources does not expand or otherwise alter TI's applicable warranties or warranty disclaimers for TI products. Unless TI explicitly designates a product as custom or customer-specified, TI products are standard, catalog, general purpose devices.

TI objects to and rejects any additional or different terms you may propose.

Copyright © 2026, Texas Instruments Incorporated

Last updated 10/2025