

Hardware auto-identification and software auto-configuration for the TLV320AIC10 DSP Codec—a “plug-and-play” algorithm

By Wendy X. Fang, *Application Specialist, Advanced Analog Products,* and Perry Miller, *Application Specialist, Advanced Analog Products*

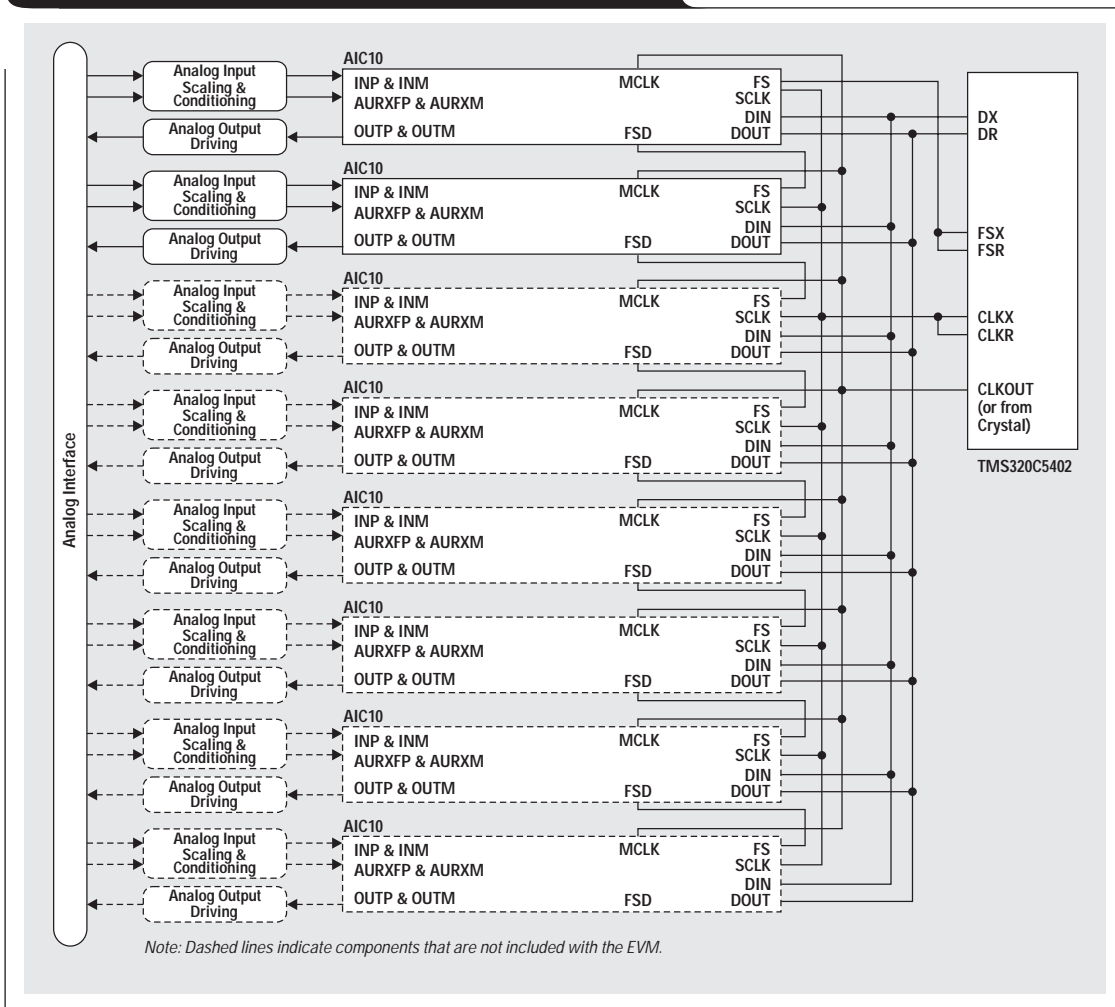
Introduction

The analog interface circuit (AIC), also called a modem Codec, is a complete data acquisition system on a chip for general-purpose telephony/speech applications such as: modem analog interface, voice-band audio processing, noise cancellation/suppression, hands-free communication, security voice systems, tone generation, echo cancellation, voice-over-internet protocol (VoIP), and industrial process control. The AIC is normally interfaced to a digital signal processor (DSP). This approach has a number of

advantages and cost-saving benefits that cannot be realized with traditional Codec analog interfacing. The benefits include:

- lower cost,
- simple interface design and fewer components,
- greater system reliability due to fewer components,
- lower power consumption,
- higher performance, and
- high programmability.

Figure 1. Block diagram of the TLV320AIC10/11 EVM



The TLV320AIC10 is a high-resolution, high-speed device that contains both ADC/DAC data paths. The ADC data path includes signal conditioning op amps, a multiplexer, an anti-aliasing filter, a programmable gain amplifier, a Sigma-Delta ADC, and a decimation filter. The DAC data path is composed of a glueless DSP interface to an interpolation filter, a Sigma-Delta DAC, a low-pass reconstruction filter, a programmable gain amplifier, and a transmit amplifier.

An internal band-gap voltage reference is used to provide a stable V_{REF} both to the ADC and the DAC. The chip contains a clock divider circuit for dividing down the MCLK from the DSP or other clock source. Two analog inputs can be multiplexed to the ADC. In practice, more than one 'AIC10 is connected in parallel to form the multiple inputs/multiple outputs solution required for today's real-world telecommunication applications. Up to 8 TLV320AIC10 devices can be paralleled, in a cascade format, to communicate with a single DSP.

Figure 1 shows the block diagram of the TLV320AIC10/11 evaluation module (EVM), where a possible maximum of 16 analog inputs and 8 analog outputs could interface with a TMS320C54x DSP through 'AIC10s, providing a practical platform for various voice-band communication applications. There are many different working structures while multiple 'AIC10s work in parallel. In the one most widely applied, an 'AIC10 is assigned as the master and the rest of the 'AIC10 devices are assigned as slaves. Detailed information and various options are outlined in Reference 1.

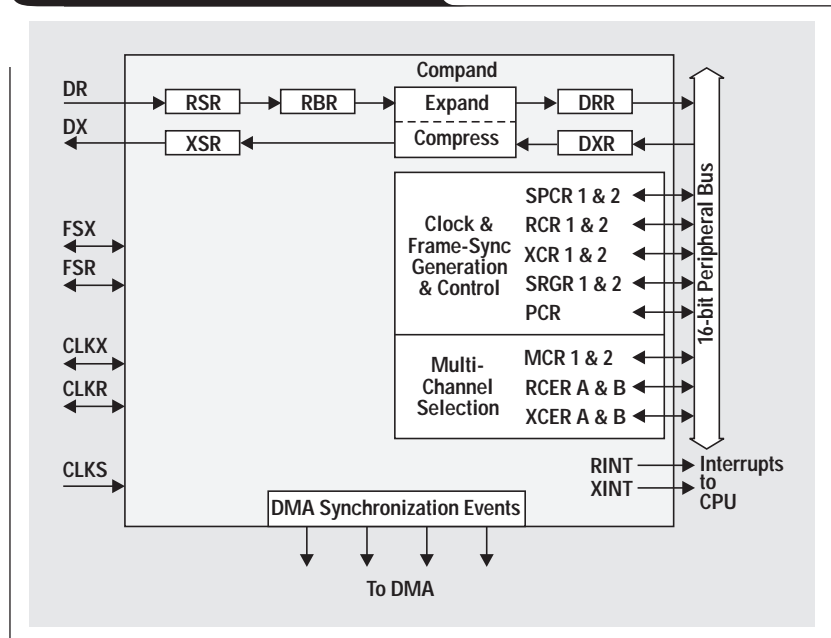
The software program used to initialize and configure the AICs depends very much on the individual device hardware configuration and often needs to be modified for every hardware change or reconfiguration. Besides, human hardware configuration mistakes also can occur and affect the working reliability of the system.

In this article we present an algorithm that automatically identifies the number of on-board functional 'AIC10 devices and their positions (as a master or a slave) and indicates hardware configuring errors if they exist. Consequently, all control registers for the master and slave 'AIC10 devices are programmed automatically. With this algorithm, a faulty master AIC will *not* jeopardize the functioning of the whole system; and hardware reconfiguration or AIC master/slave reassignment will *not* require any software changes or rewriting. This "plug-and-play" software algorithm supports the TLV320AIC10/11 EVM for various customized applications, improving the DSP/'AIC10 system's reliability, flexibility, software reusability, and troubleshooting options.

McBSP and 'AIC10 interface

The algorithm developed in this article can be considered a generic "plug-and-play" communication method that handles the initialization and interface between a multi-channel buffered serial port (McBSP) and a bank of

Figure 2. McBSP block diagram



TLV320AIC10 devices. Before the auto-identification and auto-configuration method is presented, a review of the McBSP and 'AIC10 interface is vital.

The multichannel buffered serial port

The McBSP is the full-duplex, multichannel buffered serial port in the TMS320C54x and C6x DSP families that allows direct interface among DSPs and between a DSP and other devices in a system, such as 'AIC10s.

A McBSP consists of 7 pins used to interface an external device (as shown in Figure 2):

Communication Data Lines

Data IN (from 'AIC10 to DSP): DR (data receive)
Data OUT (from DSP to 'AIC10): DX (data transmit)

Communication Clock and Control

DR Shift Clock: CLKR
DX Shift Clock: CLKX
DR Frame Sync: FSR
DX Frame Sync: FSX

McBSP System Clock

System Clock: CLKS

A 16-bit external peripheral bus forms a bridge between the DSP's CPU and the McBSP. The triple-buffered DR and double-buffered DX can be read/written by the CPU. The CPU also reads the McBSP's status bits located in the serial port control registers, which are part of the sub-address register. See Reference 2 for more details.

'AIC10

The TLV320AIC10 is a general-purpose, 3- to 5.5-V, 16-bit, 22-kps DSP Codec, mounted in a 48-pin surface-mount TFQP. The 'AIC10 contains signal conditioning op amps, filtering on both input and output, DSP interface to receive or transmit digital data to and from an external

Continued on next page

Continued from previous page

host processor, programmable gain amplifier, an ADC, and a DAC. Figure 3 shows a simplified 'AIC10 block diagram.

The interface to the analog interface pins is defined as follows:

Analog Input #1

Positive IN: INP
Negative IN: INM

Analog Input #2

Positive IN: AURXFP or AURXCP
(if receive-path amplifier is used)
Negative IN: AURXM

Analog Output

Positive OUT: OUTP or DTXOP
(if transmit-path amplifiers are used)
Negative OUT: OUTM or DTXOM
(if transmit-path amplifiers are used)

There are two digital pins on an 'AIC10 that can be considered as a pair of digital input/output ports:

Digital IN: ALTIN
Digital OUT: FLAG

For methods of interconnecting these analog and digital interface signals, please see References 1 and 3. This article concentrates on the software interface between DSP and 'AIC10. There are 11 pins provided on an 'AIC10 device to communicate with the C54x and C6x DSPs:

Main Communication Data Lines

Data IN (from DSP to 'AIC10): DIN
Data OUT (from 'AIC10 to DSP): DOUT

Communication Clock and Control

Data Shift Clock: SCLK
Frame Sync: FS
Frame Sync Delay: FSD (output to next slave AIC as its FS)

Control Register Configuring

Data IN: DCSI (direct configuration serial input)
Hardware Secondary Comm Request: FC (force configuration)

Communication Mode Setups

(configured by a jumper; will *not* change after power-up)

Frame Sync (FS) Format: M0
M1
Master/Slave Mode: M/S

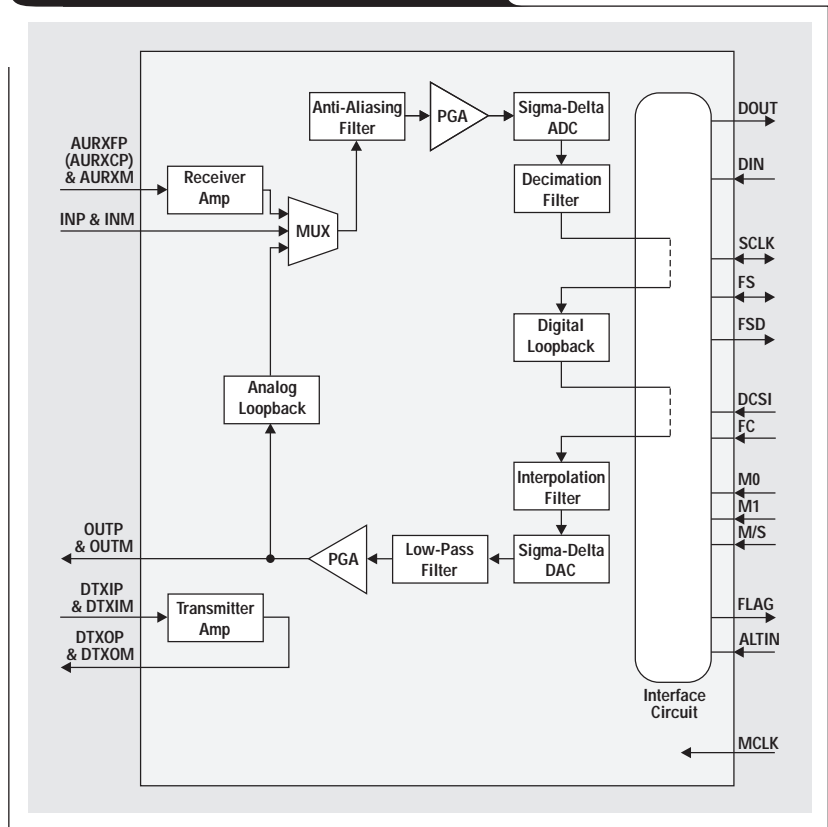
AIC Device Master Clock

Master Clock: MCLK

Each AIC device's major link to the system DSP consists of 4 lines connected to its DIN, DOUT, SCLK, and FS pins, which are the hardware interface to the SPI port or the McBSP of a DSP.

Prior to power-up of the 'AIC10s, the communication mode is established by hardware configuration and usually involves setting a number of jumpers on the 'AIC10 EVM.

Figure 3. TLV320AIC10 block diagram



Interface

Figure 1 shows the hardware connection between the McBSP and the 'AIC10s. The MCLK at each 'AIC10 is connected to the same master clock source, such as a crystal or the CLKOUT pin of the system DSP. Each of the AICs is cascaded from its FSD pin to the next slave device FS pin. The master 'AIC10 generates SCLK that goes to the SCLK pins of all other 'AIC10s.

The McBSP and 'AIC10 communication data formats at primary (or ADC/DAC data) frame and secondary (or 'AIC10 control register configuring) frame are given in Figure 4. The primary communication always occurs, but the secondary one happens only if bit 0 is set in DIN (15-bit data mode) or by hardware pin FC in the 'AIC10 device.

The McBSP DR signal comes from multiple AIC DOUT sources, and the DX signal from the McBSP goes to multiple 'AIC10 devices (Figure 1). To ensure correct data to/from the right device, the DRs and DXs in McBSP must be delivered in the order that matches the number of 'AIC10s and the position of each. This synchronization procedure is illustrated in Figure 5 and is supported by proper software configuration according to 'AIC10 hardware. This software configuration is highly hardware-dependent; every 'AIC10 hardware change or reconfiguration can require the software to be changed or rewritten. Chiefly for this reason, the "plug-and-play" algorithm is necessary.

Continued on page 12

Figure 4. Interface data format

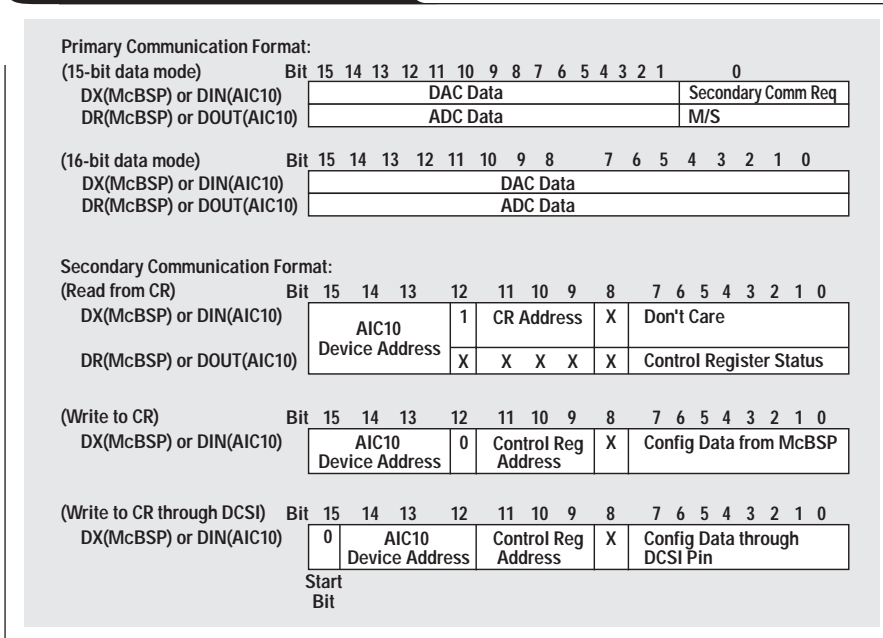
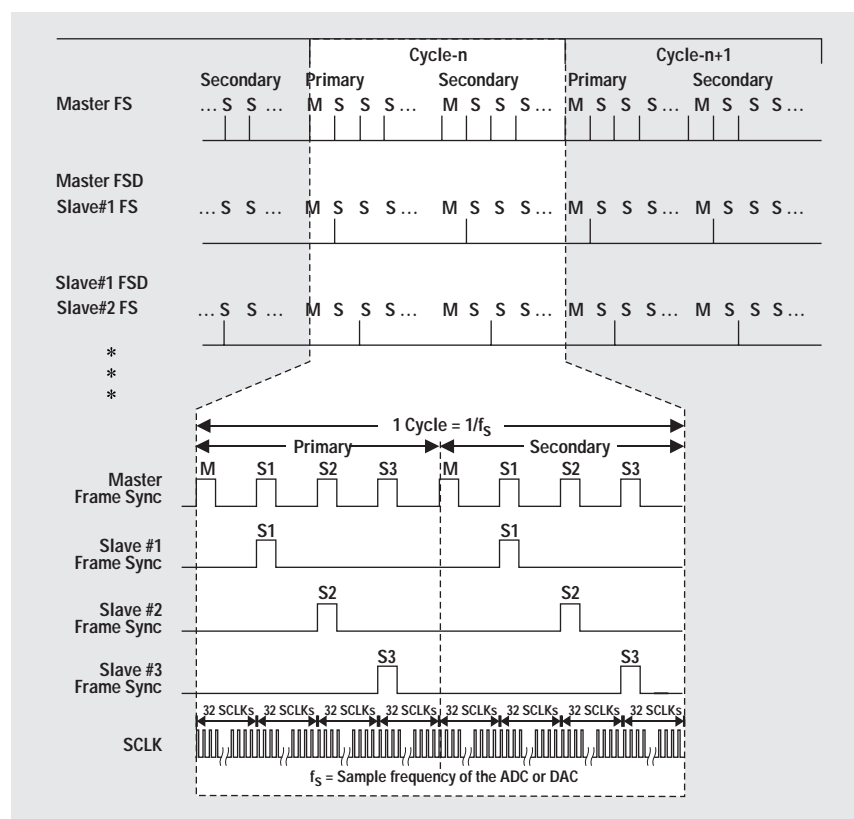


Figure 5. Paralleling 'AIC10 master/slave frame sync timing diagram



- Notes:
1. In master FS there are 32 SCLKs between a master/slave frame and a slave/slave frame.
 2. There are 256 (1 to 4 'AIC10s on board) or 512 (5 to 8 'AIC10s on board) SCLK pulses in each communication cycle (also called ADC/DAC sample interval), in which half (128 or 256) is for the primary phase and half is for the secondary phase.
 3. The secondary communication phase occurs only if required in the primary one.

Continued from page 10

Algorithm

The developed algorithm consists of two phases: automatic 'AIC10 hardware identification, called auto-identification; and automatic 'AIC10 software initialization or configuration, called auto-configuration.

Auto-identification

A new routine is added to the normal software initialization procedure (see Figure 6) that automatically identifies the on-board 'AIC10 hardware configuration and supplies the following information:

- how many 'AIC10 devices are on-board;
- which device, if any, is configured as a master device; and
- whether there are any hardware configuration mistakes, such as the M/S pins of two 'AIC10s pulled high (more than one master is not allowed).

These configurations need to be known before the second phase, auto-configuration, can take place.

The new identification routine completes the following steps:

Step 1. At system power-up, all 'AIC10 control registers enter their default condition, which means that the interface data format is in 15-bit mode. At this stage the McBSP transmitter is *not* enabled to ensure that no secondary communication is requested. Only the receiver is enabled to read data from AIC devices.

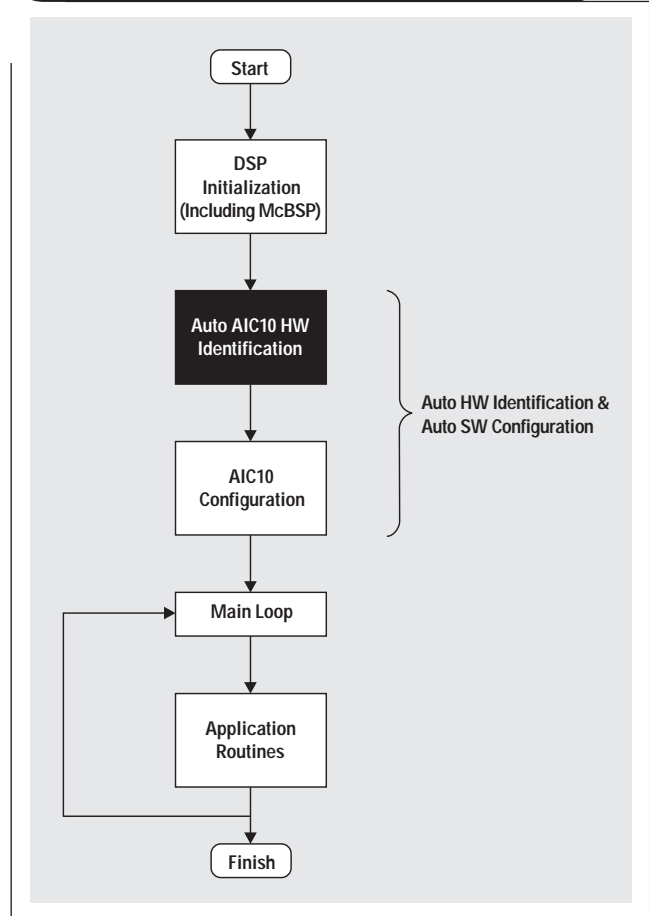
Step 2. To detect whether a master 'AIC10 is on board, the bit 0 of DR (called DOUT at 'AIC10) is checked for any data from a master 'AIC10 (refer to Figure 4).

Step 3. The communication frame number is counted before the first master 'AIC10 occurs. If the number is more than 8 (since a maximum of 8 'AIC10s can be paralleled in a system), a warning indication is displayed for "no master AIC devices."

Step 4. Otherwise, a master frame is detected at Step 3 before the frame counter reaches 8. Therefore, there is at least a master 'AIC10 on board, so the corresponding flag is set; for example, the MasterOnFlag.

Step 5. To identify the total number of 'AIC10s, the communication frames are counted, starting from the first time a master frame occurs to the next master frame (refer to Figure 5). The result, named AIC10Num, is saved. The range of the number can be from 1 to 8.

Figure 6. Top-level flow chart of the main software routine with auto-identification



Otherwise the auto-identification failure flag (IdentFail) is set, a warning message is displayed, or an alarm sounds, aborting the system to reset condition.

Step 6. The system waits for AIC10Num-1 frames before enabling the McBSP transmitter to the starting point of the next primary communication, and checks whether bit 0 is logic "0" at each DR during the (AIC10Num-1) frames. If the check fails, the IdentFail flag is set, a warning message is displayed, and the system aborts to reset condition.

Step 7. The system repeats the secondary communication request AIC10Num times by writing 0x0001 to McBSP's DX register and transmitting. It also reads bit 0 at the first DR to make sure it is logic "1" (master). Otherwise, it sets the IdentFail flag, displays a warning message, and aborts to reset condition.

Step 8. Once the first secondary communication arrives, the system sets and transmits DX with a request to read the master device's CR1, the control register #1. Then, during the same frame, it reads DR, masks out the AIC's

Figure 7. Timing diagram of the auto-identification procedure

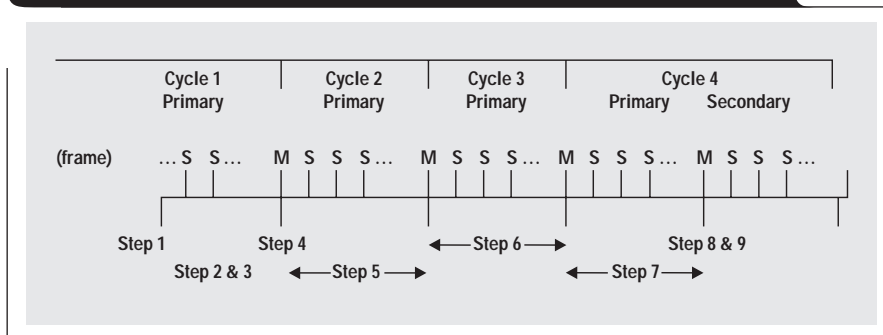
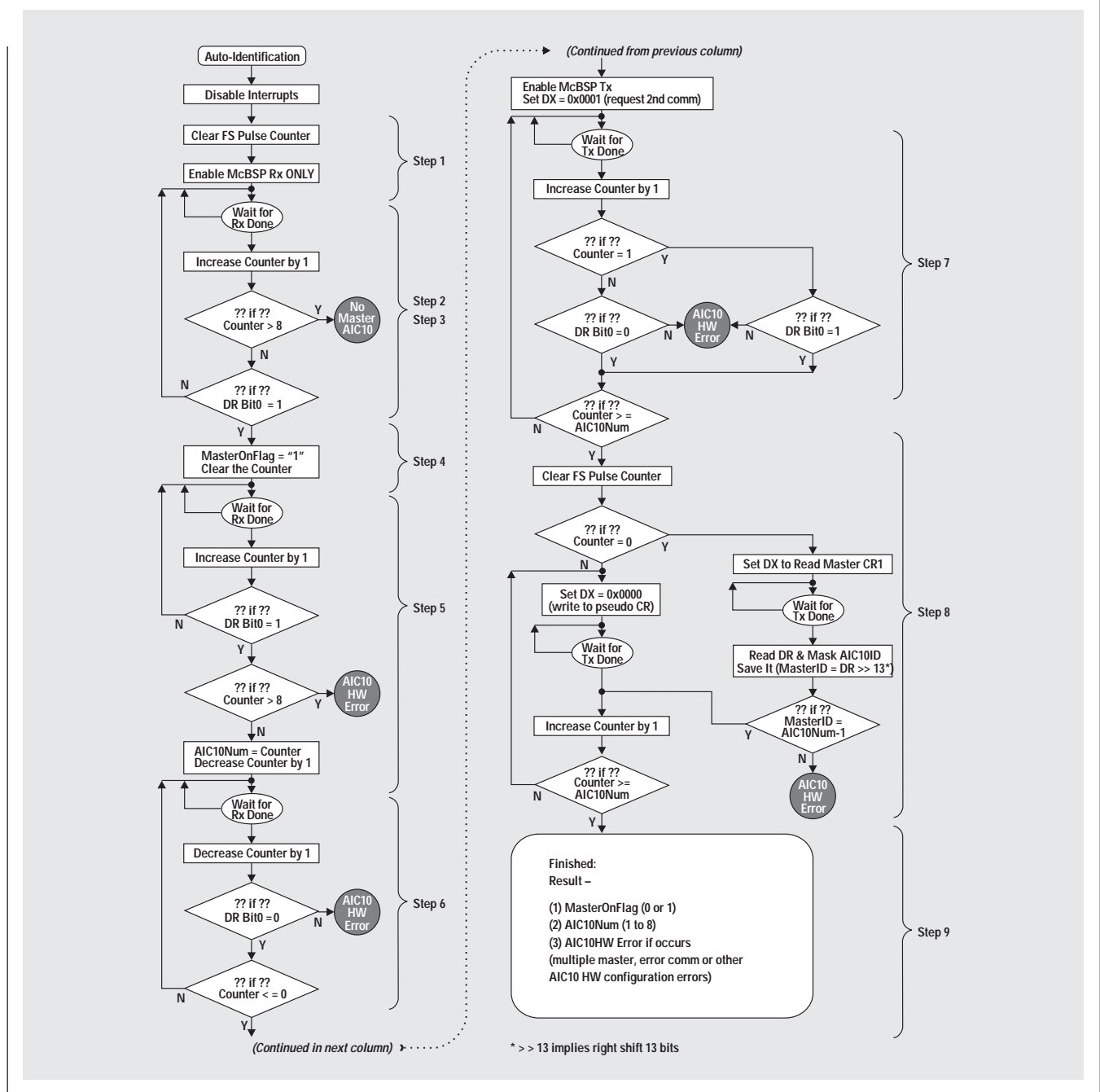


Figure 8. Automatic AIC hardware configuration and identification flow chart



device ID, and saves it to MasterID. The MasterID is compared with AIC10Num-1. If they are not equal to each other, the IdentFail flag is set, an alarm sounds, and the system aborts to reset condition.

Step 9. The auto-identification of the hardware configuration is successfully completed. There is only one master 'AIC10 device on board (the IdentFail flag would be set if more than one master appeared), and there is a total of AIC10Num 'AIC10 devices.

Figure 7 shows the full timing diagram of the auto-identification procedure and indicates the task at each of

the time intervals. It takes a maximum of 4 communication cycles to finish the auto-identification procedure. There are 256 (if there are 4 or less 'AIC10s) or 516 (if there are 5 to 8 'AIC10s) SCLKs within a communication cycle, and there are 32 SCLKs between each frame as illustrated in Figure 5.

The new auto-identification algorithm can be implemented by coding according to the flow chart in Figure 8.

Continued on next page

Continued from previous page

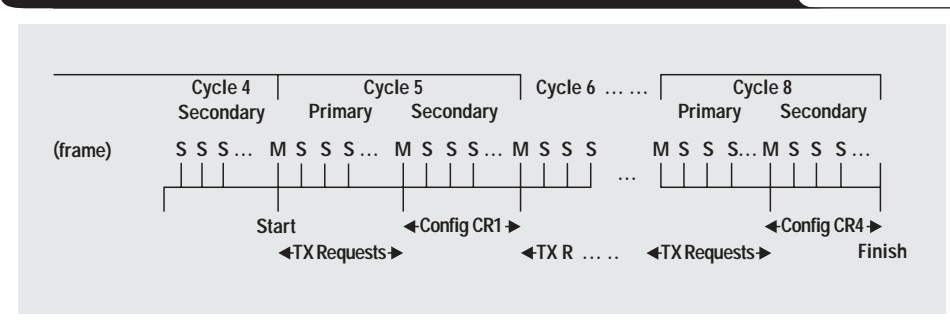
Auto-configuration

After the AIC hardware configuration is identified and the software establishes the number of 'AIC10s on board and that there is only one master, then the auto-configuration (software configuration) of the AIC control registers becomes a trivial task. Note that there are 4 control registers in an 'AIC10 device. Hence, 4 complete communication cycles are needed for the configuration. Each of the cycles includes a primary communication phase, which transmits the secondary communication requests; and a secondary communication phase, which programs the 'AIC10's control registers. Figure 9 shows the timing diagram of the auto-configuration that follows the auto-identification procedure.

Conclusion

This article presented a "plug-and-play" algorithm that uses the McBSP on the TMS320C5402 DSP to identify automatically the TLV320AIC10 EVM master/slave structure and the number of on-board TLV320AIC10 analog devices. The software automatically configures the 'AIC10s to initiate stable communication between the McBSP and 'AIC10s. The algorithm offers TLV320AIC10 users a handy and powerful tool for their applications.

Figure 9. Timing diagram for the auto-configuration procedure



References

For more information related to this article, you can download an Acrobat Reader file at www-s.ti.com/sc/techlit/litnumber and replace "litnumber" with the **TI Lit. #** for the materials listed below.

- | Document Title | TI Lit. # |
|---|------------------|
| 1. TLV320AIC10/11 EVM User's Guide | .slwu003 |
| 2. <i>TMS320C54x DSP Reference Set</i> ,
Vol. 5: <i>Enhanced Peripherals</i> , Section 2 . . . | .spru302 |
| 3. TLV320AIC10 Data Manual | .slws093 |

Related Web sites

- www.dataconverter.com
- www.ti.com/sc/docs/products/analog/tlv320aic10.html
- www.ti.com/sc/docs/products/dsp/index.htm