Application Report
SLEA002 – June 2001

TLC320AD77C Clock and Timing Issues

Jim Coates *Digital Audio Solutions / Digital Audio Video*

ABSTRACT

The Texas Instruments TLC320AD77C stereo audio codec requires a 3-ns separation in edges between MCLK and SCLK and an LRCLK phase stability of ±10 MCLKs over the duration that power is applied. This paper examines the source of these requirements and the impact they have on a design when using the codec with the Texas Instruments TMS320C54x™ DSP and the Texas Instruments TUSB3200 USB controller.

Contents

Figures

Introduction

The timing requirements on the TLC320AD77C stereo audio codec clock lines—MCLK, SCLK, and LRCLK—must be taken into account when incorporating this part into a design. This application note discusses the source of the timing requirements placed on the TLC320AD77C codec clocks and the implications these requirements have when using a Texas Instruments TUSB3200 USB controller and a Texas Instruments TMS320C54x™ DSP with the TLC320AD77C codec.

First, a discussion of these timing requirements is presented, followed by a discussion of the design implications imposed by these requirements. This discussion is followed by the presentation of three different implementations using these three components. The pros and cons of each implementation are discussed.

Timing Issues

Figure 1 depicts the first architecture to be examined. The clock and timing issues addressed in this application note only involve the TLC320AD77C stereo audio codec, the C54x™ DSP, and the TUSB3200 USB controller. For this reason, only these three components are represented in Figure 1. Note that the block diagrams shown in Figure 1 are only intended to represent functionality and not how the functions are actually implemented.

TUSB3200 USB Controller Issues

The TUSB3200 USB controller is tasked with inputting, retrieving, and buffering a USB audio stream and outputting the buffered result to the C54x DSP via an audio codec ’97 (AC’97) link. The TUSB3200 USB controller derives its internal clocks from an external 6-MHz crystal source. The 6-MHz source is up-converted to 48 MHz by an on-chip phase-locked loop (PLL) and presented to an on-chip frequency synthesizer. (The PLL also generates lower frequency clocks for other on-chip uses.)

Since the incoming audio data stream is a continuous data stream, synchronization must be maintained between the incoming data rate of the USB channel and the outgoing data rate on the AC’97 channel. If this is not done, the on-chip buffer either overflows—AC’97 output rate slower than the USB incoming rate—or underflows—AC’97 rate faster than the USB incoming rate. In either case, an audio discontinuity, which is both audible and unpleasant, results.

For the TUSB3200 USB controller, the means of achieving synchronization between the incoming and outgoing data rates is via a *soft PLL*, or a software PLL (instead of a second hardware PLL that acquires lock via the minimal number of transitions in the USB data stream). The USB input consists of digital data at a bit rate of 12 MHz. This data is layered onto packets, which in turn are layered onto 1-ms frames. The AC’97 output rate, on the other hand, is 12.288 MHz and is layered onto 256-bit frames that occur at a 48-kHz rate. Synchronization is achieved by requiring that the output of the frequency synthesizer have exactly 12.288 MHz / 10^3 = 12,288 clock periods every USB frame. The number of 12.288-MHz clock events per USB frame is monitored via a *capture* counter. The count from the capture counter is input into an on-chip 8052 μcontroller, which uses the count to adjust the frequency output by the frequency synthesizer.

The USB 1-ms frame rate is derived from a different clock source than is the 6-MHz crystal referenced 12.288-MHz AC’97 output bit rate. It is highly unlikely then that a single setting of the frequency synthesizer is sufficient to maintain synchronization. Instead, more likely, the 8052 firmware continuously adjusts the frequency synthesizer to arrive at an average of 12,288 AC’97 clock events per USB frame.

The clock delivered to the C54x DSP—24.576-MHz MCLKO—is, in all likelihood, continuously adjusted in frequency. The deviation and the rate of the deviation are dependent on the relative accuracy of the USB host clock, the 6-MHz clock, and the performance of the soft PLL.

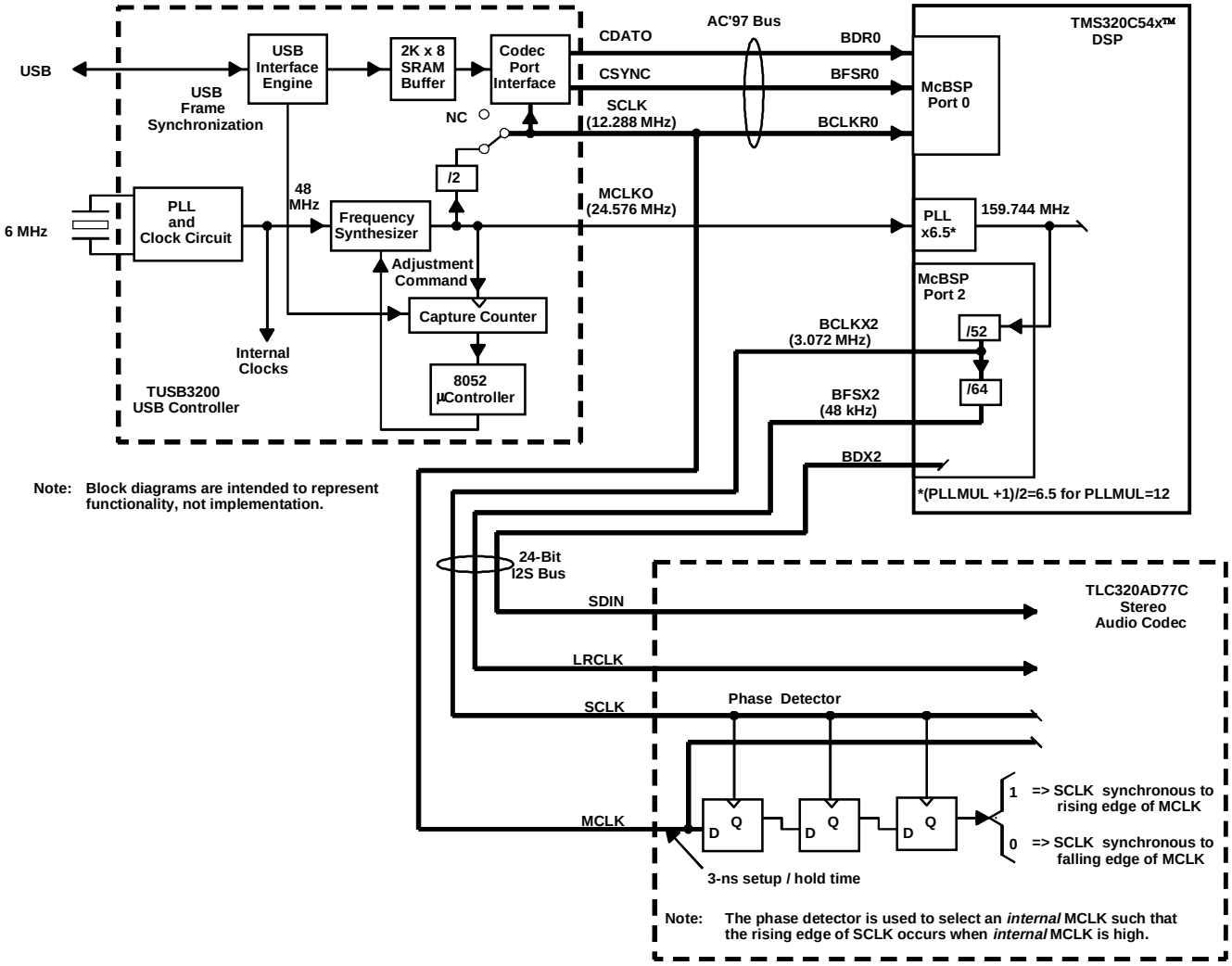

Figure 1. Clock Management Implementation No. 1

TMS320C54x DSP Issues

The C54x DSP receives the AC'97-converted USB audio stream, processes it as required, and outputs the resultant audio to codecs for conversion to the analog domain. As seen from Figure 1, two of the three McBSP ports on the C54x DSP route the audio. McBSP Port 0 receives the AC'97 data stream from the TUSB3200 USB controller, and McBSP Port 2 outputs the processed audio to the TLC320AD77C codec. Audio reception is accomplished using the clocking provided by the TUSB3200 USB controller (CSYNC and SCLK), while the clocking used to output the processed audio to the TLC320AD77C codec is generated within the DSP.

The clock source for the DSP is the TUSB3200-provided MCLKO. As previously discussed, MCLKO cannot be counted on as being a constant stable clock. When the TUSB3200 USB controller frequency synthesizer adjusts the frequency of MCLKO, it requires a readjustment of phase lock in the DSP PLL. During this relock period, the phasing of SCLK and LRCLK output to the TLC320AD77C codec cannot be specified with respect to MCLKO. More significantly, the phasing of SCLK and LRCLK to the TLC320AD77C codec cannot be specified with respect to MCLK, which is the third clock required by the TLC320AD77C codec.

The phase uncertainty between MCLK and SCLK at the TLC320AD77C codec is the primary timing issue.

TLC320AD77C Codec Issues

A key concern in designing codecs is the management of digital noise and ground bounce to minimize the noise in the analog domain of the chip. To isolate the digital noise from the analog circuitry, the bit period of the incoming data (SCLK period) is sectioned into multiple time slots. These time slots are used to control when activities take place, such as registering the incoming data stream, updating the hold capacitor in the D/A converter, etc. In this way, analog transactions do not take place when digital transactions are happening and the digital noise and ground ringing from the digital transactions decline to acceptable levels before the analog transactions happen. The sectioning is accomplished by subdividing each SCLK period such that analog switching events do not occur coincident with digital switching events and thus can be isolated from the noise generated by the digital switching events. This sectioning is accomplished by sampling the phase of MCLK at the occurrence of a rising edge of SCLK (see Figure 1). With the knowledge of the phase of MCLK relative to the rising edge of SCLK, the subdivision of the SCLK period into multiple MCLK periods can be done in such a manner as to maximize the separation of analog and digital events within the SCLK period.

However, the TLC320AD77C codec does allow for drift in SCLK and hence LRCLK, relative to MCLK. The TLC320AD77C specification states that if the LRCLK phase changes more that ± 10 MCLKs, the device automatically resets. (This phase change is an accumulated phase change over all time, and not during one LRCLK period.) The knowledge of the phase of MCLK relative to the rising edge of SCLK then is only used during device initialization; this knowledge allows the accommodation of up to a 10-MCLK slip in the phase of LRCLK.

The TLC320AD77C codec makes no assumptions as to the phase of MCLK with respect to SCLK. However, the two clocks must be synchronous (derived from the same base clock), and either edge of MCLK must be separated from the rising edge of SCLK by at least 3 ns. The 3-ns requirement prevents metastability when sampling the phase of MCLK with the rising edge of SCLK. If the sampling of the MCLK fails to yield the right answer because of a metastability event, the ±10-MCLK tolerance in the drift on LRCLK is reduced to ±9.5 MCLKs. This is certainly not a hard failure, but does result in the part being out of spec.

The 3-ns requirement and the ±10-MCLK tolerance in the drift on LRCLK are the problems associated with the implementation of Figure 1. SCLK and LRCLK are derived from the output of the C54x DSP PLL. The characteristics of this PLL are not published, and thus the phase relationship between MCLK and SCLK cannot be assumed to be constant over many releases of a given DSP type. However, the phase relationship, whatever it is, remains fixed as long as the clock into the C54x DSP PLL is stable. However, it has already been noted that the clock into the C54x DSP PLL is not a stable clock, but is continuously adjusted in frequency so that an average of 12,288 AC'97 clock events per USB frame are maintained. These adjustments in frequency result in transitional behavior in the C54x DSP PLL. (And the transitory effects are even heightened during power on, which is exactly when the 3-ns restriction comes into play.)

As a result of the transitory behavior of the C54x DSP PLL at power on and in the presence of frequency adjustments to MCLKO, the phase relationship of SCLK relative to MCLK is not only unknown, but is transitory. As implemented, there can be no assurance that the ±10 MCLK tolerance in the drift on LRCLK can be maintained over time. Failure to maintain this phase relationship resets the TLC320AD77C codec, and resetting the codec creates an audible discontinuity in the processing of the USB audio data stream.

Implementation No. 2

The solution to the problems uncovered in the implementation in Figure 1 is to stabilize the phase relationship of SCLK and LRCLK with respect to MCLK. Implementation No. 2, shown in Figure 2, is an implementation which does just that.

The only difference between the implementation of Figure 2 and the implementation of Figure 1 is the source of the bit clock signal SCLK between the TUSB3200 USB controller and the C54x DSP. In Figure 2 the source of SCLK is the C54x DSP, whereas in Figure 1, the source of SCLK is the TUSB3200 USB controller. The problems of the implementation in Figure 1 are now eliminated, because the same base clock (the C54x DSP PLL output) generates MCLK, SCLK, and LRCLK to the TLC320AD77C codec. These clocks exhibit excursions in frequency resulting from the TUSB3200 USB controller frequency synthesizer efforts to arrive at an *average* of 12,288 AC'97 clock events per USB frame. Because all 3 clocks are derived from the same source, these excursions are tracked equally by the 3 clocks, thus preventing an accumulation of phase slippage between the clocks such that the ±10-MCLK tolerance in the drift of LRCLK is exceeded.

Moreover, the implementation in Figure 2 allows the phase relationship between MCLK and SCLK into the TLC320AD77C codec to be quantified and assured. In examining Figure 2, the TLC320AD77C codec MCLK originates from McBSP Port 0, whereas the TLC320AD77C codec SCLK and LRCLK originate from McBSP Port 2. When the C54x DSP is powered up, the dividers that govern the frequencies of SCLK and MCLK are set to one. This means that at reset MCLK and SCLK are the same frequency and equal to the CPU clock / 2. Following reset, the default frequency remains at the CPU clock / 2 until the McBSP port is initialized via firmware. As soon as the firmware sets the pertinent parameters in the McBSP port and releases the McBSP port (by resetting the reset flag GRESET in the McBSP control register), the divide counter begins counting and the programmed output frequency immediately becomes active. Because MCLK and SCLK originate from different McBSPs, the releasing of these two ports cannot occur at the same time and the activation of the clocks output by the two ports is separated by an integer number of CPU clocks. Because MCLK and SCLK are integer-related, the integer number of CPU clocks that separate the phasing of MCLK and SCLK remains fixed in time. Because at least one CPU clock period must separate the release times, and because the maximum CPU frequency of the C54x DSP is currently 160 MHz, the minimum separation between the edges of MCLK and SCLK is 1 / 160 MHz = 6.25 ns, which is well above the 3-ns requirement. (Note that the time between releasing the 2 McBSP ports can be increased by adding additional software code between the 2 events. It is possible that the number of instructions added could result in the number of CPU clocks separating the phasing between SCLK and MCLK becoming equal to an integer number of SCLK periods. If this happens, the 3-ns requirement cannot be ensured. It is necessary then that firmware designs take this possibility into account and ensure that it does not happen.)

Another important issue is what happens during power-on initialization. It has already been stated that the TLC320AD77C codec samples the phase relationship between SCLK and MCLK only once during its power-on initialization procedure. But this sampling does not take place until the TLC320AD77C codec is removed from the reset condition. The TLC320AD77C codec enters a reset mode asynchronously upon the occurrence of an external reset pulse, but remains in reset until the external reset is deactivated and a rising edge of SCLK occurs. Because the C54x DSP controls the external reset to the TLC320AD77C codec via a host port interface (HPI) output, the following sequencing must be followed by the C54x DSP to ensure that the 3-ns timing constraint is met. First, the C54x DSP deactivates the reset line to the TLC320AD77C codec by driving the associated HD GPIO port to 1. The C54x DSP then activates McBSP Port 0 and McBSP Port 2, in that order. With this ordering of events, MCLK is present at the TLC320AD77C codec when the TLC320AD77C codec sees the first edge of SCLK, thereby ensuring that the 3-ns requirement is satisfied.

The implementation of Figure 2 eliminates the possibility of interrupts in the processing of the incoming audio stream due to clock phasing issues. However, as is discussed in the following section, the implementation of Figure 2 can compromise DSP throughput.

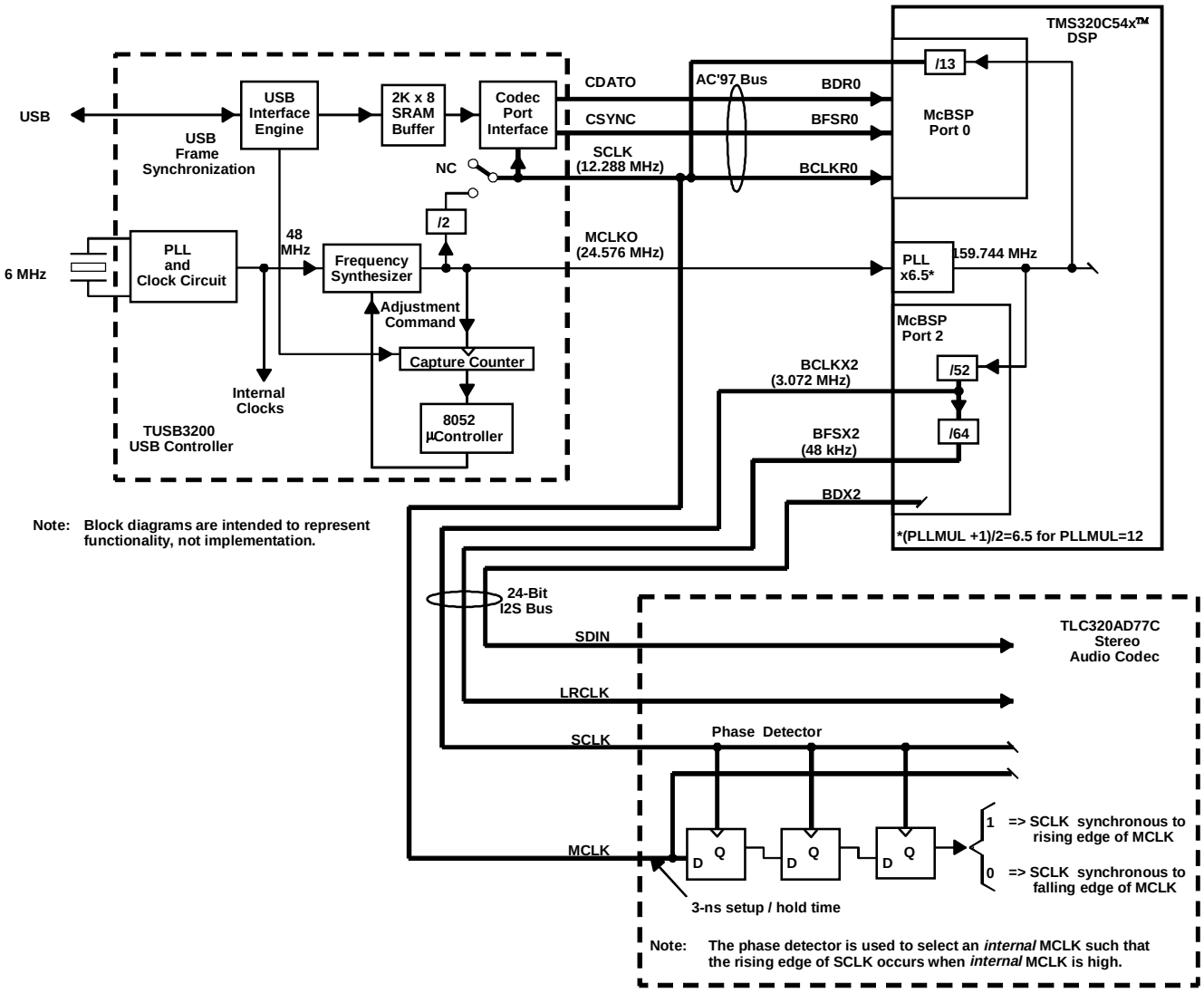

Figure 2. Clock Management Implementation No. 2

Implementation No. 3

Figure 3 is a third implementation that resolves all the issues with the implementations of Figures 1 and 2. The most serious problem with the implementation of Figure 2 is that there is the potential that DSP processing throughput can be compromised. The later releases of the C54x DSP family provide a software-configurable PLL that allows the CPU clock to be set to 1 of 31 possible multiples of the PLL input clock frequency over a range of ((0.5 to 15) ÷ N) times the PLL input clock, where N = 1 or 2. For a PLL input clock of 24.576 MHz, a multiplication factor of 6.5 yields a DSP CPU clock of 159.744 MHz, which is right at the current 160 MHz limit for the TMS320VC5416 DSP. But consider the TMS320VC5402 DSP, which is limited to a maximum CPU clock rate of 100 MHz. A PLL input clock of 24.576 MHz can only yield an optimum CPU clock rate of 98.304 MHz (PLL x 4), which is a processing loss of almost 2 MIPs. As another example, consider a future device with a maximum CPU clock frequency of 200 MHz. For this case, a PLL input clock of 24.576 MHz can only yield an optimum CPU clock rate of 196.608 MHz (PLL x 8), which is a processing loss of 3.4 MIPs.

The implementation of Figure 3 resolves this DSP processing throughput limitation by allowing a separate clock source to supply the input clock to the DSP PLL. (In Figure 3, a 20-MHz external clock is shown driving the PLL input pin, which allows the 160-MHz maximum for the internal CPU clock to be achieved). The implementation of Figure 3 can only be realized today with a TMS320VC5416 DSP, as the implementation relies on the ability of the TMS320VC5416 DSP McBSP ports to use a base clock other than the CPU clock. An external clock input pin specifically dedicated to driving the McBSP ports with an external source is not provided on any of the current generation C54x DSPs. However, the TMS320VC5416 DSP does allow BCLKR (the receive port bit clock) or BCLKX (the transmit port bit clock) to be used as the source for the base clock used in the McBSP port in question. (Note that each McBSP port provides the option for BCLKR and/or BCLKX to be supplied externally). As seen in Figure 2, McBSP Port 2 is provided an external clock via its BCLKR2. The external clock received is the 12.288-MHz SCLK from the TUSB3200 USB controller (which is also routed to the McBSP Port 0 BCLKR0 pin). McBSP Port 2 then divides BCLKR2 by 4 to yield SCLK to the TLC320AD77C codec and then by another 64 to yield LRCLK to the TLC320AD77C codec.

The TLC320AD77C MCLK thus serves as the base clock for the generation of SCLK and LRCLK; thus, synchronization among the three clocks is ensured. The 3-ns issue is also resolved because of the processing that takes place on the clock that enters McBSP Port 2 via the BCLKR2 pin. First the clock must pass through a buffer. The buffered clock then must route through selection logic and then clock the counters generating BCLKX2 and BFSX2. BCLKX2 and BFSX2 thus incur a clock-to-Q propagation delay. Lastly, BCLKX2 and BFSX2 must route through output buffers. The sum of the propagation delays through the I/O buffers and selection logic and the clock-to-Q propagation delay ensures that SCLK and LRCLK sufficiently lag MCLK at the TLC320AD77C codec to satisfy the 3-ns requirement. However, it is important to note that the order of McBSP port activation during a power-on initialization procedure must adhere to the order presented above in the discussion of Implementation Solution #2.

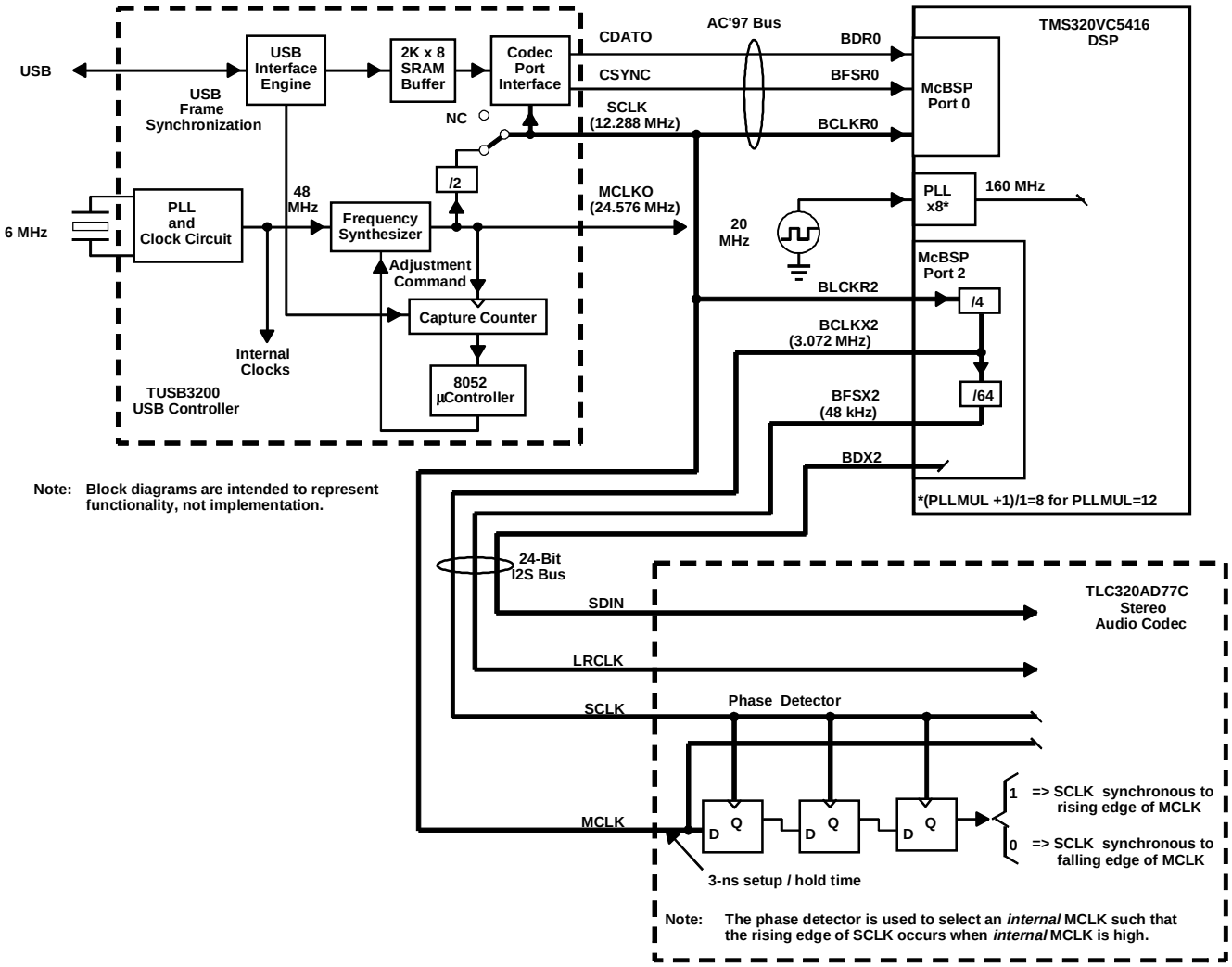

Figure 3. Clock Management Implementation No. 3

The implementation of Figure 3 resolves all issues with the implementation of Figure 1 and has the added benefit of eliminating a potential penalty in DSP processing throughput by allowing a separate clock to be the input clock to the DSP PLL. The implementation of Figure 3, however, limits the DSP selection to the TMS320VC5416 DSP.

Conclusion

The timing requirements on the TLC320AD77C stereo audio codec clock lines—MCLK, SCLK, and LRCLK—can best be served by using the TMS320VC5416 DSP. The separate McBSP clock input of the TMS320VC5416 DSP allows LRCLK and SCLK to be directly derived from MCLK, while, at the same time, driving the TMS320VC5416 DSP PLL with a clock source set to maximize throughput.

References

1. *TMS320VC5416 Fixed-Point Digital Signal Processor*, SPRS095G, Revised January 2001
2. *TLC320AD77C 24-Bit 96 kHz Stereo Audio Codec*, SLAS194, August 1999
3. *TUSB3200 Data Manual - USB Streaming Controller (STC),* SLAS240, October 1999

Trademarks

TMS320C54x and C54x are trademarks of Texas Instruments Incorporated.

IMPORTANT NOTICE

Texas Instruments and its subsidiaries (TI) reserve the right to make changes to their products or to discontinue any product or service without notice, and advise customers to obtain the latest version of relevant information to verify, before placing orders, that information being relied on is current and complete. All products are sold subject to the terms and conditions of sale supplied at the time of order acknowledgment, including those pertaining to warranty, patent infringement, and limitation of liability.

TI warrants performance of its products to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are utilized to the extent TI deems necessary to support this warranty. Specific testing of all parameters of each device is not necessarily performed, except those mandated by government requirements.

Customers are responsible for their applications using TI components.

In order to minimize risks associated with the customer's applications, adequate design and operating safeguards must be provided by the customer to minimize inherent or procedural hazards.

TI assumes no liability for applications assistance or customer product design. TI does not warrant or represent that any license, either express or implied, is granted under any patent right, copyright, mask work right, or other intellectual property right of TI covering or relating to any combination, machine, or process in which such products or services might be or are used. TI's publication of information regarding any third party's products or services does not constitute TI's approval, license, warranty or endorsement thereof.

Reproduction of information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations and notices. Representation or reproduction of this information with alteration voids all warranties provided for an associated TI product or service, is an unfair and deceptive business practice, and TI is not responsible nor liable for any such use.

Resale of TI's products or services with *statements different from or beyond the parameters* stated by TI for that product or service voids all express and any implied warranties for the associated TI product or service, is an unfair and deceptive business practice, and TI is not responsible nor liable for any such use.

Also see: Standard Terms and Conditions of Sale for Semiconductor Products. www.ti.com/sc/docs/stdterms.htm

Mailing Address:

Texas Instruments
Post Office Box 655303
Dallas, Texas 75265