

Audio4-USB



Reference Manual

**Audio4-USB
User's Guide,**
September, 2003

Domain Technologies, Inc.
811 East Plano Pkwy, Suite 115
Plano, Texas 75074
Tel.: (972) 578-1121
Fax: (972) 578-1086
E-mail: support@domaintec.com
Web page: <http://www.domaintec.com>

Disclaimer of Warranty

This software package is provided on an 'AS IS' basis and without warranty. In no event shall Domain Technologies be liable for incidental or consequential damages arising from the use of this software. This disclaimer of warranty extends to LICENSEE, to LICENSEE's customers or users of products and is in lieu of all warranties whether expressed, implied, or statutory, including implied warranties of merchantability or fitness for a particular purpose. Domain Technologies does not warrant that software furnished hereunder is free of infringement of any third party patents, copyrights, or trade secrets.

TABLE OF CONTENTS

CHAPTER 1 - General Information	5
1.1 - Audio4-USB front view.	5
1.2 - Audio4-USB circuitry.	5
1.3 - Audio4-USB block diagram	5
1.4 - Basic parameters	5
1.5 - System requirements	6
1.6 - Software Installation	6
1.7 - Sample application software.	6
1.8 - User application operation.	7
1.9 - Description of the menu entries	7
CHAPTER 2 - Circuit Diagrams	11
CHAPTER 3 - Audio4-USB PLD equations	17

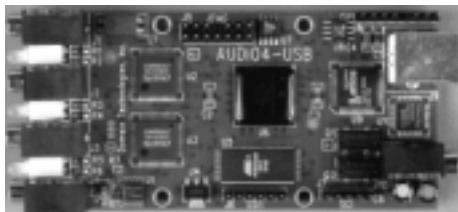
CHAPTER 1 - General Information

1.1 - Audio4-USB front view

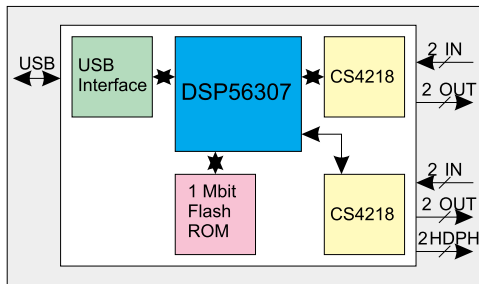


1

1.2 - Audio4-USB circuitry



1.3 - Audio4-USB block diagram



1.4 - Basic parameters

Processor	On-chip RAM	Flash ROM	A/D, D/A	Dimensions	Power	Interface
DSP56307, 100 MHz	64K x 24	1 Mbit	2 x CS4218	1" x 4" x 2.5"	400 mA	USB 12 MHz

1.5 - System requirements

System will work only with the Windows 98 and Windows 2000 operating system. Windows 95 OSRs are not supported.

1.6 - Software Installation

After connecting the Audio4-USB hardware to the USB bus, system will be automatically detected by the host PC. The dialog box will query about the location of the device driver. Driver for the Audio4-USB device is located on the CD ROM, in the directory:

1

```
X:\Usb\Win2000          for Windows 2000
X:\Usb\Win98           for Windows 98
```

After device driver is installed, BoxView software can be installed with the setup.exe from the directory:

```
X:\Usb\BoxView\disk1
```

1.7 - Sample application software

The sample user software (PC side) is located in the directory:

```
X:\Usb\Audio4
```

This is a windows console application, providing basic control for the Audio codecs, and play/record of 4 channel of data. The sources for the DSP side of the sample application are in the directory:

```
X:\Usb\A4uCodec
```

1.8 - User application operation

The user interface for the PC side of the test application looks as follow:

```
Audio4-USB Test rev. 1.02
Audio4-USEm DLL Win 32-bit (dbg)
A4USB32 rev. 1.07.46 (dbg)
  1 - Open USB1          G - Run Target
  2 - Open USB2          H - Halt Target
  3 - Open USB3          V - Get Version
  4 - Open USB4          N - Get Control
  C - Close USB          U - Increase Volume
  Q - Exit test          D - Decrease Volume
  P - Play test.bin      L - Load audio307.cld
  R - Record test.bin    K - Read Pointers
  S - Stop operation     Z - Increase Sample Rate
  A - PlayLoop           X - Decrease Sample Rate
  T - Toggle display
```

1

This is a Win32 console application.

Options are selected by pressing the single key on the keyboard, as displayed in the menu.

Basic device initialization should be executed as follows:

1 H L G (Open USB1, Halt device, Load code, Start executing the code)

After that right LED should start blinking (red/green/off)

1.9 - Description of the menu entries

- 1, 2, 3, 4 allow to open the connection to the specified USB device number. If only one USB device is installed, only option 1 can be used. This is required for all other options to operate.
- C closes interface to the USB device
- Q closes interface if it is opened, and terminates test application
- G starts execution of the user code. If the target device is executing the code, left LED is red.
- H stops execution of the user code. If the target device is halted, left LED is green
- V displays version of the user code, user code needs to be running.

1

- N displays current value of the control word for the CS4218 codec. User code needs to be executing.
- U, D increases/decreases volume in both channels. This will read back the control value, change output gain control and send value to the target
- L loads user application to the target. Target needs to be in halt mode (left LED is green)
- Z,X read current value used for the sampling frequency control, changes value for the next available frequency, and sends request to change frequency control bits. Sampling frequency is reflected by frequency of the right LED.
- K reads current values of pointers used for play/record operation
- R starts recording data to the file "test.bin" Data is stored in the 16-bit linear format, 4 channel packets.
- P plays back previously recorded "test.bin" file
- Sincerely stops play/record operations.
- A plays file "test.bin" in a loop
- T toggles display of the play/record pointers during play/record.

After device initialization, target device serial number will be displayed:

```
A4USB32 rev. 1.07.46 (dbg) Audio4-USB S/N: 010003
```

Also the status will be shown in the status area:

```
USB1, Handle 0
```

When the record operation is active, current value of the "record" pointers will be displayed, with the number of words transferred from the target to the PC. The "Err" field counts number of words lost, because of the slow USB transfer. If the value is non-zero, sampling rate needs to be decreased.

```
Record 0022EC..0050AC (001240) Err: 000000
```

After termination of the record operation, transfer history for the for the terminated operation:

```

Sampling rate: 32000 Hz
Recorded 0x103250 bytes in 4.187 s (Max 0x48D8 bytes)
001EA0 002520 002210 002410 0022C8 002428 0022B8 002328
    10    13    12    13    11    12    12    11
    8     6     7     7     7     7     8     7
    17    16    15    17    17    16    16    16
000000 000000 000000 000000 000000 000000 000000 000000
    
```

First row of numbers displays count of words transferred.

Second row shows time in milliseconds from end of last service.

Third row is a time for retrieving the pointers value

Fourth row shows time for data transfer

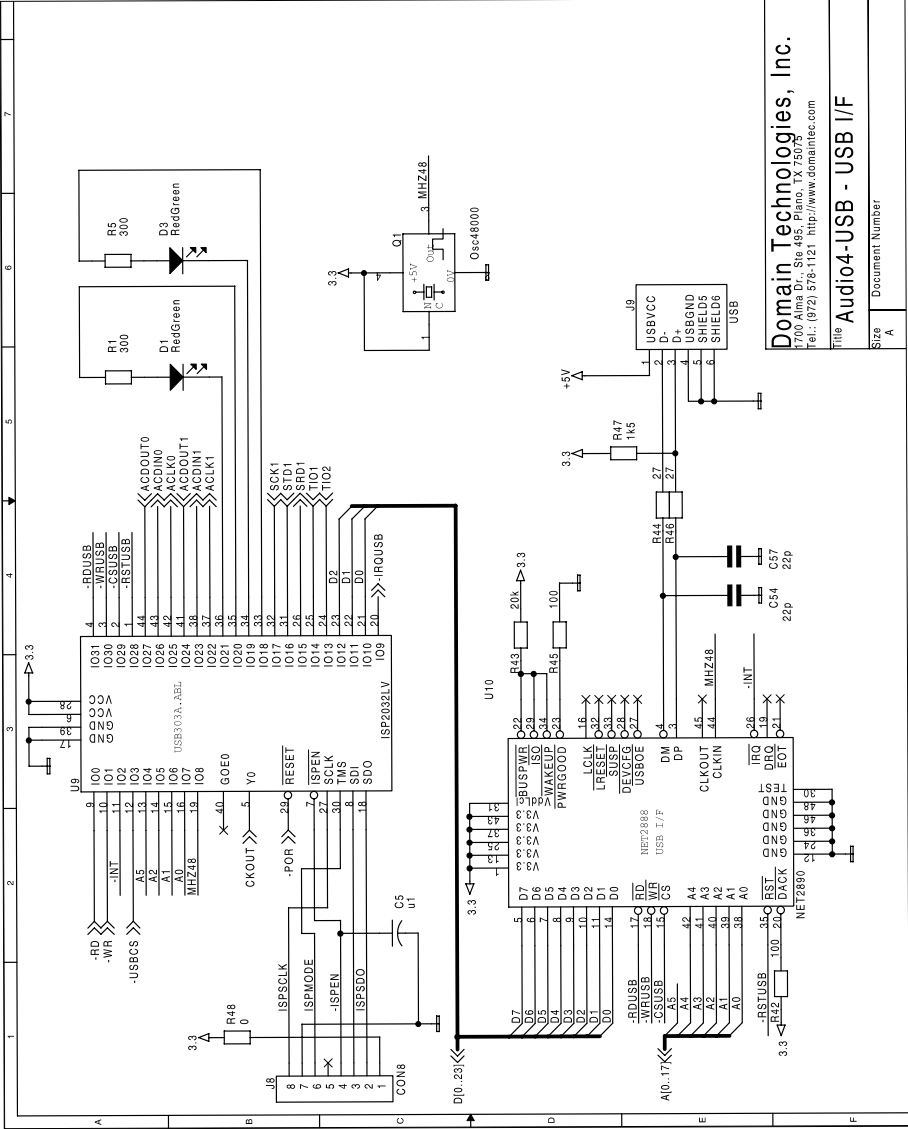
Fifth row shows current value of the error counter.

The same status is displayed for the playback operation.

1

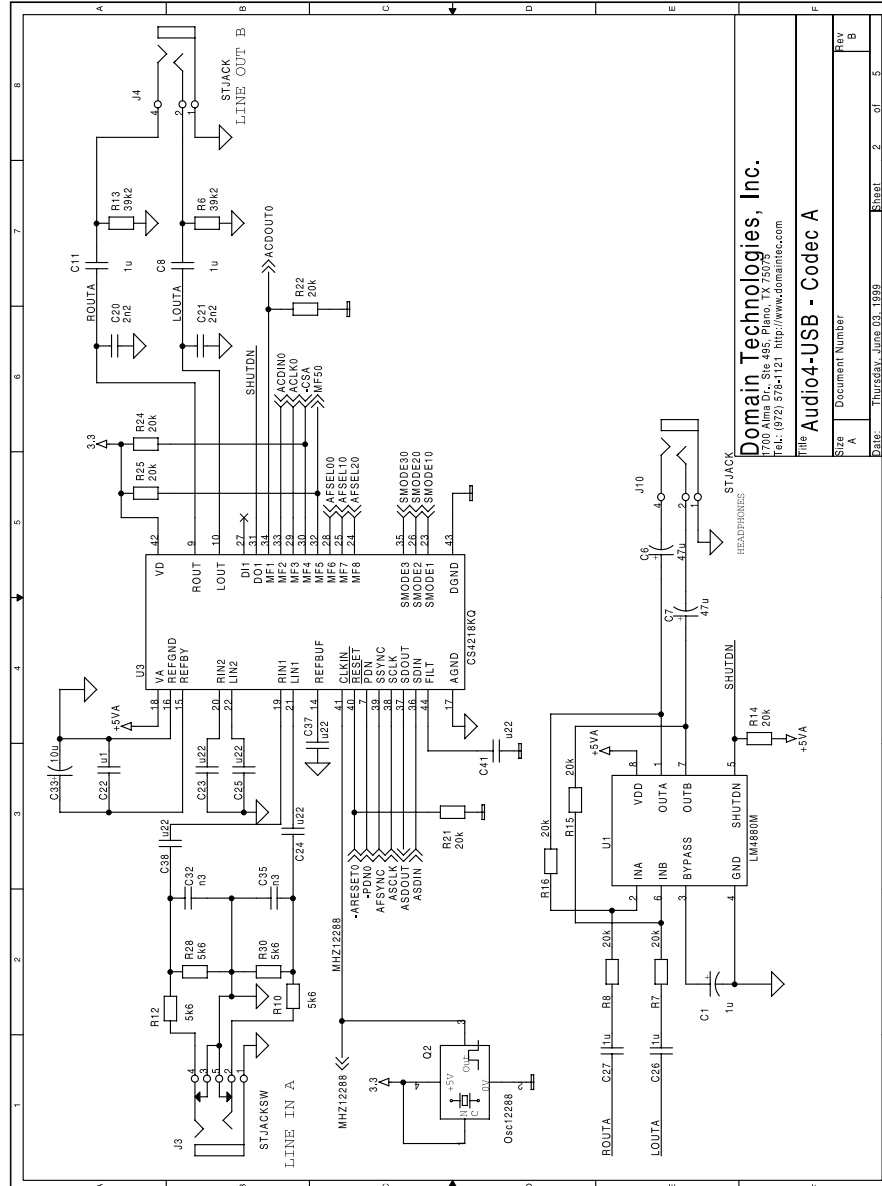
1

CHAPTER 2 - Circuit Diagrams



Domain Technologies, Inc.
 1700 Alma Dr., Ste 495, Plano, TX 75075
 Tel.: (972) 576-1121, http://www.domaintec.com

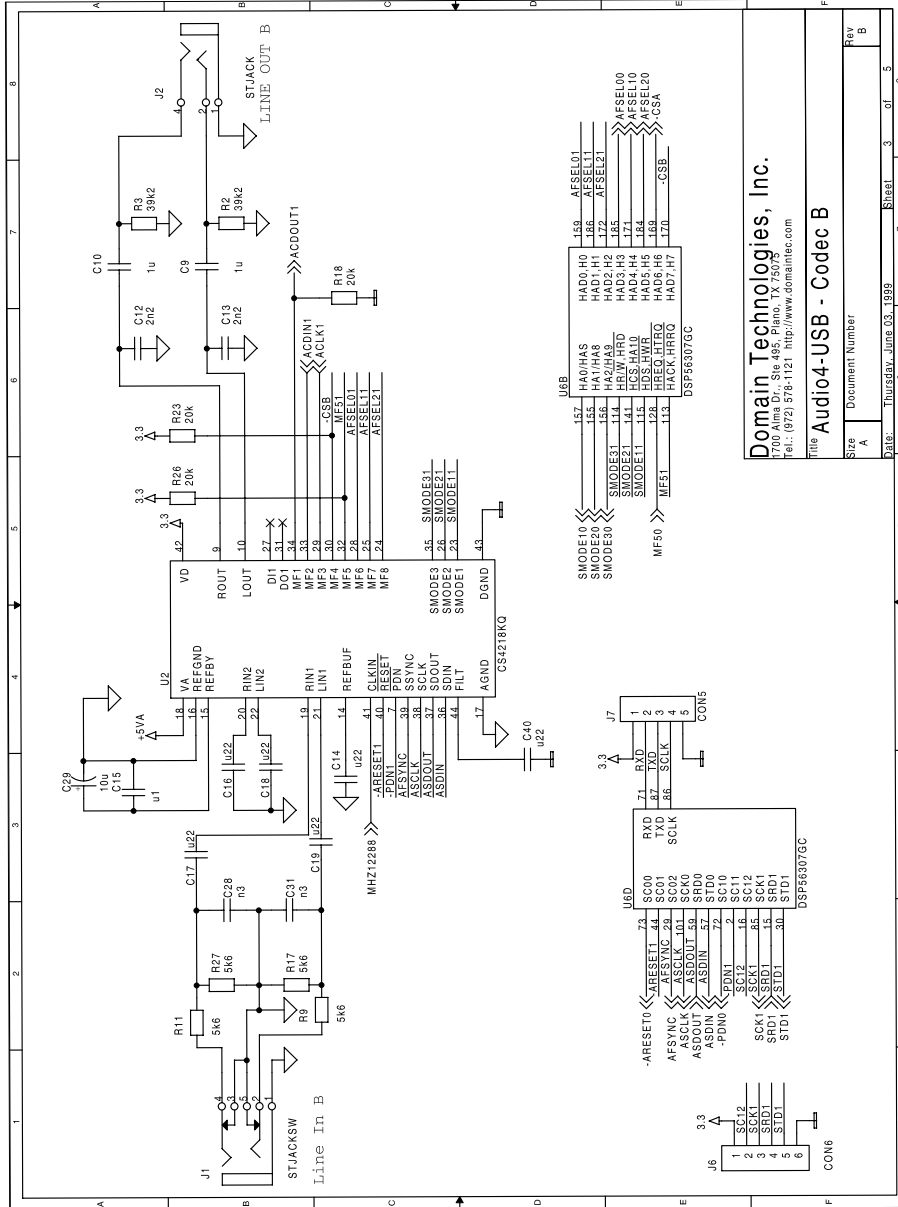
Title Audio4-USB - USB I/F
 Size A
 Document Number



Domain Technologies, Inc.
 1700 Alma Dr., Ste 495, Plano, TX 75075
 Tel: (972) 578-1121 <http://www.domaintec.com>

Title: Audio4-USB - Codec A

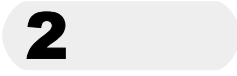
Size	Document Number
A	
Date:	Thursday, June 03, 1999
Sheet:	2 of 5
Rev	B

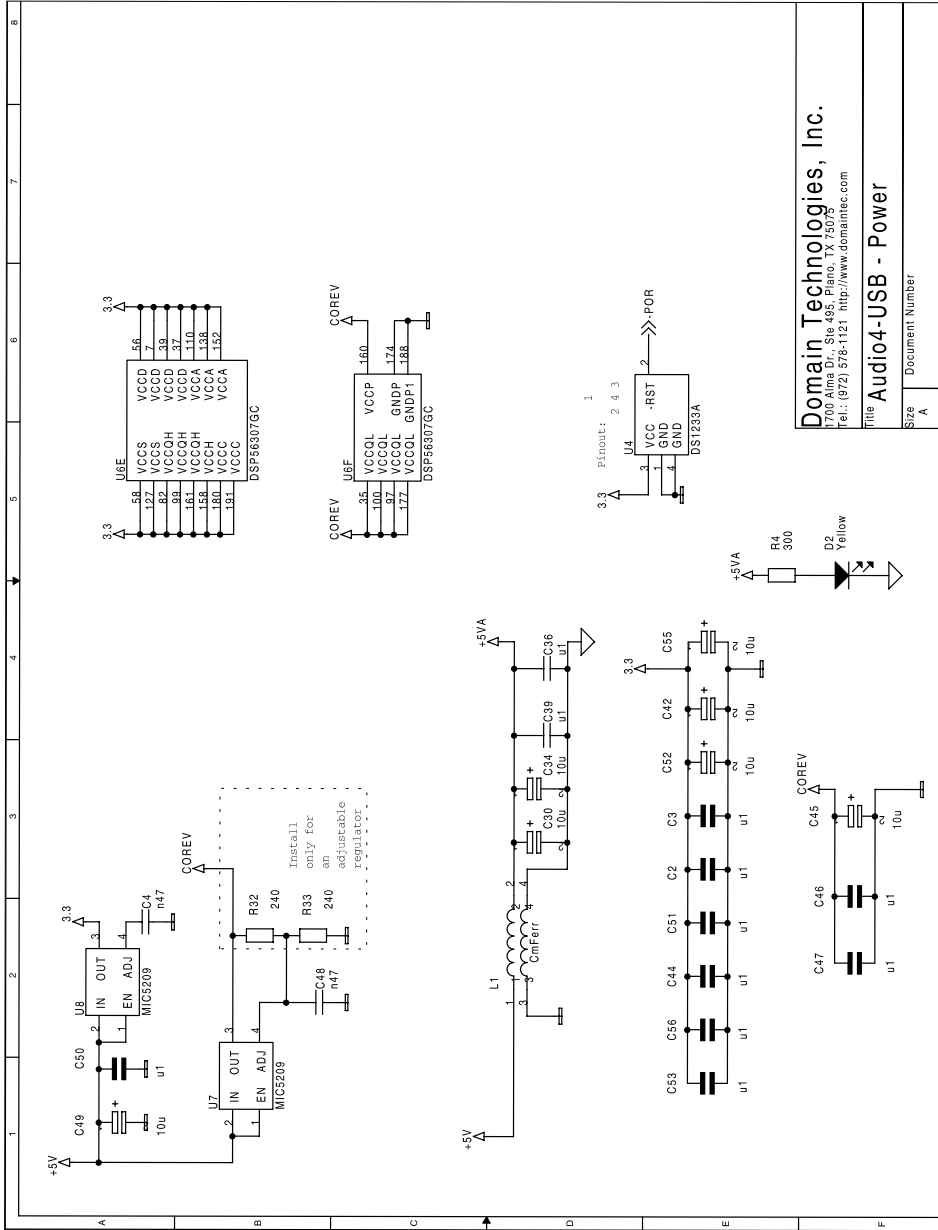


Domain Technologies, Inc.
 1700 Alma Dr., Ste 492 Plano, TX 75075
 Tel.: (972) 578-1121 <http://www.domaintec.com>

Title: Audio4-USB - Codec B

Size	Document Number
A	
Date:	Thursday, June 03, 1999
Sheet	3 of 5





Domain Technologies, Inc.
 1700 Alma Dr., Ste 495, Plano, TX 75075
 Tel.: (972) 578-1121 <http://www.domaintec.com>

Title: Audio4-USB - Power
 Size: A
 Document Number:

2

CHAPTER 3 - Audio4-USB PLD equations

```

MODULE USB307
TITLE 'USB307'
`Inputs
  !RD, !WR, !NETIRQ, !USBCS      pin 9,10,11,12;
  A5, A2, A1, A0                pin 13,14,15,16;
  CKOUT , TIO2, TIO1            pin 19,24,25;
  STD1, SCK1                    pin 31,32;
  ACDOU0, ACDOU1                pin 44,41;
`Outputs
  !IRQB                          pin 20;
  !SLRD                           pin 4;
  !SLWR                            pin 3;
  !SLCS                            pin 2;
  LED2A,LED2C                     pin 33,34   istype `reg,buffer';
  LED1A,LED1C                     pin 35,36   istype `reg,buffer';
  !RSTUSB                          pin 1       istype `reg,buffer';
  ACLK0                            pin 42;
  ACDIN0                           pin 43;
  ACLK1                            pin 37;
  ACDIN1                           pin 38;
`Data I/O
  D0..D2                          pin 21..23;
  SRD1                             pin 26;

`Outputs that don't leave this logic block (buried nodes)
  [STATE8..STATE0]   node istype `reg' ;
  [LEDMODE2..LEDMODE0]   node istype `reg';
  [DATAMODE2..DATAMODE0] node istype `reg';
  [DATA2..DATA0]       node istype `reg';
`Set declarations
  SL11CS      = USBCS & !A5;           "Address decode for USB device.
  LATCH_LED0  = USBCS & A5 & WR & !A2 & !A1 & !A0; "LED0   (y:ffffe0)
  LATCH_LED1  = USBCS & A5 & WR & !A2 & !A1 & A0;  "LED1   (e1)
  LATCH_LED_MODE = USBCS & A5 & WR & !A2 & A1 & !A0; "LED mode (e2)
  LATCH_DATA_MODE = USBCS & A5 & WR & !A2 & A1 & A0; "Dig mode (e3)
  LATCH_DATA    = USBCS & A5 & WR & A2 & !A1 & !A0; "Wr ACDIN/CLK (e4)
  READ_DATA     = USBCS & A5 & RD & A2 & !A1 & !A0; "Rd ACDOU   (e4)
` LED mode of the operation
` D2 D1 D0 mode
` 0 x 0 LED0 from latch
` 0 x 1 LED0 from TIO1/TIO2

```

```

" 0 0 x    LED1 from latch
" 0 1 x    LED1 from TIO1/TIO2
" 1 x 0    LED0 from TIO1/GND
" 1 x 1    LED0 from TIO2/GND
" 1 0 x    LED1 from TIO1/GND
" 1 1 x    LED1 from TIO2/GND

" Serial port mode of the operation
" D2 D1 D0  mode
" 0 0 0    Latch data to/from device 0
" 0 0 1    Latch data to/from device 1
" 0 1 0    ESS11 data to/from device 0
" 0 1 1    ESS11 data to/from device 1
" 1 0 0    Latch data to device 0 & 1
" 1 0 1    ESS11 data to device 0 & 1
" 1 1 0    Remove reset from USB - logic high
" 1 1 1    Assert reset to USB - logic low

USBSTATE=[STATE8..STATE0];
Equations
  IRQB      = NETIRQ;          "generate interrupt to DSP if USB signals irq
  SLCS = SLRD.PIN # SLWR.PIN;  "generate chip select on address decode
  SLRD = RD & SL11CS & STATE1 & !STATE8; "generate rd signal to USB
  SLWR = WR & SL11CS & STATE1 & !STATE5;"wr on clk 2..3 for 42ns strobe

"The read & write strobe to the NET2x USB part must be delayed from the
"beginning of the dsp write to meet the data & address setup timing
"requirements. The write strobe must be terminated to the NET2x prior to
"the dsp releasing it's write strobe to meet data hold times.

"Define a counter to implement the read & write strobe timing. The counter
"stays reset until a valid address decode to the NET2x USB device.
"The clock is ~60 MHz or about 16 ns per clock.
USBSTATE.AR = !RD & !WR; "counter stays reset until valid address decode
STATE0      := 1;
STATE1      := STATE0;
STATE2      := STATE1;
STATE3      := STATE2;
STATE4      := STATE3;
STATE5      := STATE4;

```

3

```
STATE6 := STATE5;
STATE7 := STATE6;
STATE8 := STATE7;
STATE0.C = CKOUT;      "Clock on rising edge of global clock input pin
STATE1.C = !CKOUT;     "Clock on falling edge of global clock input pin
STATE2.C = CKOUT;
STATE3.C = !CKOUT;
STATE4.C = CKOUT;
STATE5.C = !CKOUT;
STATE6.C = CKOUT;
STATE7.C = !CKOUT;
STATE8.C = CKOUT;

LED2C := D1;
LED2A := D0;
LED1C := D1;
LED1A := D0;
LED2C.C = !LATCH_LED1;
LED2A.C = !LATCH_LED1;
LED1C.C = !LATCH_LED0;
LED1A.C = !LATCH_LED0;

LEDMODE2 := D2;
LEDMODE1 := D1;
LEDMODE0 := D0;
LEDMODE2.c = !LATCH_LED_MODE;
LEDMODE1.c = !LATCH_LED_MODE;
LEDMODE0.c = !LATCH_LED_MODE;

DATAMODE2 := D2;
DATAMODE1 := D1;
DATAMODE0 := D0;
DATAMODE2.c = !LATCH_DATA_MODE;
DATAMODE1.c = !LATCH_DATA_MODE;
DATAMODE0.c = !LATCH_DATA_MODE;

DATA2 := D2;
DATA1 := D1;
DATA0 := D0;
DATA2.c = !LATCH_DATA;
DATA1.c = !LATCH_DATA;
DATA0.c = !LATCH_DATA;
```

```
ACLK0 = DATA1 & !DATAMODE0 & !DATAMODE1 & DATAMODE2 //dm=4
# DATA1 & !DATAMODE0 & !DATAMODE1 & !DATAMODE2 //dm=0
# SCK1 & DATAMODE0 & !DATAMODE1 & DATAMODE2 //dm=5
# SCK1 & !DATAMODE0 & DATAMODE1 & !DATAMODE2; //dm=2

ACDIN0 = DATA0 & !DATAMODE0 & !DATAMODE1 & DATAMODE2 //dm=4
# DATA0 & !DATAMODE0 & !DATAMODE1 & !DATAMODE2 //dm=0
# STD1 & DATAMODE0 & !DATAMODE1 & DATAMODE2 //dm=5
# STD1 & !DATAMODE0 & DATAMODE1 & !DATAMODE2; //dm=2

ACLK1 = DATA1 & !DATAMODE0 & !DATAMODE1 & DATAMODE2 //dm=4
# DATA1 & DATAMODE0 & !DATAMODE1 & !DATAMODE2 //dm=1
# SCK1 & DATAMODE0 & !DATAMODE1 & DATAMODE2 //dm=5
# SCK1 & DATAMODE0 & DATAMODE1 & !DATAMODE2; //dm=3

ACDIN1 = DATA0 & !DATAMODE0 & !DATAMODE1 & DATAMODE2 //dm=4
```

3