TAS3004 Software Reference Design Concept

Keith Engler *Digital Audio*

ABSTRACT

This document contains a reference design to describe how to implement a TAS3004 with a Microcontroller.

This document is strictly a software system design concept for reference purposes and is not available as a working system.

This document outline is based upon the Software Requirements Specification (SRS) outline in DOD-2167A to assist those developing software to SEI guidelines.

This system can work with any microcontroller or any of the TI digital audio processors.

TABLE OF CONTENTS

TABLE OF FIGURES

TABLE OF TABLES

DISCLAIMER

The information contained in this document is a practical reference design and the implementation is not guaranteed. The accuracy of the algorithms needs to be verified by the persons implementing them in their system. If this document has discrepancies with the TAS3004 Data Manual the TAS3004 Data Manual should take precedence.

1. IDENTIFICATION

1.1. Overview

The Digital Speaker Demo Board (DSDB) is designed to be a motherboard with positions for two TAS3004 devices. The DSDB contains a microcontroller to interpret toggle switch inputs and appropriately control the operation of the two TAS3004s.

The input to the board is SPDIF. The digital audio input comes from a digital audio device such as a PC or a CD player. The use of SPDIF eliminates the need to write software for the PC, because an SPDIF output audio card is used and software is supplied with it.

A stereo audio stream is brought into the CS8412 for decoding into IIS. The IIS data is routed to the FPGA where a delay is added to attempt to correct the phase between the R/L speakers and the sub-woofer. The IIS audio stream is sent to the TAS3004 for equalization, volume, and tone control and then converted to analog. This is the analog output for the right and left channels.

The IIS audio stream, which is sent to the FPGA, also converts the audio stream, which is the right and left channel information, into two left channel IIS streams (the right channels are center scale or zero). These two audio streams are sent to a second TAS3004, which digitally mixes them. This new stream, which is the sum of the right and left channels, is then equalized, and a low pass filter applied to it. Its volume is also controlled by the TAS3004. It is then converted to analog by the left channel DAC in the TAS3004. This is the analog output for the sub-woofer.

The DSDB also contains a microcontroller. This device is used to control the TAS3004s. It provides the filter coefficients for volume, tone, parametric equalization, etc. by decoding switches mounted on the DSDB.

The DSDB provides a demonstration of the TAS3004 in both a right and left speaker and a sub-woofer application. The following diagram illustrates this system.

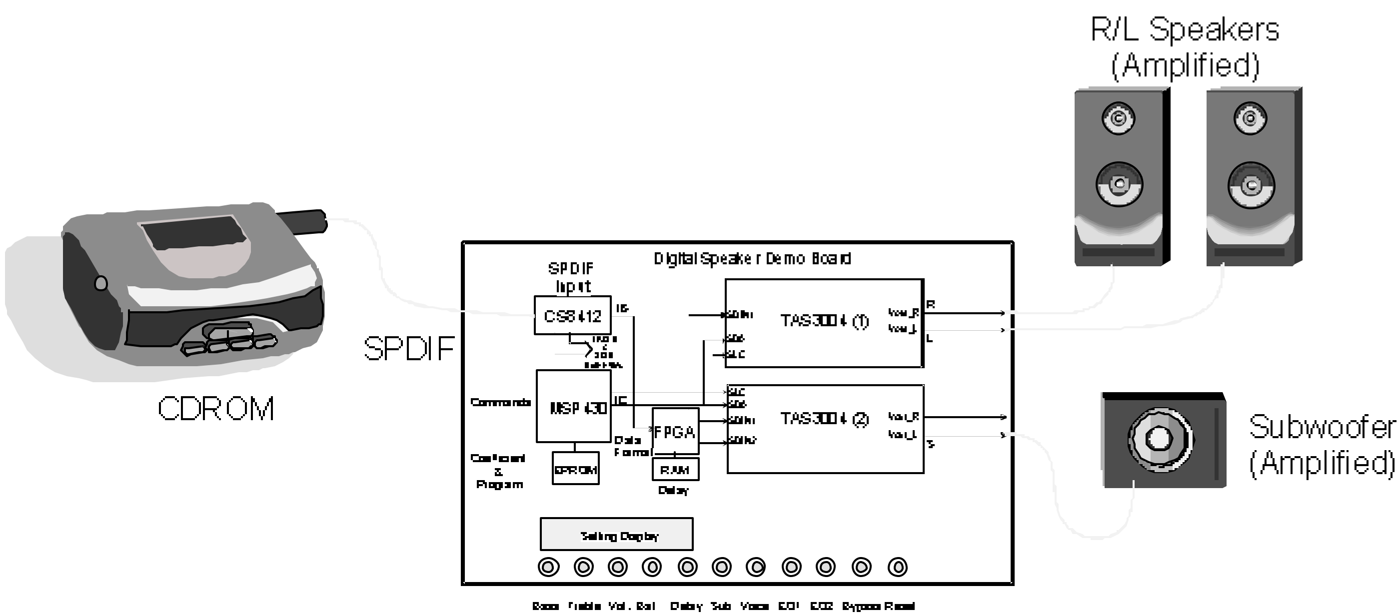


Figure 1: DSDB Description

1.1.1.Description of Board Mounted Switches

1.1.1.1. Bass Switch

The bass switch is a two-position (center off) momentary board-mounted switch that allows the bass in right/left channel speaker to be boosted or retarded.

1.1.1.2. Treble Switch

The treble switch is a two-position (center off) momentary board-mounted switch that allows the treble in the right/left channel speakers to be boosted or retarded.

1.1.1.3. Volume Switch (Volume Control)

The volume switch is a two-position (center off) momentary board-mounted switch that allows the volume in the right/left (R/L) channel and the sub-woofer to be increased or decreased. The sub-woofer may be setup to decrease at some percentage less than R/L speakers for improved performance at low listening levels.

1.1.1.4. Balance Switch (Balance Control)

The balance switch is a two-position (center off) momentary board-mounted switch that varies the volume between the right and left speakers.

1.1.1.5. Delay

The delay switch is a two-position (center off) momentary board mounted switch that varies the delay between the R/L channel and the sub-woofer.

1.1.1.6. Bypass

The bypass switch is an on/off board mounted switch that removes parametric equalization from both of the TAS3004 devices.

1.1.1.7. Sub (Sub-Woofer Volume)

The sub-woofer volume switch is a two position (center-off) momentary board mounted switch that allows the sub-woofer volume to be increased or decreased. Equalization curves for the appropriate sub-woofer is selected by the EQ1 switch.

1.1.1.8. Voice

The voice switch is a board mounted on/off switch that turns off the sub-woofer, the parametric equalization, and applies a parametric equalization curve to the R/L channel, which enhances the understanding of the human voice. This feature allows the speakers to operate at a lower listening level.

1.1.1.9. EQ1 (Loudness Contour)

The EQ1 switch is a board mounted on/off switch that applies an equalization curve that has been optimized for low listening levels for either of the speakers selected.

1.1.1.10. EQ2

The EQ2 switch is a board mounted on/off switch that applies an equalization curve that is optimized for a second manufacturer’s speakers. (When this switch is in the off position, it selects the primary manufacturer’s equalization curve.

1.1.1.11. Reset

The reset switch is a board mounted momentary switch that applies a reset signal to chips on the DSDB so that they assume their original condition at power-up.

2. Design Features

The design features are categorized as functional, performance and interface.

2.1.Functional Requirements

2.1.1.Synopsis

The microcontroller is used to provide control information to the TAS3004 via an IIC port.

The microcontroller polls for commands from the switches mounted on the TAS3004 DSDB. It then decodes which switch is pressed and references the tables to find the appropriate control value. This value is then sent to the TAS3004 via the IIC port. The DSDB controls two TAS3004s. One for the left and right speakers referred to as main TAS3004, and one referred to as sub TAS3004 which controls the sub-woofer.

2.1.2.Program Flow

2.1.2.1.Program Block Structure

The following shows the block layout of the DSDB Assembly Language Software. Further descriptions of each block and the functions within each block are described later in this document.

1. Heading
2. Includes of Vendor supplied definitions
3. Configuration Bit Definition
4. Program Variable Definition
5. Table Definition
6. Initialization Section (Functions and Variables)
7. I2C Commands
8. Delay Functions
9. Sub-Woofer (Sub-TAS3004) Command Functions
10. Left and Right Speaker (Main TAS3004) Command Functions
11. Biquad Filter Data Command Functions
12. Port B Functions
13. Port C Functions
14. Display Functions
15. Port D Functions
16. Main Program

2.1.2.2.Program Functional Description

The logical flow of the program is graphically represented in the flow charts at the end of this document. The text within that section describes how to evaluate the flowcharts. The description of each function is in the Functional Description section of this document.

2.1.3.Variable Description

This section briefly describes the constants and variables used in the program. This is for definition and description. Actual code should be referred to for further clarification. The main purpose of the program is to read a change in the switch settings and increment or decrement a counter. The counter is used as an offset into a table where the actual value to be sent to the TAS3004 exists.

Conceptually, the most difficult mathematics involved in this program are to add and subtract counters. Counters exist for the bass, treble, volume, left volume, right volume, sub-woofer volume, biquad filters and mixers.

2.1.4.Program Block and Function Description

This section describes the block and the various functions or sub-routines within each block.

2.1.4.1. Program Header

This is a standard program header, which has a short description and revision description.

2.1.4.2. Program Variable Definition

The following variables have been defined in the in general purpose RAM within the microcontroller.

```
TempB       EQU  0X1A   ; Holds the contents of Port B
TempC       EQU  0X1B   ; Holds the contents of Port C
TempD       EQU  0X1C   ; Holds the contents of Port D
SaveD       EQU  0X1D   ; keep the contents of Port D to see if it changes.
```

The remaining program variables have been defined in Bank 0 of the general purpose RAM.

2.1.4.3. Table Definition

The tables in program memory (ROM) are defined in this section. It is important that the total number of bytes within each table does not exceed 0XFF or 255 decimal. This is determined by the use of 8-bit registers. Therefore the large tables such-as the volume table has to be divided into three separate tables. It is important to define a table in an even memory boundary where the lower byte starts at 0X00. This assures that the program counter does not exceed 256 bytes as long as the table is less than 256 and is accomplished by using the ORG command. The following two sections describe a table fragment and a method to access the table.

2.1.4.3.1. Example Table Fragment

```
                ORG  0x100              ; start table at a even boundary.
VolB1TableRead  ADDWF PCL, 1            ; Adds the value in WREG to the Program Counter
                DT   0X07      ; 18.0 dB    Offset 0X00
                DT   0X07      ; 17.5 dB    Offset 0X01
                DT   0X07      ; 17.0 dB    Offset 0X02
                DT   0X06      ; 16.5 dB    etc.
                DT   0X06      ; 16.0 dB
                DT   0X05      ; 15.5 dB
                DT   0X05      ; 15.0 dB
                DT   0X05      ; 14.5 dB
                DT   0X05      ; 14.0 dB
                DT   0X04      ; 13.5 dB
                DT   0X04      ; 13.0 dB
```

2.1.4.3.2. Example Data Retrieval from a Table

```
MOVLW     0X02             ; Move 0X02 into the Working Register (WREG)
CALL      VolB1TableRead   ; Call the label defined at Offset 0X00 in the table above.
```

The DT command in the table returns the value 0X07 corresponding to 17.0 dB in WREG

2.1.4.4. Initialization Section (Functions and Variables)

2.1.4.4.1. initPortA

Prepare Port A to send I2C data.

2.1.4.4.2. initPortB

Prepare Port B to read the volume, bass and treble switches.

2.1.4.4.3. initPortC

Prepare Port C to read the sub-volume, and balance switches.

Prepare Port C to send a series of delay pulses to the FPGA.

2.1.4.4.4. initPortD

Prepare Port D to read the status of the static switches for the biquad filter and EQ configuration.

2.1.4.4.5. initPortE

Prepare Port E for the display

2.1.4.4.6. initPortF

Prepare Port F for the display

2.1.4.4.7. initPortG

Prepare Port G to send out a reset sync pulse. The hardware does not currently use this feature.

2.1.4.4.8. initVariables

Initialize the program variables to some startup condition.

2.1.4.4.9. InitMixers

Initialize the main TAS3004 and the sub TAS3004 mixer value. This is currently not used since the initial value is determined by the static switch conditions in Port D.

2.1.4.4.10. InitMainTAS3004 and Init Sub TAS3004

Setup the main TAS3004and the sub TAS3004 Control Registers to 0X69, which corresponds to the following configuration:

0110 1001 = 0X69

Where:

C(7) = 0 for Normal Mode

C(6) = 1 for SCLK = 64*FS

C(5:4) = 10 set Output Serial Port to I2S

C(3:2) = 10 set Input Serial Port to I2S

C(1:0) = 01 set Serial Port Word Length to 18 bits

2.1.4.4.11. InitBiquadFilters

Used to set the initial values of the mixer and the biquad filters as determined by the static switch conditions in Port D.

2.1.4.4.12. ResetSync

Sends out a reset pulse to sync the hardware. This is not being used.

2.1.4.4.13. initDisply

Initializes the display.

2.1.4.5. I2C Commands

2.1.4.5.1. StartI2C

Set the START enable bit (SEN) in SSPCON2.

I2C START condition is defined as:

High-to-low transition of the SDA when the SCL is high. The SDA line can only change state when the SCL line is low.

2.1.4.5.2. I2CSendByte

Data is in WREG when this function is called. The data in WREG is transferred to the SSPBUF and then all 8 bits are automatically shifted out the SDA pin after the falling edge of the SCL is asserted

2.1.4.5.3. StopI2C

Set the STOP enable bit (PEN) in SSPCON2.

I2C STOP condition is defined as:

Low-to-high transition of the SDA when the SCL is high. The SDA line can only change state when the SCL line is low.

2.1.4.5.4.WriteSubTAS3004

Calls the StartI2C command, addresses the sub TAS3004 with 0X6A and prepares the TAS3004 to be written to.

2.1.4.5.5.WriteMainTAS3004

Calls the StartI2C command, addresses the main TAS3004 with 0X68 and prepares the TAS3004 to be written to.

2.1.4.5.6.ReadSubTAS3004

Calls the StartI2C command, addresses the sub TAS3004 with 0X6B and prepares the TAS3004 to be read from.

2.1.4.5.7.ReadMainTAS3004

Calls the StartI2C command, addresses the main TAS3004 with 0X69 and prepares the TAS3004 to be read from.

2.1.4.5.8.ControlAddress

Addresses the Control Register in the TAS3004

2.1.4.5.9.VolAddress

Addresses the Volume Register in the TAS3004

2.1.4.5.10.TrebleAddress

Addresses the Treble Register in the TAS3004

2.1.4.5.11.BassAddress

Addresses the Bass Register in the TAS3004

2.1.4.5.12.Mixer1Address

Addresses the Mixer 1 Register in the TAS3004

2.1.4.5.13.Mixer2Address

Addresses the Mixer 2 Register in the TAS3004

2.1.4.5.14.LeftBQ0Address

Addresses the Left Biquad 0 Register in the TAS3004

2.1.4.5.15.LeftBQ1Address

Addresses the Left Biquad 1 Register in the TAS3004

2.1.4.5.16.LeftBQ2Address

Addresses the Left Biquad 2 Register in the TAS3004

2.1.4.5.17.LeftBQ3Address

Addresses the Left Biquad 3 Register in the TAS3004

2.1.4.5.18.LeftBQ4Address

Addresses the Left Biquad 4 Register in the TAS3004

2.1.4.5.19.LeftBQ5Address

Addresses the Left Biquad 5 Register in the TAS3004

2.1.4.5.20.RightBQ0Address

Addresses the Right Biquad 0 Register in the TAS3004

2.1.4.5.21.RightBQ1Address

Addresses the Right Biquad 1 Register in the TAS3004

2.1.4.5.22.RightBQ2Address

Addresses the Right Biquad 2 Register in the TAS3004

2.1.4.5.23.RightBQ3Address

Addresses the Right Biquad 3 Register in the TAS3004

2.1.4.5.24.RightBQ4Address

Addresses the Right Biquad 4 Register in the TAS3004

2.1.4.5.25.RightBQ5Address

Addresses the Right Biquad 5 Register in the TAS3004

2.1.4.6. Delay Functions

There are many functions defined for various program delays using NOP commands. It would be more efficient to rewrite this section to use timer interrupts.

2.1.4.7. Sub-Woofer (Sub-TAS3004) Command Functions

2.1.4.7.1. MixerSubTAS3004

This function changes the sub-woofer mixer values. Three bytes must be sent to change the mixer.

One counter is used to monitor the position in the Mixer tables.

2.1.4.7.2. VolSubTAS3004

Six bytes must be sent to change the sub-woofer volume. The first three bytes are for the left channel and the next three bytes are for the right channel. The two channels are combined together for one sub-woofer output.

2.1.4.7.3. MuteVolSubTAS3004

Six bytes must be sent to mute the sub-woofer volume. The first three bytes are for the left channel and the next three bytes are for the right channel. The two channels are combined together for one sub-woofer output. The last position in the volume table is the mute.

2.1.4.7.4. BassSubTAS3004

This function changes the sub-woofer speaker bass. Two bytes must be sent to change the bass. The first byte is for the left channel and the next byte is for the right channel. One counter is used to monitor the position in the bass table. One counter exists for each left and right channel called BassCounter.

2.1.4.8. Left and Right Speaker (Main TAS3004) Command Functions

2.1.4.8.1. MixerMainTAS3004

This function changes the left and right speaker mixer values.

2.1.4.8.2. VolMainTAS3004

Six bytes must be sent to change the left and right volume. The first three bytes are for the left channel and the next three bytes are for the right channel. Three counters exist for the volume. One the volume level which is called Vcounter, and one each for the left and right volume. They are called VLcounter and VRcounter and are used to control the balance.

2.1.4.8.3. MuteVolMainTAS3004

Six bytes must be sent to mute the left and right volume. The first three bytes are for the left channel and the next three bytes are for the right channel.

2.1.4.8.4. BassMainTAS3004

This function changes the left and right speaker bass. Two bytes must be sent to change the bass. The first byte is for the left channel and the next byte is for the right channel. One counter is used to monitor the position in the bass table. One counter exists for each left and right channel called BassCounter.

2.1.4.8.5. TrebleMainTAS3004

This function changes the left and right speaker treble. Two bytes must be sent to change the treble. The first byte is for the left channel and the next byte is for the right channel. One counter is used to monitor the position in the treble table. One counter exists for each left and right channel called TrebleCounter.

2.1.4.9. Biquad Filter Data Command Functions

2.1.4.9.1. Spk1normal

Send the Normal Biquad Filter data defined for the first set of speakers.

2.1.4.9.2. Spk2normal

Send the Normal Biquad Filter data defined for the second set of speakers.

2.1.4.9.3. Spk1Contour

Send the Contour Biquad Filter data defined for the first set of speakers.

2.1.4.9.4. Spk2Contour

Send the Contour Biquad Filter data defined for the second set of speakers.

2.1.4.9.5. VoiceBiquad

Send the Voice Biquad Filter data defined for the all speakers.

2.1.4.9.6. BypassBiquad

Send any speaker into an all-pass condition.

2.1.4.9.7. Subnormal

Send the Normal Biquad Filter data defined for the sub-woofer.

2.1.4.9.8. SubContour

Send the Contour Biquad Filter data defined for the sub-woofer.

2.1.4.10. Port B Functions

2.1.4.10.1. CalcVolPlus

Refer to the flowcharts at the end of this document for details of this function.

2.1.4.10.2. CalcVolMinus

Refer to the flowcharts at the end of this document for details of this function.

2.1.4.10.3. CalcBassPlus

Refer to the flowcharts at the end of this document for details of this function.

2.1.4.10.4. CalcBassMinus

Refer to the flowcharts at the end of this document for details of this function.

2.1.4.10.5. CalcTreblePlus

Refer to the flowcharts at the end of this document for details of this function.

2.1.4.10.6. CalcTrebleMinus

Refer to the flowcharts at the end of this document for details of this function.

2.1.4.10.7. EvalPortB

Refer to the flowcharts at the end of this document for details of this function.

2.1.4.10.8. InitIncrement

Refer to the flowcharts at the end of this document for details of this function.

2.1.4.10.9. SetIncrement

Refer to the flowcharts at the end of this document for details of this function.

2.1.4.11. Port C Functions

2.1.4.11.1. InitFPGAdelay

This is an implementation specific function.

2.1.4.11.2. FPGAdelayMinus

This is an implementation specific function.

2.1.4.11.3. FPGAdelayPlus

This is an implementation specific function.

2.1.4.11.4. CalcSubVolPlus

Refer to the flowcharts at the end of this document for details of this function.

2.1.4.11.5. CalcSubVolMinus

Refer to the flowcharts at the end of this document for details of this function.

2.1.4.11.6. NormalBalance

Refer to the flowcharts at the end of this document for details of this function.

2.1.4.11.7. MoreRightBalance

Refer to the flowcharts at the end of this document for details of this function.

2.1.4.11.8. MoreLeftBalance

Refer to the flowcharts at the end of this document for details of this function.

2.1.4.11.9. CalcBalanceRight

Refer to the flowcharts at the end of this document for details of this function.

2.1.4.11.10. CalcBalanceLeft

Refer to the flowcharts at the end of this document for details of this function.

2.1.4.11.11. EvalPortC

Refer to the flowcharts at the end of this document for details of this function.

2.1.4.12.Display Functions

2.1.4.12.1. ChooseASCIIchar

Converts a number to an ASCII character.

2.1.4.12.2. WriteToDisply

Writes a value to the display.

2.1.4.12.3. WaitToDisply

Gets the display ready to be written to.

2.1.4.12.4. Digit Selection Functions (Eight)

Eight functions exist to select which digit to write to for the display.

2.1.4.12.5. ASCII Character Functions (various)

A function exists for most of the characters and numbers in the ASCII table.

2.1.4.12.6. Display functions (various)

Various functions exist to send a particular message to the display.

2.1.4.13. Port D Functions

2.1.4.13.1. Scroll

Refer to the flowcharts at the end of this document for details of this function.

2.1.4.13.2. ImplementBypass

Read the bypass switch and send the biquad to the TAS30004.

2.1.4.13.3. ImplementVoice

Read the voice switch and send the biquad to the TAS30004.

2.1.4.13.4. Spk1NormEQ

Read the Norm EQ 1 switch and send the biquad to the TAS30004.

2.1.4.13.5. Spk2NormEQ

Read the Norm EQ 2 switch and send the biquad to the TAS30004.

2.1.4.13.6. Spk1ContourEQ

Read the Contour EQ 1 switch and send the biquad to the TAS30004.

2.1.4.13.7. Spk2ContourEQ

Read the Contour EQ 2 switch and send the biquad to the TAS30004.

2.1.4.13.8. NormalEQ

Read the Normal EQ switch and send the biquad to the TAS30004.

2.1.4.13.9. ContourEQ

Read the Contour EQ switch and send the biquad to the TAS30004.

2.1.4.13.10. EvalPortD

2.1.4.14. Main Program

The main program has an initialization section and the main loop is looking for a switch to change.

2.2. Performance Requirements

The microcontroller and TAS3004 core features can be listed here.

2.3. Interface Requirements

This section identifies the requirements imposed on the software because of its relationship to other hardware and/or software. The software is downloaded to the microcontroller. The DSDB is controlled with toggle switches.

2.3.1. I2C Address

The IIC address of the TAS3004 is 0x68

2.3.2. Toggle Switch Descriptions

There are two types of switches on the DSDB that interface to the microcontroller. The two types consist of momentary switches and static switches. Tables 10, 11, and 12 show the switches, how they map to the microcontroller and the value of their default conditions.

The momentary switches are normally held low. Therefore a quick poll of the port where the momentary switch is connected shows a switch change if the value is not zero. Interrupts are not required to determine a change in a switch setting. The polling speed of the microcontroller is much faster than a human depressing a toggle switch.

The default state of the static switches is user defined and unknown by the software. Therefore the condition of the static switches is stored in memory to determine if the switch changes. A quick poll of the port where the static switch is connected shows a switch change if the value is not equal to the previous value.

Switch	Volume Plus	Volume Minus	Bass Plus	Bass Minus	Treble Plus	Treble Minus		
Pin #	RB2	RB5	RB4	RB3	RB1	RB0		
Bit Position	Bit7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Default = 0	0	0	0	0	0	0	X	X

Table 1 : Momentary Switches Port B

Switch	Balance Left	Balance Right	Sub-Volume Plus	Sub-Volume Minus	Delay Plus	Delay Minus		
Pin #	RC7	RC6	RC5	RC4	RC3	RC2		
Bit Position	Bit7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Default = 0	0	0	0	0	0	0	X	X

Table 2 : Momentary Switches Port C

Switch	Bypass	Voice	Loudness Contour	EQ Select	Status Display			
Pin #	RD6	RD7						
Bit Position	Bit7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Default is Not Known	X	X	X	X	X	X	X	X

Table 3 : Static Switches Port D

3. TESTING

This section identifies and defines the testing and validation requirements. It includes methods, levels of testing, tools, facilities and personnel requirements. A table showing the source of and the method for testing each requirement has been developed and is too detailed for this document.

The Program should be able to handle the following interaction:

- ✓ Read multiple input on the Toggle Buttons: The program is able to read all the input ports and then only act on the input of the highest priority.
- ✓ Handle the case where one button is held down continuously: The program repeats the following sequence (read - store - clear - act - delay). . Therefore, the program's built-in delay prevents the controller from running away.

4. Flowcharts

This section contains the flow charts describing the flow the software within the microcontroller. The flow charts are partitioned into multiple figures. The logical flow of the program is graphically represented in the flow charts. The text within this section describes how to evaluate the flowcharts that follow.

The job of the main program is to perform the proper initialization and then poll for a change to one or more of the switches on the board. The main loop is represented graphically in Figure 1. The microcontroller assembly language commands are inserted in the flow chart. Refer to Section 2.3.2. Toggle Switch Descriptions to determine how each individual switch maps to the microcontroller ports.

To follow the flow of the program, assume that the toggle switch was hit to increase the volume. In Table 1, the “Volume Plus” toggle points to Port B bit 7. Therefore since the variable “TempB” contains the value at Port B, the command “BTFSC” skips the command directly after it, if TempB is not zero. Since Temp B bit 7 is 1 it then executes the following function “EvalPortB”. The function “EvalPortB” determines which bit changed in port B. A description for “EvalPortB” is in Figure 2. Figure 4 shows the flow for the “Volume Plus” function. The “Volume Plus” function only calculates the position in the data table for the volume. The action for searching the data tables is not explicitly shown in the flow chart.

Note that the volume, bass, treble, sub volume, and delay are all similar algorithms. The balance algorithm is unique and all the Port D switches just require table lookups.

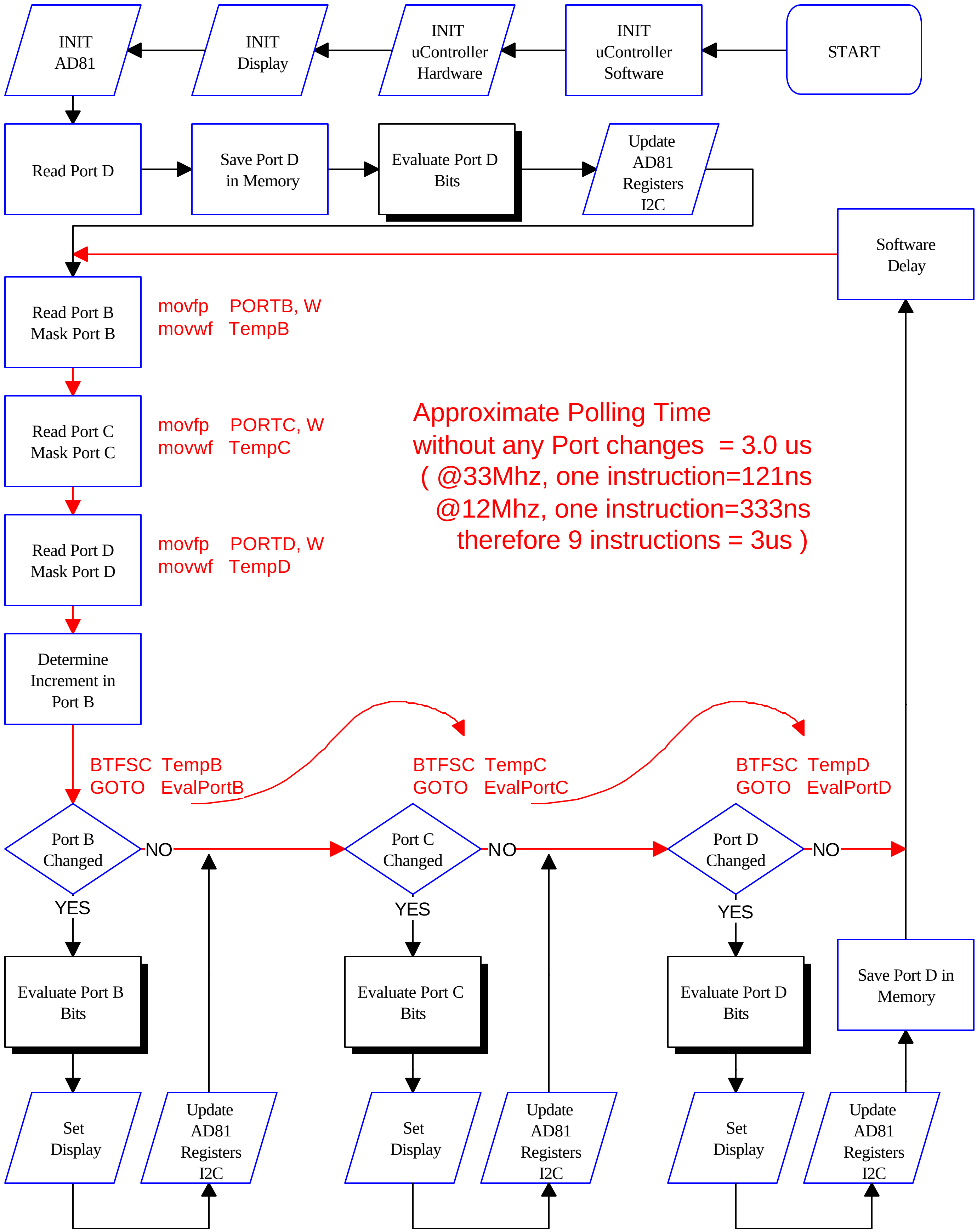
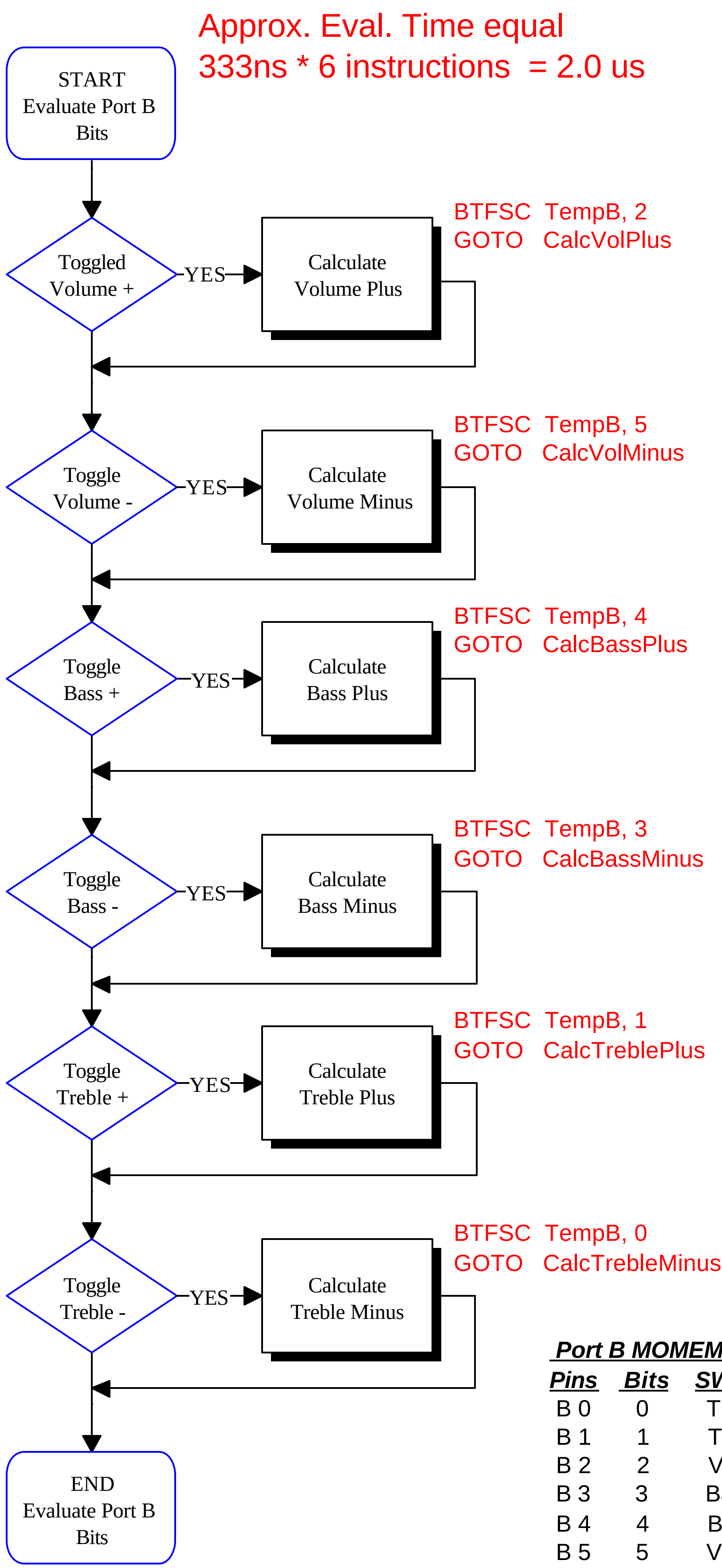


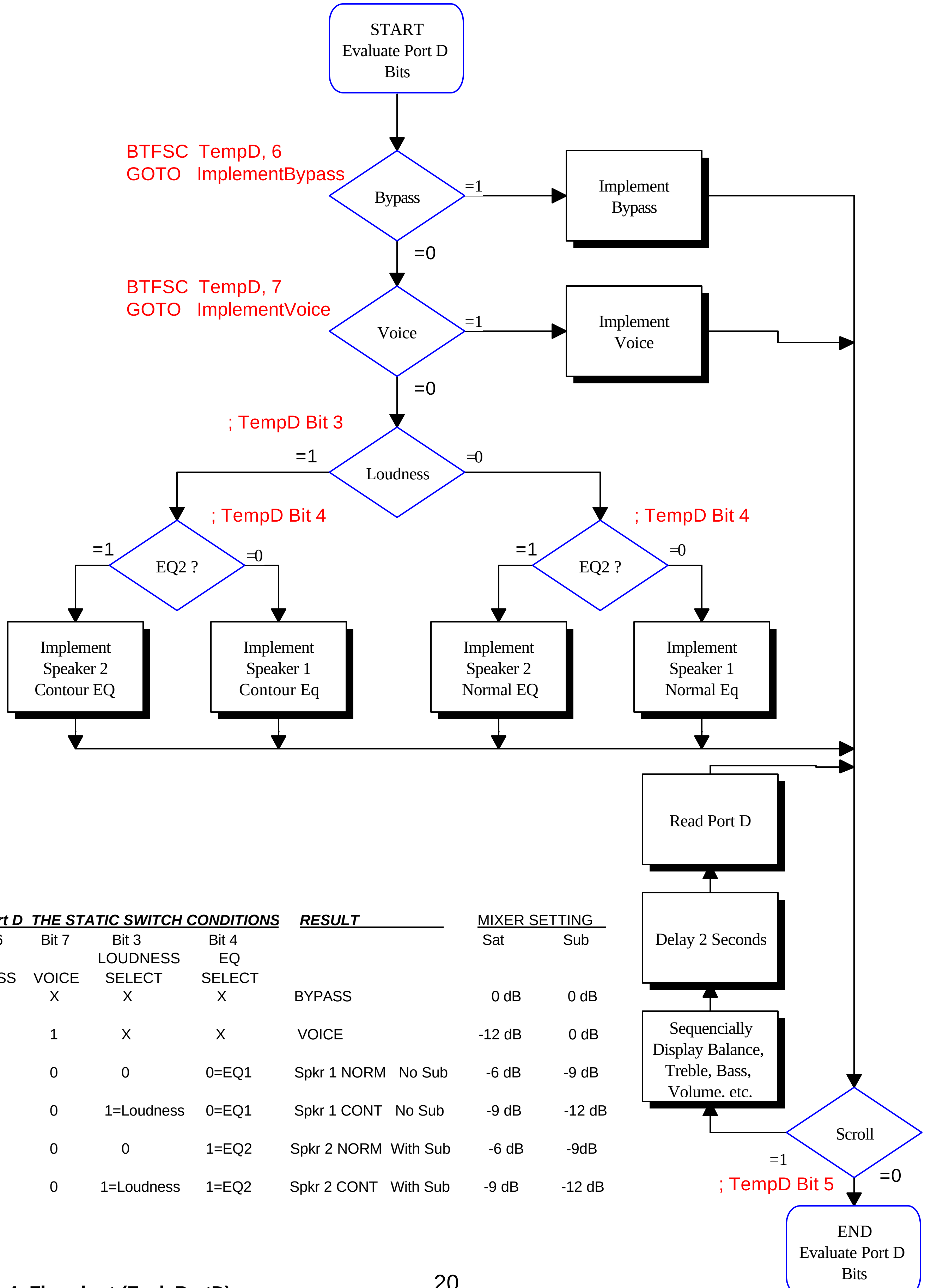
Figure 2: Flowchart (Main Program)



Port B MOMEMTARY SWITCHs

Pins	***Bits***	***SWITCH***	***COMMENT***
B 0	0	Treble -	
B 1	1	Treble+	
B 2	2	Volume +	
B 3	3	Bass -	
B 4	4	Bass +	
B 5	5	Volume -	
B 6	6	Increment Select	Static Switch
B 7	7	Increment Select	Static Switch

Figure 3: Flowchart (Eval. PortB & PortC)



Port D THE STATIC SWITCH CONDITIONS				RESULT		MIXER SETTING	
Bit 6	Bit 7	Bit 3	Bit 4			Sat	Sub
BYPASS	VOICE	LOUDNESS SELECT	EQ SELECT				
1	X	X	X	BYPASS		0 dB	0 dB
0	1	X	X	VOICE		-12 dB	0 dB
0	0	0	0=EQ1	Spkr 1 NORM	No Sub	-6 dB	-9 dB
0	0	1=Loudness	0=EQ1	Spkr 1 CONT	No Sub	-9 dB	-12 dB
0	0	0	1=EQ2	Spkr 2 NORM	With Sub	-6 dB	-9dB
0	0	1=Loudness	1=EQ2	Spkr 2 CONT	With Sub	-9 dB	-12 dB

Figure 4: Flowchart (Eval. PortD)

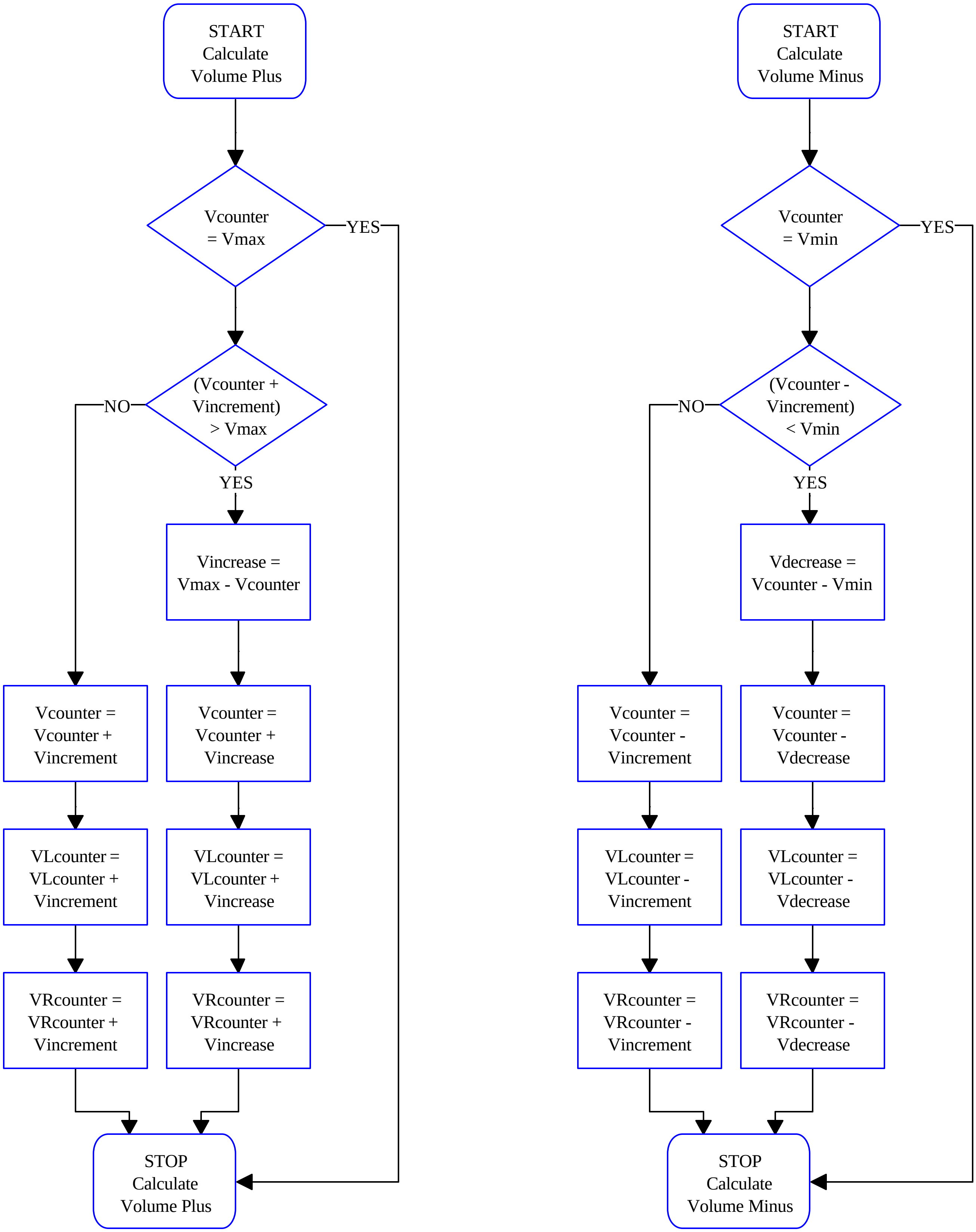


Figure 5: Flowchart (Calculate Volume)

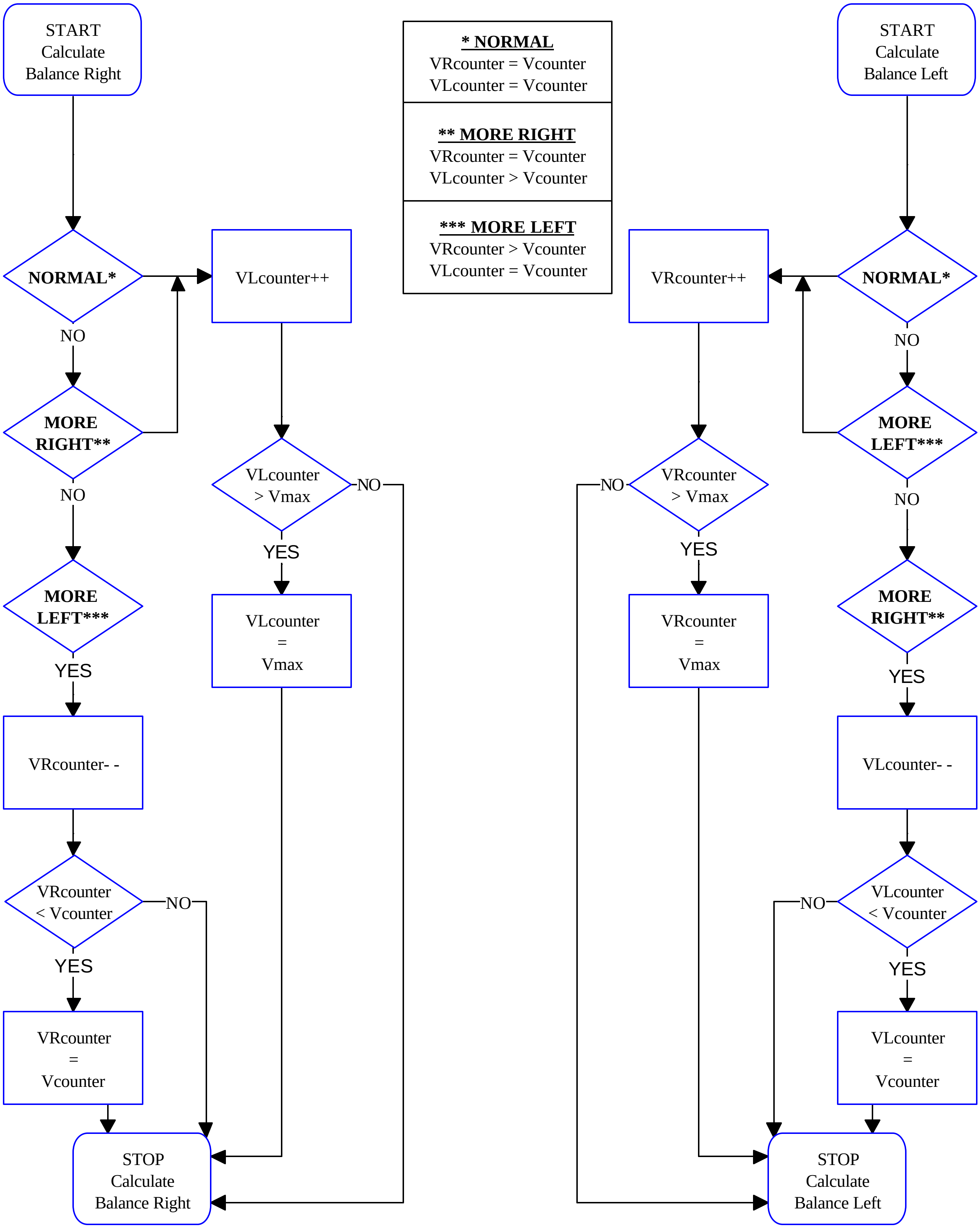
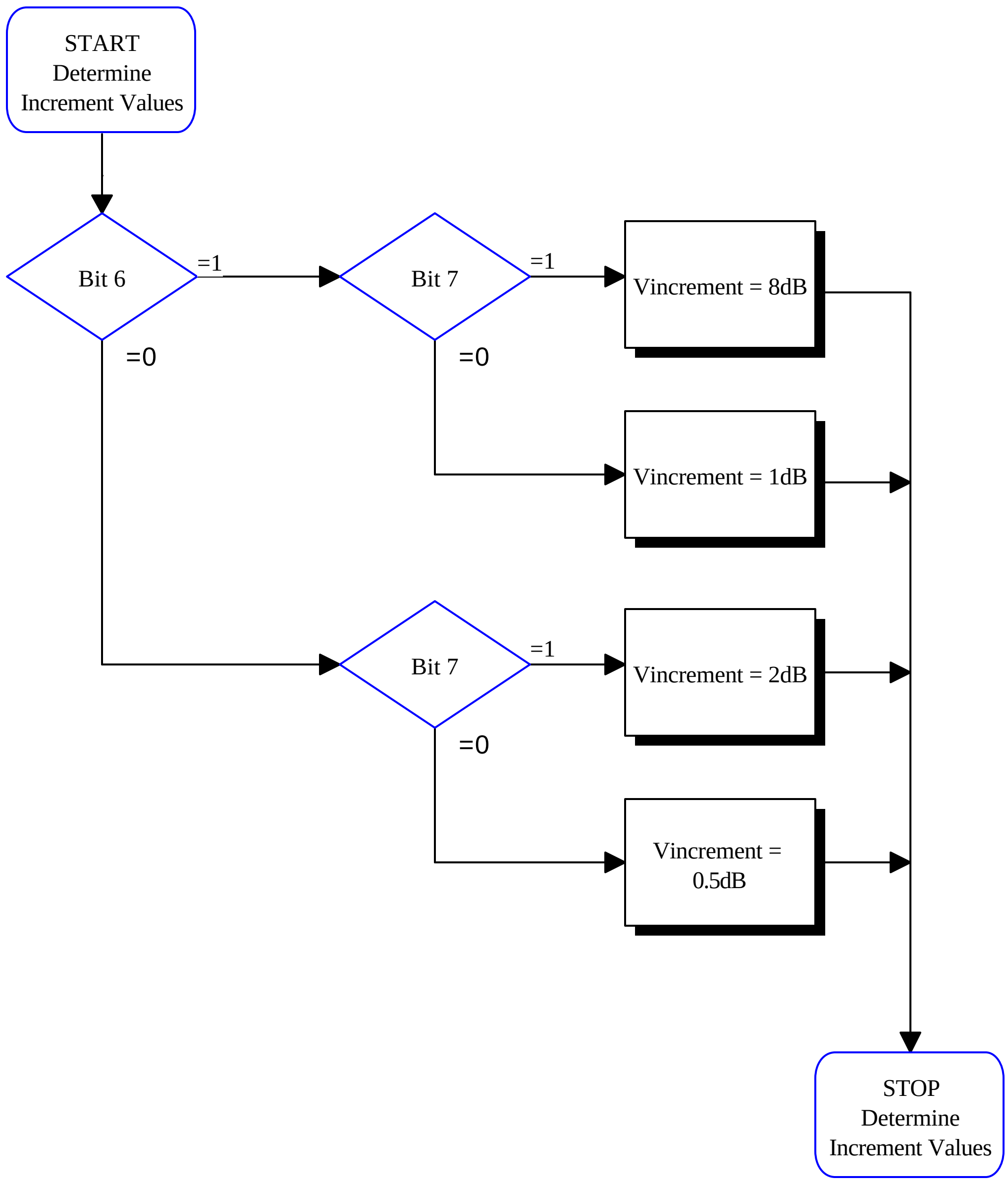


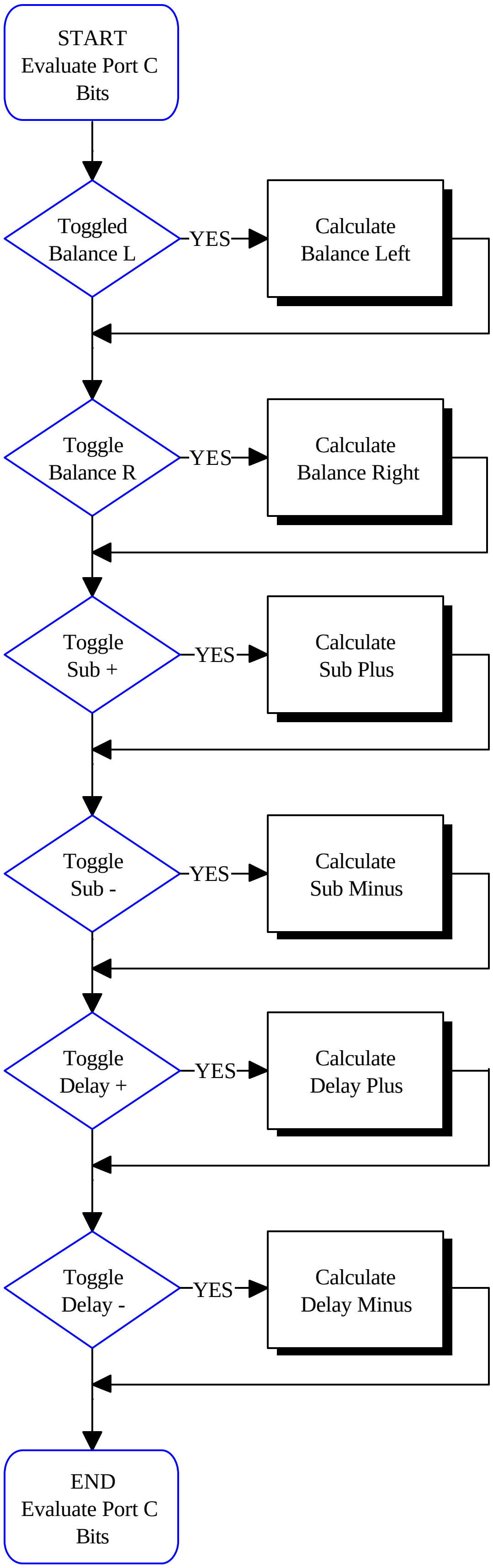
Figure 6: Flowchart (Calculate Balance)



Port B

STATIC SWITCH CONDITIONS		***RESULT***
Bit 7=INC1	Bit 6=INC0	
0	0	0.5 dB Increment
0	1	1 dB Increment
1	0	2 dB Increment
1	1	8 dB Increment

Figure 7: Determine Increment



Port C MOMENTARY SWITCHs

Pins	***Bits***	***SWITCH***	***COMMENT***
C0	0	Delay Out -	Output to FPGA
C1	1	Delay Out +	Output to FPGA
C2	2	Delay In -	
C3	3	Dealy In +	
C4	4	Sub Volume -	
C5	5	Sub Volume +	
C6	6	Balance L	
C7	7	Balance R	

Figure 8: Evaluate Port C

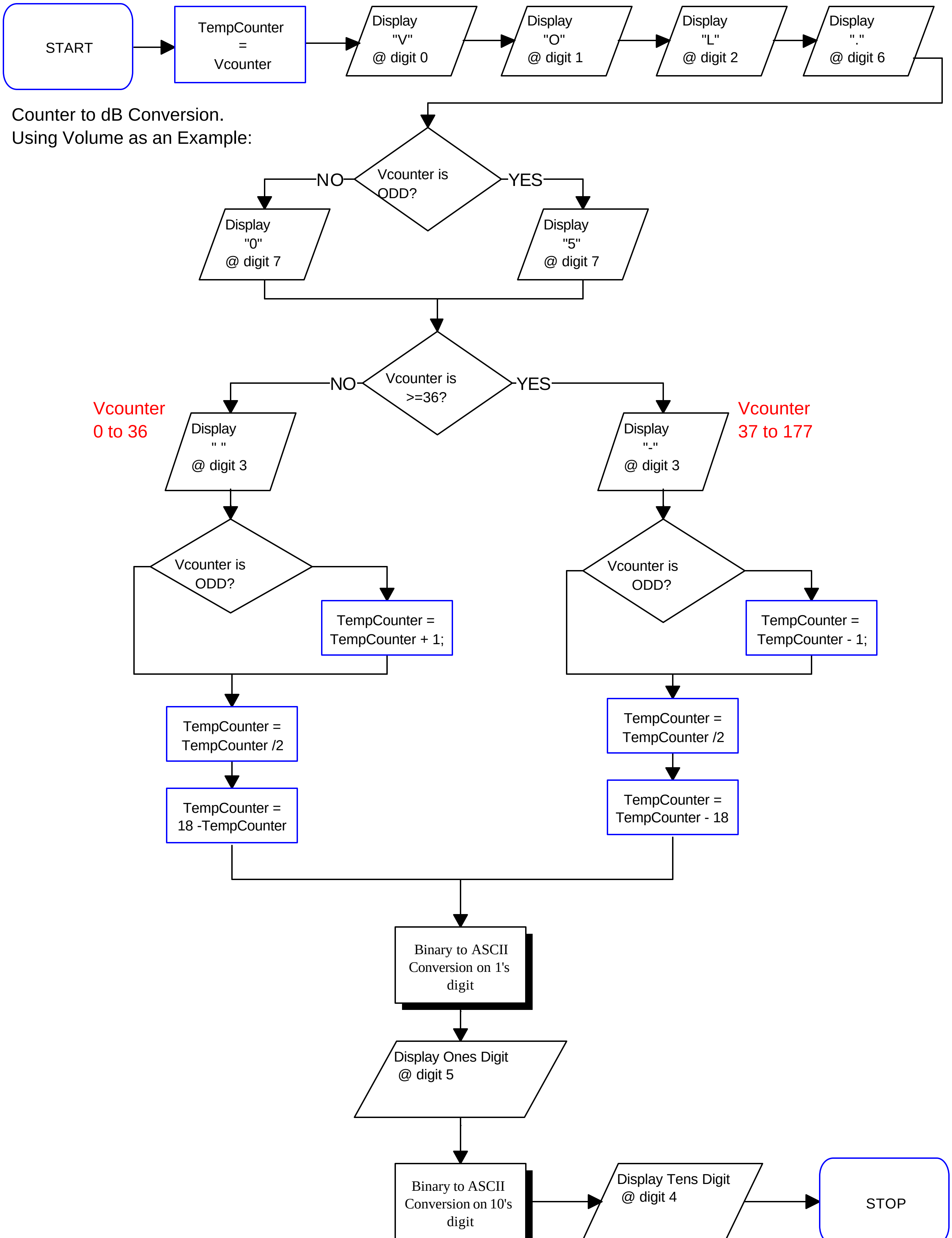


Figure 9: Hex to ASCII Conversion

Following are URLs where you can obtain information on other Texas Instruments products and application solutions:

Products		**Applications**	
Amplifiers	amplifier.ti.com	Audio	www.ti.com/audio
Data Converters	dataconverter.ti.com	Automotive	www.ti.com/automotive
DSP	dsp.ti.com	Broadband	www.ti.com/broadband
Interface	interface.ti.com	Digital Control	www.ti.com/digitalcontrol
Logic	logic.ti.com	Military	www.ti.com/military
Power Mgmt	power.ti.com	Optical Networking	www.ti.com/opticalnetwork
Microcontrollers	microcontroller.ti.com	Security	www.ti.com/security
		Telephony	www.ti.com/telephony
		Video & Imaging	www.ti.com/video
		Wireless	www.ti.com/wireless

Mailing Address: Texas Instruments
Post Office Box 655303 Dallas, Texas 75265