USS-725

Data Sheet

(For Hardware Revision D)

Revision 0.94

December 6, 2000

Table of Contents

Table of Tables

Table of Figures

Pin Information

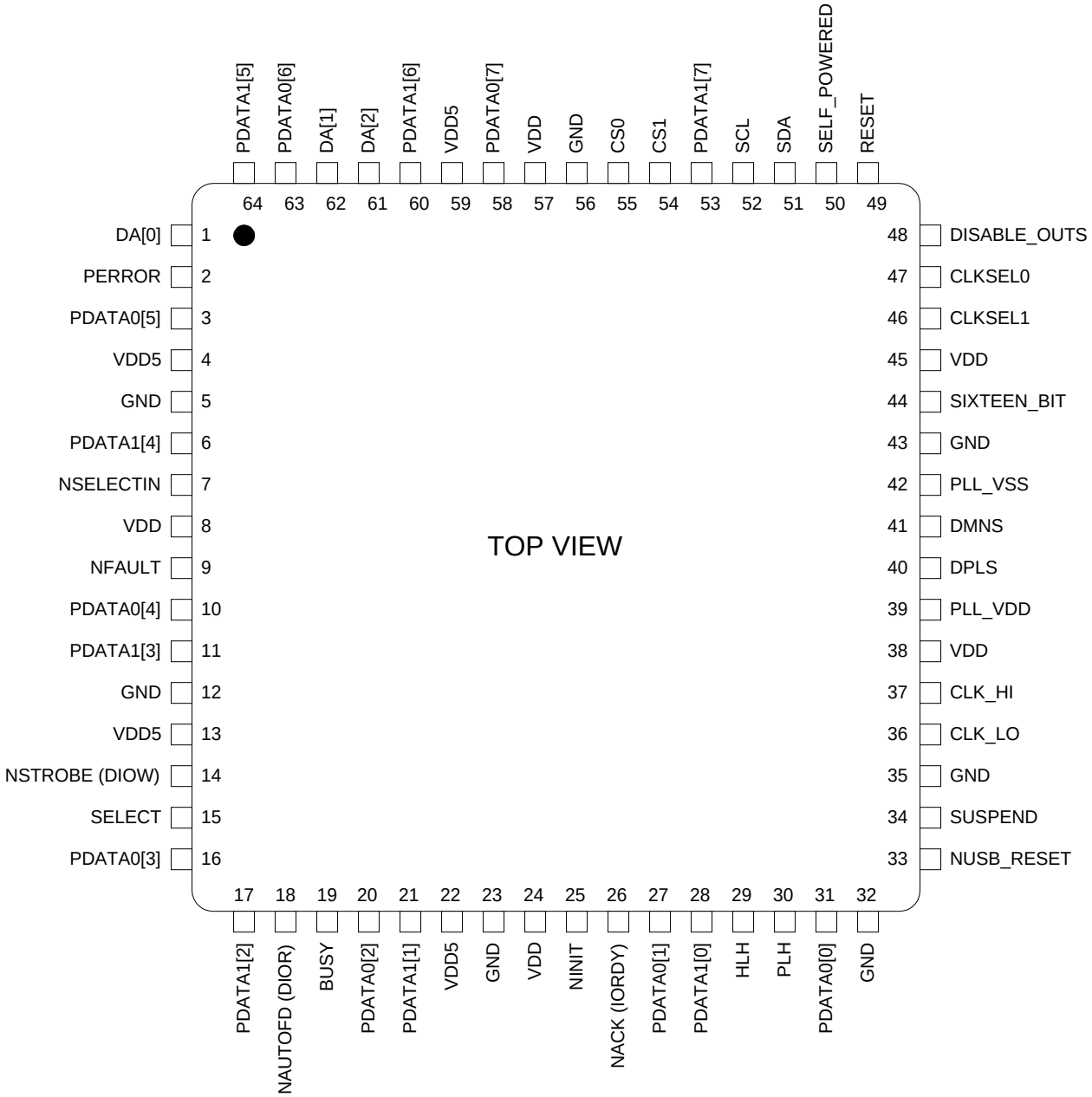

Figure 1 – Pin Layout

Pin	Symbol	Type	Name/Description
1	DA[0]	O	ATA address bit DA(0)
2	PERROR	I	Parallel port PError signal
3	PDATA0[5]	I/O	Parallel port data bus bit 5
4	VDD5	P	5 V (or 3.3 V) digital power supply
5	GND	P	Digital ground
6	PDATA1[4]	I/O	High byte data bus bit 4
7	NSELECTIN	O	Parallel port nSelectIn signal (active low)
8	VDD	P	3.3 V digital power supply
9	NFAULT	I	Parallel port nFault signal (active low)
10	PDATA0[4]	I/O	Parallel port data bus bit 4
11	PDATA1[3]	I/O	High byte data bus bit 3
12	GND	P	Digital ground
13	VDD5	P	5 V (or 3.3 V) digital power supply
14	NSTROBE (DIOW)	O	Parallel port nStrobe signal (active low); DIOW- in ATA mode
15	SELECT	I	Parallel port Select signal
16	PDATA0[3]	I/O	Parallel port data bus bit 3
17	PDATA1[2]	I/O	High byte data bus bit 2
18	NAUTOFD (DIOR)	O	Parallel port nAutoFd signal (active low); DIOR- in ATA mode
19	BUSY	I	Parallel port Busy signal
20	PDATA0[2]	I/O	Parallel port data bus bit 2
21	PDATA1[1]	I/O	High byte data bus bit 1
22	VDD5	P	5 V (or 3.3 V) digital power supply
23	GND	P	Digital ground
24	VDD	P	3.3 V digital power supply
25	NINIT	O	Parallel port nInit signal (active low)
26	NACK (IORDY)	I	Parallel port nAck signal (active low); IORDY in ATA mode
27	PDATA0[1]	I/O	Parallel port data bus bit 1
28	PDATA1[0]	I/O	High byte data bus bit 0
29	HLH	O	Parallel port Host Logic High (HLH) signal
30	PLH	I	Parallel port Peripheral Logic High (PLH) signal
31	PDATA0[0]	I/O	Parallel port data bus bit 0
32	GND	I	Digital ground
33	NUSB_RESET	O	USB Reset (active low) Indicates that there is a bus reset condition on the USB bus.
34	SUSPEND	O	Suspend Indicates USS-725 is in suspend (due to inactivity on the USB bus).
35	GND	P	Digital ground
36	CLK_LO	I	Clock Low (crystal or 3.3 V CMOS input)
37	CLK_HI	O	Clock High (crystal or no connection)
38	VDD	P	3.3 V digital power supply
39	PLL_VDD	P	3.3 V analog power supply for PLL
40	DPLS	I/O	USB DPLS (Differential +) signal
41	DMNS	I/O	USB DMNS (Differential -) signal
42	PLL_VSS	P	Analog ground for PLL
43	GND	P	Digital ground
44	SIXTEEN_BIT	I	Enables PDATA1[7:0] and ATA mode operation
45	VDD	P	3.3 V digital power supply

46	CLKSEL1	I	Clock Select Connect both CLKSEL1 and CLKSEL0 to GND to use the USS-725 with an external 12 MHz crystal. Connect both CLKSEL1 and CLKSEL0 to a high level (3.3 V) to use an external 48MHz clock.
47	CLKSEL0	I	
48	DISABLE_OUTS	I	Disable Parallel Port Outputs Disables (sets into a high impedance state) PDATA0[7:0], PDATA1[7:0], HLH, NINIT, NAUTOFD (DIOR), NSTROBE (DIOW), NSELECTIN, CS1, CS0, and DA[2:0]. The Disable Outputs feature may only be used when CLKSEL1 and CLKSEL0 are low.
49	RESET	I	This signal will cause a chip reset when driven to a high level.
50	SELF_POWERED	I/O	Causes USS-725 to report a self-powered configuration when high and a bus-powered configuration when low.
51	SDA	I/O	Serial EEPROM data line
52	SCL	O	EEPROM serial bus clock
53	PDATA1[7]	I/O	High byte data bus bit 7
54	CS1	O	ATA CS1- signal
55	CS0	O	ATA CS0- signal
56	GND	P	Digital ground
57	VDD	P	3.3 V digital power supply
58	PDATA0[7]	I/O	Parallel port data bus bit 7
59	VDD5	P	5 V (or 3.3 V) digital power supply
60	PDATA1[6]	I/O	High byte data bus bit 6
61	DA[2]	O	ATA address bit DA(2)
62	DA[1]	O	ATA address bit DA(1)
63	PDATA0[6]	I/O	Parallel port data bus bit 6
64	PDATA1[5]	I/O	High byte data bus bit 5

Table 1 – Pin Descriptions

Overview

The USS-725 creates a bridge between one USB port and one parallel port. The USS-725 parallel port provides automatic support for IEEE 1284. It also includes a register mode (with a sequencer for automation of register operations) which has hardware support for IEEE 1284 Compatibility, ECP (with or without automatic hardware-based RLE compression and decompression), and EPP modes. The parallel port also provides hardware support for the ATA PIO mode protocol, for connecting to ATA-based mass storage devices.

The USS-725 contains an integrated USB transceiver, integrated IEEE 1284 buffers, an onboard oscillator, PLL, and reset block for single-chip operation.

In use, the USB port of the USS-725 is connected via a USB cable to a host computer via the downstream port of a USB hub. Host software sends commands and data to the USS-725 and receives status and data from the USS-725 using the USB protocol.

The USS-725 parallel port is connected to a peripheral device. If the peripheral is IEEE 1284 compatible, then the IEEE 1284 features and communication modes can be used. The USS-725 provides automatic support for IEEE 1284, and also provides for manual operation of the parallel port with hardware assist for IEEE 1284 and ATA (hard disk) protocols. A sequencer is provided to automate register operations and data transfer operations in manual mode.

USB Port

The USB port on the USS-725 is electrically and logically compliant with the *Universal Serial Bus Specification Revision 1.1*.

Descriptors

Supported Descriptors

- *Device*
- *Configuration*
- *Interface*

  The USS-725 device supports one interface with three alternate settings.
  Interface 0, alternate settings 0 and 1 are compliant with the *Universal Serial Bus Device Class Definition for Printing Devices, Version 1.0*.
  Interface 0, alternate setting 2 is a vendor-specific interface.
- *Endpoint*

  The USS-725 supports the following endpoints:
  - Control endpoint. Accessible as endpoint 0 in all three alternate settings.
  - Bulk Out endpoint. Accessible as endpoint 1 in all three alternate interface settings.
  - Bulk In endpoint. Accessible as endpoint 2 in alternate interface settings 1 and 2.
- *String*

  The USS-725 supports arbitrary string descriptors. For more information on strings, **see String Descriptors** on page 5.

Descriptor Locations

Descriptor data is supplied from an external ROM or other memory device. The USS-725 provides support for the 24LC01-16 family EEPROM interface. The USS-725 also contains an on-board set of device,

configuration, interface, and endpoint descriptors. Retrieval of the on-board descriptors will occur if no external descriptor data is supplied. These on-board descriptors may be used during development, prototyping, and manufacturing. For actual products using the USS-725, however, real descriptors must be supplied via an external device to provide correct device-specific information.

String Descriptors

The USS-725 supports USB string descriptors. These strings can be referenced by standard descriptors (e.g. a manufacturer name string indexed by the iManufacturer field in the Device Descriptor), or can be independent strings. These are stored in the external I^2C memory device in the space following that used for standard descriptors.

Table 15 shows the format of string index 0. The format and location of this string are fixed. String index 0 may contain the LANGID of exactly one language, as the USS-725 supports only a single language. Microsoft defines the LANGID codes for Windows, as described in *Developing International Software for Windows 95 and Windows NT*, Nadine Kano, Microsoft Press, Redmond, Washington. Note that the LANGID code for English is 0x0409.

In Table 16, the description for string index 0x9 is shown as an example of how to format string descriptors in general. All strings other than string index 0 are addressed by the value specified in the Descriptor Index field in the Get_Descriptor command, multiplied by four. For example, a Get_Descriptor command requesting the String Descriptor at index 0x1A would retrieve the string starting at address 0x68 in the external I^2C memory device.

ROM Contents and Formats

Internal (on-board) ROM addresses, the contents of those locations, the associated field name, and an indication whether the value may be overridden using an external EEPROM, are shown in Table 3 through Table 8. External EEPROM addresses for values that may be overridden are shown in Table 10 through Table 16. Note that the format for string index 0 is fixed, and can only include one language.

Externally supplied descriptors **must** be used for all products. The internal ROM is provided only as a convenience for prototyping and manufacturing activities. The internal ROM supplies enough information to the USB host so that the USS-725 will enumerate even when connected to an unprogrammed EEPROM.

The USS-725 supports the I^2C "fast mode" interface, as found in the 24(L)C01-16 EEPROM family interface. That family allows for up to 2 kB of external storage, which may be used to store either descriptors or sequencer subroutines (see **Sequencer** on page 25). Note that if a 24(L)C04/08/16 part is used, no descriptor is allowed to span multiple pages within the EEPROM.

USS-725 Configuration Byte

The USS-725 Configuration Byte is located at the first address (address 0) in the external I^2C memory device. This byte, shown in Table 2 and referenced in Table 9, is read at power-up. Its value is used to control two parameters in the USS-725.

USS-725 Configuration Byte								
Bit	7	6	5	4	3	2	1	0
7:2	Reserved; set to '0'.							
1	This bit controls whether the parallel port digital filters are turned on or off by default. '0' selects filters off, and '1' selects filters on. See **Filter Bypass Mode** on page 29.							
0	This bit controls whether the parallel port buffers always operate with totem-pole drive or whether they are normally open-drain, with a 750 ns period of totem-pole drive upon a signal transition on the parallel port ("High Drive" capability). '0' selects totem-pole drive, and '1' selects open-drain with High Drive. See **High Drive Mode** on page 29.							

Table 2 – USS-725 Configuration Byte

Internal ROM Format

Device Descriptor
There is only one device descriptor for each USB device. This descriptor contains the definitions of the device class and device subclass, among other things.

Internal Address	Field Name	Internal Value	External Override	Description
0x0	bLength	0x12		Size of this descriptor in bytes.
0x1	bDescriptorType	0x01		Device descriptor type.
0x2	bcdUSB (LSB)	0x10		USB Specification release number in BCD.
0x3	bcdUSB (MSB)	0x01		
0x4	bDeviceClass	0x00	Yes	0x00 indicates "interface specific." See Table 6 through Table 8.
0x5	bDeviceSubClass	0x00	Yes	0x00 indicates "interface specific." See Table 6 through Table 8.
0x6	bDeviceProtocol	0x00	Yes	0x00 indicates "interface specific." See Table 6 through Table 8.
0x7	wMaxPacketSize0	0x40		Maximum packet size for endpoint 0 (0x40 = 64 bytes).
0x8	idVendor (LSB)	0xab	Yes	Vendor ID for In-System Design.
0x9	idVendor (MSB)	0x05	Yes	
0xA	idProduct (LSB)	0x01	Yes	Product ID.
0xB	idProduct (MSB)	0x10	Yes	
0xC	bcdDevice (LSB)	0x00	Yes	Device release number in BCD.
0xD	bcdDevice (MSB)	0x04		
0xE	iManufacturer	0x00	Yes	Index of string descriptor describing manufacturer.
0xF	iProduct	0x00	Yes	Index of string descriptor describing this product.
0x10	iSerialNumber	0x00	Yes	Index of string descriptor describing the device serial number.
0x11	bNumConfigurations	0x01		Number of possible configurations.

Table 3 – Onboard Device Descriptor

Configuration Descriptor

The USS-725 has one configuration descriptor (see Table 5). This descriptor has one interface, which has three alternate settings. The three alternate settings and the endpoints that they support are described in Table 4.

Endpoint (packet size)	Interface		
	Alternate Setting 0	**Alternate Setting 1**	**Alternate Setting 2**
Control pipe	Endpoint Number 0 (64 bytes)	Endpoint Number 0 (64 bytes)	Endpoint Number 0 (64 bytes)
Bulk Out pipe	Endpoint Number 1 (64 bytes)	Endpoint Number 1 (64 bytes)	Endpoint Number 1 (64 bytes)
Bulk In pipe	–	Endpoint Number 2 (64 bytes)	Endpoint Number 2 (64 bytes)

Table 4 – Alternate Settings and Supported Endpoints

Internal Address	Field Name	Internal Value	External Override	Description
0x12	bLength	0x09		Size of this descriptor in bytes.
0x13	bdescriptorType	0x02		Configuration descriptor type.
0x14	bTotalLength (LSB)	0x47		Number of bytes in this configuration. This includes the configuration descriptor plus all the interface and endpoint descriptors.
0x15	bTotalLength (MSB)	0x00		
0x16	bNumInterfaces	0x01		The USS-725 has one interface.
0x17	bConfigurationValue	0x01		Value to use as an argument to Set Configuration to select this configuration.
0x18	iConfiguration	0x00	Yes	Index of string descriptor describing this configuration.
0x19	bmAttributes	0x80	Yes (upper two bits only)	Configuration characteristics: Bit Description On-board default 7 Bus-powered. '1' 6 Self-powered. '0' 5 Remote wake-up. '0' 4-0 Reserved, set to 0. '0'
0x1A	maxPower	0x31	Yes	Maximum power consumption of this configuration. Units are mA*2. 0x31 = 98 mA.

Table 5 – Onboard Configuration Descriptor

Interface 0, Alternate Setting 0 (I0:A0)

Internal Address	Field Name	Internal Value	External Override	Description
Interface Descriptor I0:A0				
0x1B	blength	0x09		Size of this descriptor in bytes.
0x1C	bDescriptorType	0x04		Interface descriptor type.
0x1D	bInterfaceNumber	0x00		Zero-based value identifying the number of this interface.
0x1E	bAlternateSetting	0x00		Value used to select this alternate interface.
0x1F	bNumEndpoints	0x01		Number of endpoints used by this descriptor.
0x20	bInterfaceClass	0x07	Yes	Interface class (0x07 = Printer Class).
0x21	iInterfaceSubClass	0x01	Yes	Interface subclass (0x01 = Printer Subclass).
0x22	bInterfaceProtocol	0x01	Yes	Interface protocol (0x01 = Unidirectional interface).
0x23	iInterface	0x00	Yes	Index of string descriptor describing this interface.
Bulk Out Endpoint Descriptor I0:A0:E1				
0x24	bLength	0x07		Size of this descriptor in bytes.
0x25	bDescriptorType	0x05		Endpoint descriptor type.
0x26	bEndpointAddress	0x01		This is an Out endpoint, endpoint number 1.
0x27	bmAttributes	0x02		This is a Bulk endpoint.
0x28	wMaxPacketSize (LSB)	0x40		Maximum data transfer size (0x0040 = 64 bytes).
0x29	wMaxPacketSize (MSB)	0x00		
0x2A	bInterval	0x00		Does not apply to Bulk endpoints.

Table 6 – Interface 0, Alternate Setting 0

Interface 0, Alternate Setting 1 (I0:A1)

Internal Address	Field Name	Internal Value	External Override	Description
Interface Descriptor I0:A1				
0x2B	bLength	0x09		Size of this descriptor in bytes.
0x2C	bDescriptorType	0x04		Interface descriptor type.
0x2D	bInterfaceNumber	0x00		Zero-based value identifying the number of this interface.
0x2E	bAlternateSetting	0x01		Value used to select this alternate interface.
0x2F	bNumEndpoints	0x02		Number of endpoints used by this descriptor.
0x30	bInterfaceClass	0x07	Yes	Interface class (0x07 = Printer Class).
0x31	iInterfaceSubClass	0x01	Yes	Interface subclass (0x01 = Printer Subclass).
0x32	bInterfaceProtocol	0x02	Yes	Interface protocol (0x02 = Bidirectional interface).
0x33	iInterface	0x00	Yes	Index of string descriptor describing this interface.
Bulk Out Endpoint Descriptor, I0:A1:E1				
0x34	bLength	0x07		Size of this descriptor in bytes.
0x35	bDescriptorType	0x05		Endpoint descriptor type.
0x36	bEndpointAddress	0x01		This is an Out endpoint, endpoint number 1.
0x37	bmAttributes	0x02		This is a Bulk endpoint.
0x38	wMaxPacketSize (LSB)	0x40		Maximum data transfer size (0x0040 = 64 bytes).
0x39	wMaxPacketSize (MSB)	0x00		
0x3A	bInterval	0x00		Does not apply to Bulk endpoints.
Bulk In Endpoint Descriptor, I0:A1:E2				
0x3B	bLength	0x07		Size of this descriptor in bytes.
0x3C	bDescriptorType	0x05		Endpoint descriptor type.
0x3D	bEndpointAddress	0x82		This is an In endpoint, endpoint number 2.
0x3E	bmAttributes	0x02		This is a Bulk endpoint.
0x3F	wMaxPacketSize (LSB)	0x40		Maximum data transfer size (0x0040 = 64 bytes).
0x40	wMaxPacketSize (MSB)	0x00		
0x41	bInterval	0x00		Does not apply to Bulk endpoints.

Table 7 – Interface 0, Alternate Setting 1

Interface 0, Alternate Setting 2 (I0:A2)

Internal Address	Field Name	Internal Value	External Override	Description
Interface Descriptor I0:A2				
0x42	bLength	0x09		Size of this descriptor in bytes.
0x43	bDescriptorType	0x04		Interface descriptor type.
0x44	bInterfaceNumber	0x00		Zero-based value identifying the number of this interface.
0x45	bAlternateSetting	0x02		Value used to select this alternate interface.
0x46	bNumEndpoints	0x02		Number of endpoints used by this descriptor.
0x47	bInterfaceClass	0xFF	Yes	Interface class (0xFF = vendor-specific).
0x48	iInterfaceSubClass	0x00	Yes	Interface subclass (N/A).
0x49	bInterfaceProtocol	0xFF	Yes	Interface protocol (0xFF = vendor-specific).
0x4A	iInterface	0x00	Yes	Index of string descriptor describing this interface.
Bulk Out Endpoint Descriptor, I0:A2:E1				
0x4B	bLength	0x07		Size of this descriptor in bytes.
0x4C	bDescriptorType	0x05		Endpoint descriptor type.
0x4D	bEndpointAddress	0x01		This is an Out endpoint, endpoint number 1.
0x4E	bmAttributes	0x02		This is a Bulk endpoint.
0x4F	wMaxPacketSize (LSB)	0x40		Maximum data transfer size (0x0040 = 64 bytes).
0x50	wMaxPacketSize (MSB)	0x00		
0x51	bInterval	0x00		Does not apply to Bulk endpoints.
Bulk In Endpoint Descriptor, I0:A2:E2				
0x52	bLength	0x07		Size of this descriptor in bytes.
0x53	bDescriptorType	0x05		Endpoint descriptor type.
0x54	bEndpointAddress	0x82		This is an In endpoint, endpoint number 2.
0x55	bmAttributes	0x02		This is a Bulk endpoint.
0x56	wMaxPacketSize (LSB)	0x40		Maximum data transfer size (0x0040 = 64 bytes).
0x57	wMaxPacketSize (MSB)	0x00		
0x58	BInterval	0x00		Does not apply to Bulk endpoints.

Table 8 – Interface 0, Alternate Setting 2

External EEPROM Formatting

EEPROM Address	Field Name
0x00	USS-725 Configuration
0x01	not used

Table 9 – Configuration Byte

EEPROM Address	Field Name
0x02	bDeviceClass
0x03	bDeviceSubClass
0x04	bDeviceProtocol
0x05	idVendor (LSB)
0x06	idVendor (MSB)
0x07	idProduct (LSB)
0x08	idProduct (MSB)
0x09	bcdDevice (LSB)
0x0A	not used
0x0B	iManufacturer
0x0C	iProduct
0x0D	iSerialNumber

Table 10 – Device Descriptor Fields

EEPROM Address	Field Name
0x0E	iConfiguration
0x0F	bmAttribute
0x10	maxPower

Table 11 – Configuration Descriptor Fields

EEPROM Address	Field Name
0x11	bInterfaceClass
0x12	bInterfaceSubClass
0x13	bInterfaceProtocol
0x14	iInterface

Table 12 – Interface 0:Alternate Setting 0

EEPROM Address	Field Name
0x15	bInterfaceClass
0x16	bInterfaceSubClass
0x17	bInterfaceProtocol
0x18	iInterface

Table 13 – Interface 0: Alternate Setting 1

EEPROM Address	Field Name
0x19	bInterfaceClass
0x1A	bInterfaceSubClass
0x1B	bInterfaceProtocol
0x1C	iInterface

Table 14 – Interface 0: Alternate Setting 2

EEPROM Address	Field Name
0x1D	bLength (0x4)
0x1E	bDescriptorType (0x3)
0x1F	LANGID (LSB)
0x20	LANGID (MSB)
0x21 – 0x23	not used
Note: See LANGID table in Microsoft documentation (the code for English is 0x0409)	

Table 15 – String Descriptor Index (0)

EEPROM Address	Field Name
0x24	bLength
0x25	bDescriptor Type
0x26	bString
0x27	bString
0x28	…
Note: Strings are addressed by Index*4.	

Table 16 – String Descriptor Index (0x9)

Pipes

The USS-725 provides three USB pipes: Control, Bulk Out, and Bulk In.

Control Pipe

The Control pipe is the default pipe, used for USB standard requests. Its maximum packet size is 64 bytes. The Control pipe is also used for class-specific and vendor-specific commands that:

- Perform a soft reset of the USS-725.
- Read Printer Class Get_Device_ID data.
- Read Printer Class Port Status data.
- Read and write the USS-725's registers.
- Control the USS-725's sequencer.
- Program data into an external EEPROM.

Bulk In Pipe

The Bulk In pipe is used to read data bytes from the peripheral in Automatic Mode, register-based ECP Mode, and when using certain sequencer instructions. Its maximum packet size is 64 bytes.

Bulk Out Pipe

The Bulk Out pipe is used to send data to the peripheral in Automatic Mode and in register-based Compatibility, EPP, or ECP Modes, and when using certain sequencer instructions. Its maximum packet size is 64 bytes.

Inter-pipe Synchronization

With commands and data going to different pipes, and data being buffered inside the USS-725, the host software must take care to maintain serialization of operations on the peripheral. This can be done by reading the registers to determine the status of the USS-725, either explicitly or using sequencer instructions.

Requests

The USS-725 responds to three different types of request:

- Standard USB device requests
- Printer Class requests
- Vendor-specific requests

Standard Requests

The USS-725 supports all USB standard device requests except the optional Set Descriptor request. These requests, which are described in Chapter 9, Device Framework, of the *USB Specification*, are:

- Clear Feature
- Get Configuration
- Get Descriptor (for information on String Descriptors, see **String Descriptors** on page 5)
- Get Interface
- Get Status
- Set Address
- Set Configuration
- Set Interface
- Set Feature

Printer Class Specific Requests

Printer Class-specific requests supported by the USS-725 are listed in Table 17.

Label	bmRequestType	bRequest	wValue	wIndex	wLength	Data
GET_DEVICE_ID	10100001B	0	Zero	Interface (MSB) and Alternate (LSB) Setting index.	Maximum length to be returned	IEEE 1284 Device ID String
GET_PORT_STATUS	10100001B	1	Zero	Zero-based Interface index.	1	Port Status byte
SOFT_RESET	00100001B	2	Zero	Zero-based Interface index.	Zero	[None]

Table 17 – Printer Class-Specific Requests

GET_DEVICE_ID
This request returns an IEEE 1284 Device ID string. This command is supported in all three alternate interface settings.

The Device ID is a length field followed by a case-sensitive string of ASCII characters. The first two bytes contain the length of the sequence, including the two length bytes. The first byte is most significant; length values of 0x0000, 0x0001, and 0x0002 are reserved.

Following the two length bytes, the sequence is composed of a series of keys and values of the form:

key: value {, value};

repeated for each key. As indicated, each key will have one value, and may have more than one value. The minimum necessary keys are MANUFACTURER, COMMAND SET, and MODEL (case sensitive). These keys may be abbreviated as MFG, CMD, and MDL. Each key (and each value) is a string of characters. Any characters except colon (:), comma (,), and semicolon (;) may be included as part of the key or value string. Any leading or trailing white space in the string is ignored by the parsing program, but still is counted as part of the overall length of the sequence.

For more details, see *IEEE 1284-1994*, Section 7.6.

Note: The USS-725 satisfies this request by reading Device ID data from the attached peripheral. If the peripheral does not support Device ID, the USS-725 will return a zero-length data packet to the host.

GET_PORT_STATUS
This request returns the current status of the printer. Table 18 defines the data returned and describes the format of the status data. This command is supported by the Printer Class-specific alternate interface settings (0 and 1).

Bit	Description
7-6	Reserved, will always read ‘0’
5	Paper error
4	Select
3	Not error: ‘0’ = error, ‘1’ = no error
2-0	Reserved, will always read ‘0’

Table 18 – Get Port Status Data

Note: The USS-725 satisfies this request by reporting the state of the IEEE 1284 PError, Select, and nFault signals driven by the attached peripheral. However, if the interface is in any state other than Compatibility

Mode, these signals lose their normal meanings. If the USS-725 detects that the interface is not in Compatibility Mode, it will return "benign" status of 0x18 in response to a GET_PORT_STATUS request.

SOFT_RESET
This request flushes all buffers and resets the Bulk Out and Bulk In pipes to their default states, and also resets all parallel port hardware and registers to their default state. USB conditions such as address, configuration, stall status, and data toggle status, will be unchanged. The rest of the USS-725 will be in a power-up reset state. This command is supported in all three alternate interface settings.

Vendor Specific Requests

The USS-725 supports the following vendor-specific requests:

Label	bmRequestType	bRequest	wValue	wIndex	wLength	Data
GET_1284_REGISTER	11000000B	3	MSB: register address LSB: ignored	Zero	Maximum length to be returned	Register data
SET_1284_REGISTER	01000000B	4	MSB: register address LSB: Data	Zero	Zero	[None]
GET_SEQ_STATUS	11000000B	5	Zero	Zero	Maximum length to be returned	Sequencer status bytes
SEQ_LOAD	01000000B	6	MSB: 0x00 LSB: LSB of sequence global loop count	MSB: ignored LSB: MSB of sequence global loop count	Sequencer instruction image length	Sequencer instruction image
SEQ_EXECUTE	01000000B	6	MSB: 0x80 LSB: LSB of sequence global loop count	MSB: ignored LSB: MSB of sequence global loop count	Sequencer instruction image length	Sequencer instruction image
SEQ_ABORT	01000000B	7	Zero	Zero	Zero	[None]
LOAD_EEPROM	01000000B	8	External address (lower 11 bits)	Zero	EEPROM image length	EEPROM data image
READ_EEPROM	11000000B	9	External address (lower 11 bits)	Zero	EEPROM image length	EEPROM data image

Table 19 – Vendor Specific Requests

GET_1284_REGISTER
The high byte of the wValue field specifies the address of the register that is to be read. The USS-725 responds by returning the current values in all of the parallel port registers. Thus the address specified by the host is actually only significant if the specification of the address affects the state of the USS-725. The only register addresses for which this is true are the EPP Address and EPP Data Registers. For reads of those registers, the read request will cause the appropriate read cycle to be performed on the parallel port interface, and the resulting data from the peripheral to be returned in the USB request data phase.

SET_1284_REGISTER
The wValue field specifies the address of the parallel port register to be written in the high byte and the value to be written in the low byte. The USS-725 will write the specified data into the designated register. For the EPP Address and EPP Data registers, the write will cause the appropriate write cycle to be performed on the parallel port interface.

For more information on the effect of writing registers, see **Registers** on page 17.

GET_SEQ_STATUS
The wLength field specifies the maximum length of status bytes to be returned. The USS-725 responds by returning the sequencer status bytes, as shown below.

Byte	Name	Description
0	Program Counter	Position of sequence byte currently being executed
1	Main Loop Count High	Upper eight bits of the sequencer global loop count
2	Main Loop Count Low	Lower eight bits of the sequencer global loop count
3	Status / Loop Count High	Bit 7: Sequencer active Bit 6: Sequencer crashed Bit 5: Sequencer executing subroutine Bit 4: '0' Bits 3:0: Upper four bits (19:16) of the instruction loop count
4	Loop Count Middle	Middle eight bits (15:8) of the instruction loop count
5	Loop Count High	Lower eight bits (7:0) of the instruction loop count

Table 20 – GET_SEQ_STATUS Data

For more information on using the sequencer, see **Sequencer** on page 25.

SEQ_LOAD
The low byte of the wValue field specifies the low byte of the sequencer global loop count. The low byte of the wIndex field specifies the high byte of the sequencer global loop count. The global loop count is a zero-based count; setting it to 0 will cause the sequence to be executed once, setting it to 1 will cause the sequence to be executed twice, etc. wLength specifies the length in bytes of the sequencer instruction image (which may be up to 128 bytes long). The data phase of this request consists of sequencer instruction image. The USS-725 loads the sequencer with sequencer instruction image and initiates execution, and immediately completes the status phase of the control request. If the host software needs to verify that the sequence has completed, it must use the GET_SEQ_STATUS request to do so.

This request should only be issued when the USS-725 is set to interface 0, alternate setting 2.

SEQ_EXECUTE
This request is identical to the SEQ_LOAD request, except that the USS-725 will not complete the status phase of the control request until the USS-725 completes execution of the sequence (either through normal completion, crashing, or aborting).

SEQ_ABORT
This request aborts the execution of the sequencer. Upon receipt of this request, the USS-725 will immediately terminate execution of the sequence.

LOAD_EEPROM
The lower 11 bits of wValue specify the starting address in the external EEPROM address to which data is to be written. Data may only be written starting on eight-byte boundaries, which means that the address value must be evenly divisible by eight. The wLength field contains the length of EEPROM image to be written. The data phase of the request consists of the data to be written to the EEPROM.

READ_EEPROM
The lower 11 bits of wValue specify the starting address in the external EEPROM address from which data is to be read. Data may only be read starting on eight-byte boundaries, which means that the address value must be evenly divisible by eight. The wLength field contains the length of EEPROM image to be read. The data phase of the request returns the requested data from the EEPROM.

IEEE 1284 Port

The IEEE 1284 port on the USS-725 is compliant with IEEE 1284-1994 standard (*IEEE Standard Signaling Method for a Bidirectional Parallel Peripheral Interface for Personal Computers*). The parallel port operates in two distinct modes. In fully Automatic Mode, the IEEE 1284 protocol is implemented completely in hardware. Compatibility Mode, Nibble Mode, ECP Mode, and ECP with RLE Mode (with or without hardware-based RLE compression) are supported, with all negotiation, termination, and other features of the protocol handled by the hardware. The USS-725 also features a Register Mode, which presents a standard register interface to the host. These two modes provide the host with two distinct operating paradigms. In Automatic Mode, the software interacts with the USS-725 as if a USB-capable printer; while in Register Mode, the USS-725 emulates standard PC parallel port hardware. In Automatic Mode, the host application software need not know that the data being transferred over USB is actually being transferred to the peripheral via the IEEE 1284 protocol. In Register Mode, the application can interact with the USS-725 registers as if they were standard parallel port registers. The sequencer may also be used to automate register reads, writes, and data transfers.

Register-Based Operation

In its Register Mode of operation, the USS-725 emulates standard host-side parallel port hardware, with the register accesses being performed remotely over a USB connection. As in the standard register set, the Mode field in the Extended Control Register controls the interface mode. The supported modes and their meanings are listed in the **Extended Control Register (ECRR)** section on page 21, and are also described below.

Standard Mode (000)
In this mode, the parallel port is under full software control, with no form of hardware assist. Software has complete control of all parallel port signals. This mode can be used for negotiations, terminations, proprietary handshake sequences, etc. As in standard host-side parallel port hardware, the parallel port data lines are unidirectional outputs in this mode.

Bi-directional Mode (001)
This mode is identical to Standard Mode (000), except that the direction of the parallel port data lines may be controlled with Direction bit in the Control Register.

Compatibility Mode (010)
This mode provides hardware-based Compatibility Mode data transfers. Data sent to the USS-725 Bulk Out pipe will be transferred automatically to the peripheral using Compatibility Mode.

ECP Mode (011)
This mode provides hardware-based ECP Mode data transfers. To use ECP, the host software should negotiate for ECP Mode via the Control and Status Registers, then set Mode to 011. The direction of the ECP transfer is controlled by the Direction bit in the Control Register. Changing the direction of the transfer from reverse to forward also indicates to the USS-725 that any data in the buffers may be made available for reading via the Bulk In pipe.

EPP Mode (100)
This mode provides hardware-based EPP Mode data transfers. To use EPP, the host software should negotiate for EPP Mode via the Control and Status Registers, and then set Mode to 100. Data sent to the USS-725 Bulk Out pipe will be automatically transferred to the peripheral. Address reads and writes, and data reads, are accomplished by reading or writing the EPP Address and EPP Data Registers. These accesses may also be automated with the sequencer, which provides vastly improved EPP data read performance using the Bulk In pipe.

ATA Mode (111)
This mode enables a 16-bit data path for connecting the USS-725 to an ATA hard disk drive. Packing and unpacking of the data from the 8-bit USB data path to the 16-bit ATA bus is handled automatically, and hardware assist provides high-speed ATA PIO mode data and register reads and writes.

Sequencer

The USS-725 sequencer greatly improves the speeds of register transactions between the host and peripheral device by allowing a series of commands to be transferred to the USS-725 in a single USB transaction. The sequencer thus allows some of the controlling intelligence to be moved from the host CPU to the other side of the USB connection. This allows the latency penalty imposed by the host USB hardware and software to be paid only once for a large set of instructions being transferred across the bus, instead of once for each individual access.

After the USS-725 receives the instruction sequence, the sequencer parses the instructions and executes them, performing the register reads, writes, enabling of Bulk In or Bulk Out transfers, register read-modify-writes, register poling, etc. It can perform these actions several orders of magnitude faster than the same operations could be performed by software on the host PC.

The sequencer may be used to implement various protocols. While the sequencer provides access to special hardware-assist state machines for IEEE 1284 and ATA protocols, it also can be viewed as a programmable state machine which can provide control over a set of general-purpose inputs and outputs.

For added functionality and flexibility, the Sequencer may call subroutines from the external memory device which is also used to store descriptor data. These subroutines must have been programmed into the memory device before being called by an ordinary sequence.

For more information, see **Sequencer** on page 25.

Registers

The USS-725 provides eleven registers for access by the host. These are read and written using the GET_1284_REGISTER and SET_1284_REGISTER vendor-specific requests described above, and may also be accessed using the sequencer. The SET_1284_REGISTER request writes a value to a particular register. Writes may either affect the configuration of the hardware and/or have a direct affect on parallel port lines. In the case of the EPP Address and Data Registers, writes initiate write cycles on the parallel port. Note that no register other than the Control Register may be written when the Auto Mode bit in the Control Register is set.

The GET_1284_REGISTER returns nine register values: Status, Control, Extended Control, USS-725 Control, Data, EPP Address/Data, USS-725 Setup, Timeout, and ATA Address Registers, in that order (see Table 21).

Most registers are not affected by reads. However, a read targeting the EPP Address or EPP Data Register will initiate the appropriate read cycle on the parallel port. The value returned will be the address or data byte read from the peripheral as a result of that read cycle. Also, some bits which indicate the equivalent of interrupt status are cleared by being read.

Byte	Symbol	Register Name	Address
0	STAT	Status	1
1	CTRL	Control	2
2	ECCR	Extended Control	6
3	CCTR	USS-725 Control	7
4	DATA	Data	0
5	EPPA/EPPD	EPP Address/Data	3/4
6	SETU	USS-725 Setup	8
7	TIMO	Timeout	9
8	ATAA	ATA Address	10 (0xA)

Table 21 – GET_1284_REGISTER Data

The eleven registers are described in detail in Table 22 through Table 32. The meanings of the Access designations for each bit in those registers are as follows:

R/W = Value read matches value written
R = Bit is read only. Values written will have no effect.
W = Bit is write only. Values read are not values written.

Data Register (DATA)					Address: 0		Reset value: 0x00	
Bit	7	6	5	4	3	2	1	0
Symbol	D7	D6	D5	D4	D3	D2	D1	D0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

7:0	D_7-D_0 (Data) This register is equivalent to and operates in the same manner as the Data register in standard host-side parallel port hardware. The register may be written when Auto Mode is '0' and the Mode field in the Extended Control Register is set to 000, 001, 101, or 110, and is always readable. The read value will be the value of the data latched into the register unless the Mode field is set to 001 and the Direction bit in the Control Register is set to '1' (input mode). In this case, the read value will be the value present on the parallel port data lines.

Table 22 – Data Register (DATA)

Status Register (STAT)					Address: 1		Reset value: N/A	
Bit	7	6	5	4	3	2	1	0
Symbol	nBusy	nAck (IORDY)	PError	Select	nFault	Outputs Disabled	PLH	Timeout
Access	R	R	R	R	R	R	R	R

7	An inverted version of the parallel port Busy signal.
6	The parallel port nAck signal. In ATA Mode (111), this signal is IORDY.
5	The parallel port PError signal.
4	The parallel port Select signal.
3	The parallel port nFault signal.
2	Outputs Disabled. This bit indicates that logic external to USS-725 has taken control of the parallel port by asserting the DISABLE_OUTS signal. The USS-725 parallel port outputs are held in a high impedance state.
1	Peripheral Logic High. The parallel port PLH signal.
0	EPP Timeout. This bit indicates that a timeout has occurred during an EPP read or write. If the peripheral fails to respond to an EPP read or write for longer than 10 µs, the EPP read will abort and this bit will be set (unless the EPP Mask bit in the Control register is set). This bit is cleared by a read.

Table 23 – Status Register (STAT)

Control Register (CTRL)					Address: 2		Reset value: 0x4C	
Bit	7	6	5	4	3	2	1	0
Symbol	HLH	EPP mask	Direction	Int enbl	SelectIn	nInit	AutoFd (DIOR)	Strobe (DIOW)
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

7	The parallel port HLH signal.
6	This bit masks the generation of interrupt status (the setting of the EPP Timeout bit in the Status register) upon timeout of an EPP data or address transfer. Note that in typical host-side parallel port controller chips, this interrupt condition is grouped with and controlled by the Interrupt Enable bit in this register, so for closer emulation of typical parallel port hardware, this bit should always be written with the inverse of the Interrupt Enable bit.
5	When the Mode field in the Extended Control Register is set to 001, this bit controls the direction of the parallel port data lines. When this bit is set to '0' the lines are in output mode, and when set to '1' they are in input mode (see Table 22 above). In ECP Mode (011), this bit controls the direction of the interface, and also is used to protect data bytes in the Bulk In buffer. Setting the bit to '0' will free up any bytes in the buffer for reading via a Bulk In request. This bit has no effect in Modes 000 or 010 (which are unidirectional only), or 100 (where the type of access to the EPP registers uniquely determines the direction). This bit also controls the direction of data transfers using the Bulk pipes when in ATA mode (111).
4	This bit enables the generation of interrupt status on nAck events. If this bit is set, interrupt status will be generated on transitions of nAck from low to high (this status being reflected by the nAck Interrupt bit in the Extended Control Register).
3	An inverted version of the parallel port nSelectIn signal.
2	The parallel port nInit signal.
1	An inverted version of the parallel port nAutoFd signal. In ATA mode (111), this signal is an inverted version of DIOR- (but is under hardware control).
0	An inverted version of the parallel port nStrobe signal. In ATA mode (111), this signal is an inverted version of DIOW- (but is under hardware control).

Table 24 – Control Register (CTRL)

EPP Address Register (EPPA)					Address: 3		Reset value: 0xXX	
Bit	7	6	5	4	3	2	1	0
Symbol	A7	A6	A5	A4	A3	A2	A1	A0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

7:0	A_7-A_0 (EPP Address) This register is equivalent to and operates in the same manner as the EPP Address register in standard host-side parallel port hardware. The register may be written when Auto Mode is '0' and the Mode field in the Extended Control Register is set to 100. A write to this register initiates an EPP Address Write transfer on the parallel port. This register is always readable. When the Mode is set to 100 a read access will initiate an EPP Address Read transfer on the parallel port, and the value returned will be the address value read from the peripheral. Reads when not in Mode 100 will return whatever value has been previously latched, but will not have any effect on the parallel port. In ATA mode (111), this register is used for reading from or writing to disk registers, as selected by the address and chip selects in the ATA Address register. A write to this register while in ATA mode will initiate an ATA PIO register write, while a read from this register will initiate an ATA PIO register read.

Table 25 – EPP Address Register (EPPA)

EPP Data Register (EPPD)					Address: 4		Reset value: 0xXX	
Bit	7	6	5	4	3	2	1	0
Symbol	D7	D6	D5	D4	D3	D2	D1	D0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

7:0	D_7-D_0 (EPP Data) This register is equivalent to and operates in the same manner as the EPP Data register standard host-side parallel port hardware. The register may be written when Auto Mode is '0' and the Mode field in the Extended Control Register is set to 100. A write to this register initiates an EPP Data Write transfer on the parallel port. This register is always readable. When the Mode is set to 100 a read access will initiate an EPP Data Read transfer on the parallel port, and the value returned will be the data value read from the peripheral. Reads when not in Mode 100 will return whatever value has been previously latched, but will not have any effect on the parallel port.

Table 26 – EPP Data Register (EPPD)

ECP Command Register (ECPA)					Address: 5		Reset value: N/A	
Bit	7	6	5	4	3	2	1	0
Symbol	C7	C6	C5	C4	C3	C2	C1	C0
Access	W	W	W	W	W	W	W	W

7:0	C_7-C_0 (ECP Command) This register is equivalent to the ECP Address FIFO register in standard host-side parallel port hardware, but has some different restrictions on its usage. The register may be written when Auto Mode is '0', the Mode field in the Extended Control Register is set to 011, and there is no ECP data in either the Bulk Out buffers or in the process of being transmitted. Writes to this address in a mode other than 011 will be ignored; writes while in mode 011 and the hardware is busy will not complete until the hardware is freed up and the write can complete. This register may also be written when Auto Mode is '1' and the interface is in ECP Forward phase. The value written to this register will be transferred to the peripheral as an ECP command.

Table 27 – ECP Command Register (ECPA)

Extended Control Register (ECRR)					Address: 6		Reset value: 0x03	
Bit	7	6	5	4	3	2	1	0
Symbol	Mode(2)	Mode(1)	Mode(0)	nAck interrupt	nFault interrupt	Bulk In interrupt	Bulk In empty	Bulk Out empty
Access	R/W	R/W	R/W	R	R	R	R	R

7:5	000	Standard Mode	Full software control, data lines are output only
	001	Bi-directional Mode	Full software control, data lines are bi-directional
	010	Compatibility Mode	Hardware handshaking
	011	ECP Mode	Software negotiation, hardware data transfers
	100	EPP Mode	Software negotiation, hardware data transfers
	101	reserved	
	110	reserved	
	111	ATA Mode	ATA mode
	In Automatic mode, this read value of this field is used to indicate the state of the USS-725 parallel port interface as follows: 000 Compatibility Mode 001 Nibble Mode 010 ECP Mode 011 ECP with RLE Mode 100 Device ID in Nibble Mode 101 idle 110 Device ID in ECP Mode 111 Device ID in ECP with RLE Mode		
4	This bit will be set when the parallel port nAck signal makes a transition from ‘0’ to ‘1’ while the Interrupt Enable bit in the Control Register is set to ‘1’. Interrupt status is cleared by any register read.		
3	This bit will be set when the parallel port nFault signal makes a transition from ‘1’ to ‘0’ while the nFault Interrupt Mask bit in the USS-725 Control Register is set to ‘0’. Interrupt status will also be generated if the Mask bit goes low while nFault is low. Interrupt status is cleared by any register read.		
2	This bit will be ‘1’ when Bulk In data is available for reading by the host, and ‘0’ otherwise.		
1	This bit will be clear when there is Bulk In data available for reading by the host or in the process of being made available, and clear when there is not.		
0	This bit will be set when there is Bulk Out data waiting in the buffers or in the process of being transmitted over the parallel port, and clear otherwise.		

Table 28 – Extended Control Register (ECRR)

USS-725 Control Register (CCTR)					Address: 7		Reset value: 0x59	
Bit	7	6	5	4	3	2	1	0
Symbol	Manual Reverse	Slow RRCM and NOOP	Slow Instruction	Return On No Data	nFault int mask	reserved	Compress Enable	Auto Mode
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

7	This bit should be set before doing any operation in register mode (using the sequencer) which will put data in the Bulk In buffer, and cleared when that operation is complete, to allow the Bulk In pipe to correctly handle transmitting the end-of-data condition to the host. Failure to do so may result in USB data transfers terminating prematurely due to short packets being sent. Exception: this bit does not need to be set when using ECP Mode (011), where control of the end-of-data condition is handled by the Direction bit in the Control Register.
6	When set, each iteration of the RRCM and NOOP instructions occur at 16.75 µs intervals. If clear, each iteration is delayed instead by 500 ns. The range of delay time added to RRCM and NOOP instructions is thus: RRCM: 0 to 32.8 ms (fast) or 0 to 1.1 s (slow) NOOP : 500 ns to 2.05 ms (fast) or 16.75 µs to 68.6 ms (slow)
5	When set, each instruction is delayed by a length of time equal 16.75 µs times the value set in the Timeout Register. If clear, each instruction is delayed by 500 ns times the value set in the Timeout Register. The range of additional instruction delay is thus 0 to 128 µs (fast) or 0 to 4.3 ms (slow)
4	If this bit is set, in automatic mode the USS-725 will respond immediately to Bulk In requests for reverse-channel data by sending a zero-length data packet if the peripheral is determined not to have any reverse channel data available. If this bit is clear, the USS-725 will wait in reverse mode until either data is returned by the peripheral or the amount of time set by the Timeout Register expires with no data received.
3	This bit masks the generation of interrupt status on the falling edge of nFault when in ECP Mode (011).
2	For correct operation, this bit **must** always be written to '0'.
1	This bit enables automatic hardware-based RLE compression of reverse-channel data in ECP with RLE Mode (either in Automatic or Register mode). If software desires to use this feature, this bit should be set before attempting to retrieve reverse-channel data. This bit **must** be cleared for proper operation in normal ECP (without RLE) Mode.
0	Setting this bit puts the chip in fully automatic mode. When set, USS-725 can handle all communications with the peripheral with no assistance from software, and all registers except the USS-725 Control Register are read-only. Clearing this bit enables write access to the other registers and disables all automatic operation, provided that none has yet taken place. If any data transfer in automatic mode has occurred, the USS-725 should be reset via the SOFT_RESET request before clearing this bit. The bit can be set again to return the USS-725 to automatic mode at any time, but it is the responsibility of the host software to make sure that the interface and buffers are in a rational state before doing so.

Table 29 – USS-725 Control Register (CCTR)

USS-725 Setup Register (SETU)					Address: 8		Reset value: 0x04	
Bit	7	6	5	4	3	2	1	0
Symbol	Hold bulk	Termination Wait	Clear Bulk Out	Force Negotia-tion	Force Compati-bility	Auto HTR	Filter Software Override	Filter Enable
Access	R/W	R/W	W	W	W	R/W	R/W	R/W

7	When this bit is set, no data will be able to flow into or out of the buffers. Effective only in register mode.
6	When set, a 1-2 ms delay is inserted between any automatic IEEE 1284 mode termination and the subsequent negotiation, and also between the ECP Setup and Forward phases (that is, between Event 31 and Event 33 in *IEEE 1284-1994*) in automatic mode.
5	When this bit is written with a '1', all Bulk Out data will be cleared. This bit will always read as '0'.
4	When this bit is written with a '1' (and the Force Compatibility bit is simultaneously written with a '0'), the hardware will begin automatic negotiation. Effective only in automatic mode. This bit will always read as '0'.
3	When this bit is written with a '1', the hardware will terminate to Compatibility Mode. If the interface is already in Compatibility Mode, no action will be taken. If both this bit and the Force Negotiation bit are written to '1' at the same time, USS-725 will terminate to Compatibility Mode and then assert nInit for approximately 35 ms. Effective only in automatic mode. This bit will always read as '0'.
2	When this bit is clear, automatic performance of Host Transfer Recovery handshaking will be disabled. Note that this will make it impossible for the USS-725 to negotiate to ECP Reverse phase if the peripheral stops taking forward data.
1	When this bit is set, software can control the digital filtering of incoming parallel port signals with the Filter Enable bit. When clear, filters are controlled by Bit 1 of the USS-725 Configuration Byte in the external I2C memory device. (See **Filter Bypass Mode** on page 29 and **ROM Contents and Formats** on page 5.)
0	Controls digital filtering of incoming parallel port signals when the Filter Software Override bit is set. Setting to '1' enables filtering; clearing this bit disables filtering.

Table 30 – USS-725 Setup Register (SETU)

Timeout Register (TIMO)					Address: 9		Reset value: 0x0A	
Bit	7	6	5	4	3	2	1	0
Symbol	T7	T6	T5	T4	T3	T2	T1	T0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

7:0	T_7-T_0 (Timeout value) When the host attempts to read reverse data in automatic mode, the USS-725 will negotiate to the appropriate reverse mode. If the peripheral indicates that reverse data is available but does not transfer any data for the length of time specified by this value, USS-725 will automatically return to forward. The outstanding Bulk In request will be terminated by USS-725 sending whatever data has been received, which will be a 0-byte data packet if the peripheral has transferred no data to the USS-725. The timeout counter operates at a 10-Hz frequency, giving the counter a timeout range of 0 to 25.5 seconds and a default value of 1 second. This timeout value is also used in the sequencer to count time waited each instruction. The timer granularity of that timer is either 500 ns or 16.75 µs, as determined by Slow Instruction bit in the USS-725 Control Register. For more information, see Table 29 on page 22.

Table 31 – Timeout Register (TIMO)

ATA Address Register (ATAA)					Address: 10 (0xA)		Reset value: 0x1F	
Bit	7	6	5	4	3	2	1	0
Symbol	reserved	Mux High	Ignore nACK	CS1 (G4)	CS0 (G3)	DA2 (G2)	DA1 (G1) (Auto RLE)	DA0 (G0) (Auto ECP)
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

7	Set to '0'.
6	If this bit is set, the input to the USS-725 data pipeline will be taken from the upper eight bits of the data bus (PDATA1[7:0]) rather than the lower eight (PDATA0[7:0]). This gives the USS-725 the capability of supporting arbitrary protocols for 16-bit data transfers using the sequencer and the Bulk In pipe. This bit is only effective if the SIXTEEN_BIT input to the USS-725 is asserted high. Note that for data output, the data presented on PDATA1 is always the next byte in the output pipeline after the byte currently presented on PDATA0.
5	If this bit is set, the Compatibility Mode state machine will ignore nAck in the Compatibility Mode handshake. If clear, the USS-725 will check for the peripheral to properly toggle both Busy and nAck.
4	In ATA Mode (111), this bit is the chip select signal CS1-.
3	In ATA Mode (111), this bit is the chip select signal CS0-.
2:0	In ATA Mode (111), these bits are the register address DA2:0.
1	Enable Automatic ECP with RLE. In Automatic mode, if this bit is set (as it is by default), then the hardware will first try ECP with RLE Mode negotiation. If this bit is cleared, the hardware will not attempt ECP with RLE Mode negotiation..
0	Enable Automatic ECP. In Automatic mode, if this bit is set (as it is by default), then the hardware will first try ECP with RLE Mode negotiation (if the Auto RLE bit is set), and then try ECP Mode. If this bit is cleared, the hardware will not attempt either ECP with RLE Mode negotiation or normal ECP Mode negotiation. Thus all Automatic mode transfers will be made in Compatibility and Nibble Modes, regardless of the capabilities of the attached peripheral device.
4:0	G_4-G_0 (General-purpose outputs) These bits may be used in non-ATA applications to provide five general-purpose output signals. Note that in Automatic mode, however, only the upper three bits (G_4-G_2) are available for use as general-purpose outputs.

Table 32 – ATA Address Register (ATAA)

Sequencer

Overview

The USS-725 contains a sequencer that can be used to automate register mode access to the USS-725 registers and control the flow of data into and out of the USS-725 data pipelines.

The host software utilizes the sequencer as follows. The host software generates a sequence of instructions, which may be up to 128 bytes long, and downloads them to the USS-725 using the SEQ_LOAD or SEQ_EXECUTE request (see page 15). As part of the SEQ_LOAD or SEQ_EXECUTE request, the host software specifies a global loop count, which tells the USS-725 how many times to execute the associated sequence. The global loop count is a zero-based count that can be set to anywhere from 0 (the sequence executes once) to 65,535 (the sequence executes 65,536 times).

As soon as the transfer of the instruction sequence completes, the sequencer begins parsing and executing the instructions, reading and writing registers and enabling data transfer as indicated by the instructions. The sequence will continue to run until the entire sequence has been executed the number of times specified by the global loop count, or until the sequence aborts. The sequence can abort due to the detection of an invalid instruction or an RRCM mismatch timeout, or when aborted by the host software via the SEQ_ABORT request (see page 15).

The USS-725 sequencer also supports calls to subroutines stored in an external I^2C memory device. These sequences of instructions must be stored in the external memory before the sequence which calls them is sent to the USS-725. Subroutines may be programmed into the external memory using the LOAD_EEPROM request (see page 15). Each subroutine must be terminated with a RETN command. Note that subroutines stored in an external memory device are read at I^2C interface speeds, which are much slower than normal sequencer execution speeds.

Status about the sequence currently running (if the sequencer is running) or the most recent sequence (if the sequencer is not running) can be retrieved with the GET_SEQ_STATUS request (see page 15).

Sequencer Instructions

The sequencer supports 12 instructions that can be used to automate access to the USS-725 registers and control the flow of data into and out of the USS-725 data pipelines. All instructions must be byte aligned. Details of the instructions are shown in Table 33 through Table 44.

In the instructions that follow, the instruction mnemonic is given first, followed by the actual bit encoding of the instruction. The following codes are used to indicate the purpose of each bit:

R: Register Address
L: Loop Count
D: Data
M: Data Mask
U: Upper/Lower nibble
A: External memory device address
X: Reserved (set to zero)

RWRI	1010		
1010RRRR	DDDDDDDD		
Register Write The Data byte will be written to the specified register.			

Table 33 – Register Write

RRMW	0001		
0001RRRR	DDDDDDDD	MMMMMMMM	
Register Read-Modify-Write The addressed register will be read and the returned data will be modified with the Data and Mask fields and written back. All bits in the original register data for which the corresponding bit Mask byte is set to '1' will be replaced with the corresponding bits from the Data byte. Bits in the original register data for which the corresponding Mask byte is '0' will be unchanged after the RRMW instruction.			

Table 34 – Register Read-Modify-Write

RRCM	0010		
0010RRRR	LLLLLLLL LLLLLLLL	DDDDDDDD	MMMMMMMM
Register Read Compare Until Match The addressed register will be read and data bits for which the corresponding Mask bit is '1' will be compared with the corresponding bits in the Data byte. Data bits for which the Mask bit is '0' will not be compared. If the bits selected by the Mask byte match, execution will pass to the next instruction. If the selected bits don't match, the loop count will be decremented and the instruction will execute again after a delay controlled by the Slow RRCM and NOOP bit in the USS-725 Control Register (see Table 29). The read/mask/compare process will continue until a match is made or the loop count expires, whichever comes first. If the loop count expires, the sequence will be aborted.			

Table 35 – Register Read Compare Until Match

RREN	0011		
0011XXXU			
Register Read Extract Nibble The Status register will be read and the bits corresponding to the lines used in IEEE 1284 Nibble Mode will be extracted and packed into the designated nibble (U = '1' for the upper nibble) in the data stream. If U is set to '1', the current nibble and a previously-latched nibble will be concatenated and loaded into the Bulk In buffer.			

Table 36 – Register Read Extract Nibble

WBIB	0100		
0100LLLL	LLLLLLLL	LLLLLLLL	
Wait for Bulk In Byte The sequencer will enable the data path and wait until the designated number of bytes has been placed into the Bulk In buffer before continuing. Execution will resume when the designated number of bytes has been made available to the USB host. This does not indicate that the host has actually read the data. This command is useful for transferring data using the hardware-assisted input modes provided by the USS-725: ECP (011) and ATA (111).			

Table 37 – Wait for Bulk In Byte

WBOB	0101		
0101LLLL	LLLLLLLL	LLLLLLLL	
Wait for Bulk Out Byte The sequencer will enable the data path and wait until the designated number of bytes has been read from the Bulk Out buffer before continuing. This command is useful for transferring data using the hardware assisted output modes provided by the USS-725: Compatibility (010), ECP (011), EPP (100), and ATA (111).			

Table 38 – Wait for Bulk Out Byte

DATI	0110		
0110RRRR			
Read Register Into Pipe The specified register will be read and the data pushed into the Bulk In buffer.			

Table 39 – Read Register Into Pipe

DATO	0111		
0111RRRR			
Write Register From Pipe A byte will be removed from the Bulk Out buffer and written into the specified register.			

Table 40 – Write Register From Pipe

EPPI	1001		
1001LLLL	LLLLLLLL	LLLLLLLL	
EPP Data Read. The EPP Data register will be read the specified number of times, and the data returned will be pushed into the Bulk In buffer.			

Table 41 – EPP Data Read

NOOP	1000		
1000LLLL	LLLLLLLL		
No Operation The sequencer will wait for the designated number of time periods as selected by the Slow RRCM and NOOP bit in the USS-725 Control register (see Table 29) before execution passes to the next instruction.			

Table 42 – No Operation

SUBR	1011		
1011XAAA	AAAAAAAA		
Execute Subroutine Transfers control to a sequence stored in the external memory device (e.g. EEPROM), beginning at the specified physical address. Sequences in the external memory device may be as long as desired, as long as they do not cross page boundaries, which gives a maximum subroutine length of 256 bytes.			

Table 43 – Execute Subroutine

RETN	1011		
1011XXXX			
Return from Subroutine Returns control back to the main sequence. Execution is transferred to the instruction following the SUBR instruction in the main sequence.			

Table 44 – Return from Subroutine

Operational Modes

Filter Bypass Mode

The USS-725 is equipped with digital filters on the incoming parallel port signals that are used in protocol handshaking (Busy, nAck, PError, nFault, and Select). These are three-sample, unanimous-vote filters running at 12 MHz, and thus filter out all events closer together than 167 ns and pass all events further apart than 250 ns. Disabling the digital filters provides a slight performance improvement, at the expense of increased susceptibility to noise. It is **not** recommended to disable the digital filters when using the USS-725 in an application that drives across a cable.

The default state of these filters, on or off, is controlled by bit 1 in the USS-725 Configuration Byte, the first byte of the attached EEPROM (or other I^2C memory device). If that bit is set, the filters will be enabled. If the bit is clear, the filters will be disabled. See **USS-725 Configuration Byte** on page 6.

The state of the digital filters can also be controlled by software, which overrides the EEPROM default (see Table 30).

Note: If no external memory device is present, digital filters will be enabled.

High Drive Mode

The USS-725 output pads can operate with either totem-pole or open-drain drive. In totem-pole mode, the pads drive continuously with 14 mA current source/sink capability. In open-drain mode, the pads actively pull low when the output signal is low, but let the output float when the output signal is high. However, the pads drive actively for approximately 750 ns after the transition of any output signal to provide fast, clean signal transitions. This is the preferred mode of operation for IEEE 1284 applications.

The option of totem-pole or open-drain drive is controlled by bit 0 in the USS-725 Configuration Byte, the first byte of the attached EEPROM (or other I^2C memory device). If that bit is set, the pads will operate with open-drain drive and High Drive on signal transitions. If the bit is clear, the pads will operate with constant totem-pole drive. See **USS-725 Configuration Byte** on page 6.

Note: If no external memory device is present, the pads will operate with open-drain drive plus High Drive on signal transitions.

Self-Powered Mode

The USS-725 can be configured to operate in either bus-powered or self-powered applications. Configuration is accomplished by supplying a high or low voltage to the SELF_POWERED input pin.

If this input is high, the USS-725 will report a MaxPower of 0x00 (no power used from the USB bus) in the USB Configuration Descriptor. This value will be reported regardless of what is supplied by the external I^2C memory device. Also, the USS-725 will report a '1' (indicating self-powered status) in bit 0 of the information returned in response to a Get Status request from the host.

If the SELF_POWERED pin is low, the USS-725 will return the MaxPower value supplied by the external memory device without modification, and will report a '0' (indicating bus-powered status) in the information returned in response to a Get Status request from the host.

External Clock Mode

The USS-725 is designed to be used with an external 12 MHz crystal (see **External Circuitry Requirements** on page 31). The USS-725 can also be supplied with an external 3.3 V clock signal. If an external clock signal is used, it should be connected to the CLK_LO pin, and CLK_HI should be left unconnected.

The frequency of the supplied crystal or clock signal is selected with the CLKSEL0 and CLKSEL1 input pins. To use an external crystal or 12 MHz input clock signal, tie the CLKSEL0 and CLKSEL1 inputs to ground. To use a 48 MHz input clock signal, tie the CLKSEL0 and CLKSEL1 inputs high (3.3 V).

Limbo Mode

The USS-725 provides a "limbo mode" in which all of its output pads are placed in a high-impedance state. This mode is enabled by setting:

Pin number	Signal name	State
46	CLKSEL1	'1'
47	CLKSEL0	'0'
48	DISABLE_OUTS	'0'

Table 45 – Limbo Mode

Disable Outputs Mode

The USS-725 provides a "disable output mode" in which all of the parallel port output pads are placed in a high-impedance state. The disabled output pins are PDATA0[7:0], PDATA1[7:0], NSELECTIN, NSTROBE (DIOW), NAUTOFD (DIOR), NINIT, HLH, CS1, CS0, and DA[2:0]. Setting the USS-725 into this mode will also cause the assertion of the Outputs Disabled bit in the Status Register (see page 18). This mode is enabled by setting:

Pin number	Signal name	State
46	CLKSEL1	'0'
47	CLKSEL0	'0'
48	DISABLE_OUTS	'1'

Table 46 – Disable Outputs Mode

External Circuitry Requirements

The USS-725 is intended to be as close as possible to a single-chip solution. To accomplish this, it features an on-board USB transceiver, IEEE 1284 transceivers, oscillator, and power-up reset circuit.

The internal oscillator requires attaching a 12 MHz crystal and bias capacitors (see Figure 2).

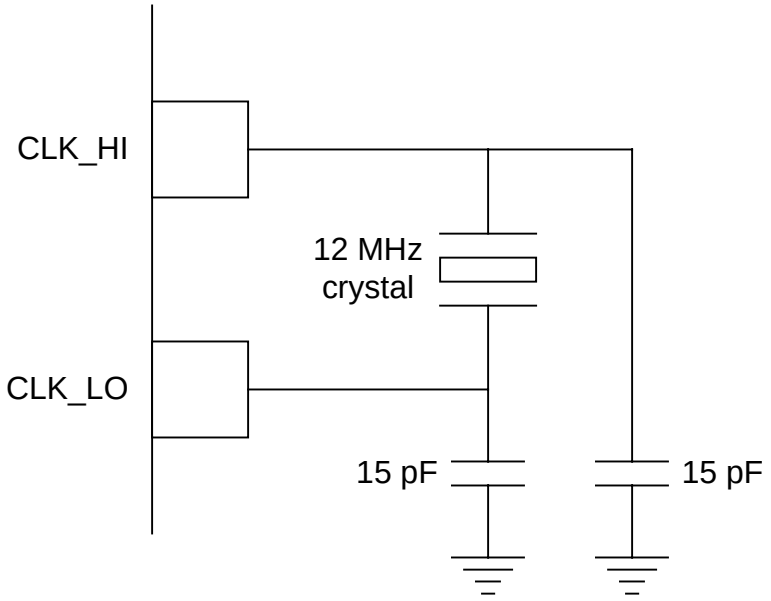

Figure 2 – External Crystal Connection

To use the USS-725 parallel port as an IEEE 1284 level 2 compliant interface, all IEEE 1284 signals (except PLH) require an external 1.2 kΩ ± 5% pull-up resistor to 5 V (see Figure 3). The PLH signal requires a 7.5 kΩ ± 5% pull-down resistor to ground. The USS-725 VDD5 power supply should also be connected to a 5 V supply.

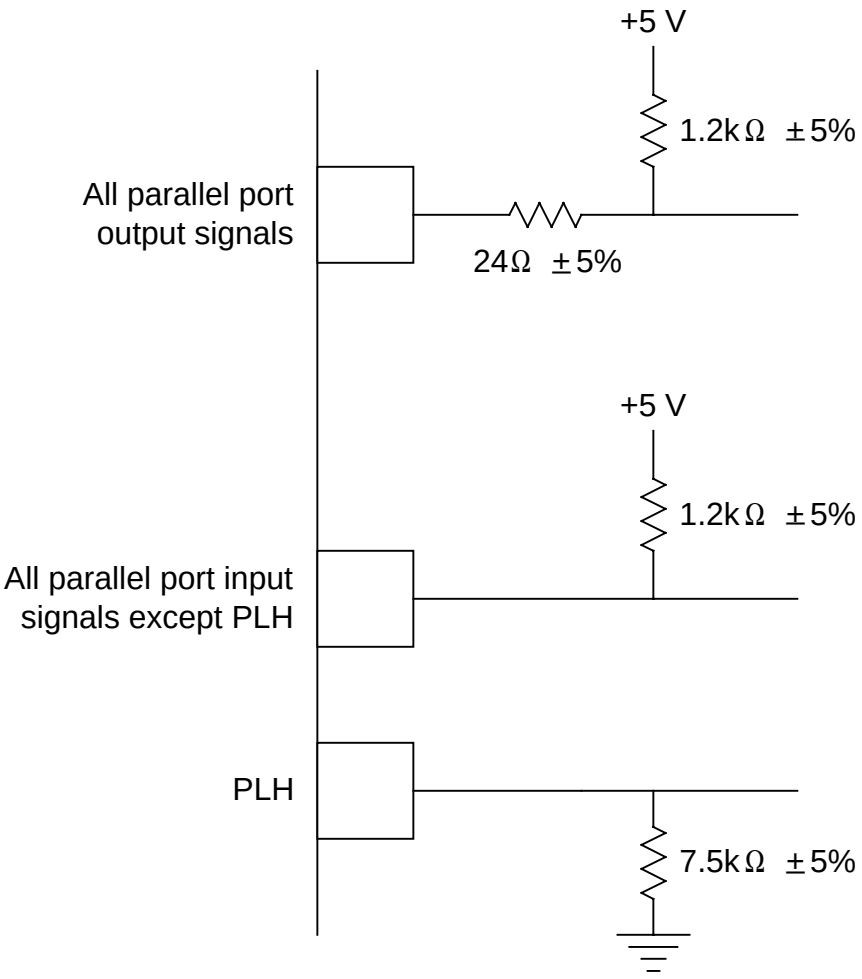

Figure 3 – IEEE 1284 Connections

The USS-725 connection to the USB cable for a bus-powered application is shown in Figure 4. Both DPLS and DMNS require 24 Ω ± 1% series resistors for matching the impedance of the USS-725 USB transceiver to the USB cable. A 1.5 kΩ ± 5% pull-up resistor to 3.3 V is required on the DPLS line for the device to enumerate as a full-speed device.

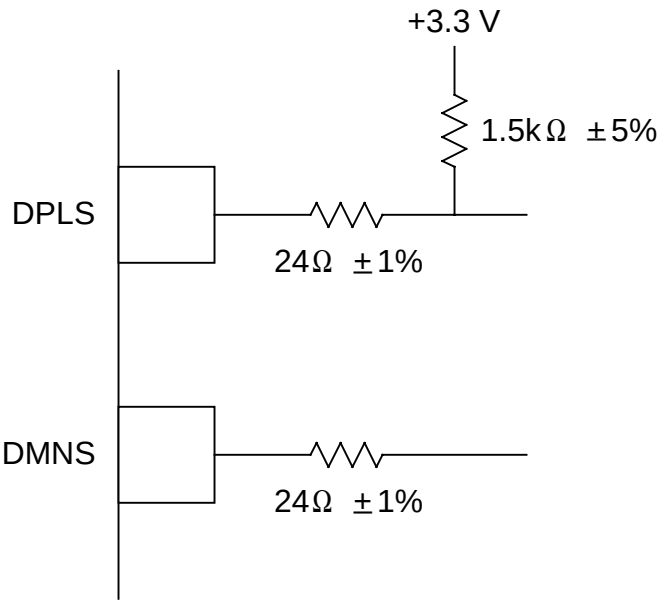

Figure 4 – USB Transceiver Connection (bus-powered)

For a self-powered application, the device must refrain from supplying power through the pull-up resistor when the device is attached to an unpowered bus. The external circuitry must also ensure that the DPLS and DMNS lines are in an appropriate state when the device is powered but not attached to a USB. Figure 5 shows an example connection to meet these requirements.

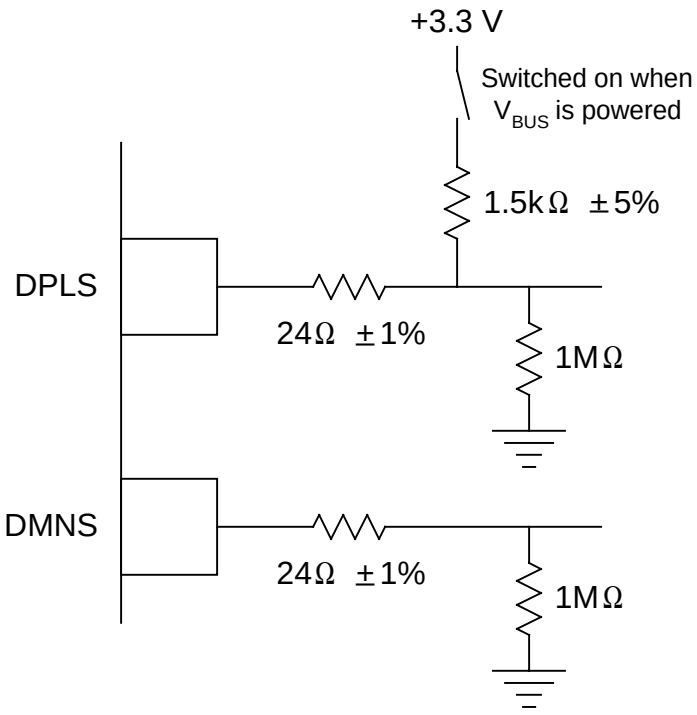

Figure 5 – USB Transceiver Connection (self-powered)

Absolute Maximum Ratings

Stresses in excess of the absolute maximum ratings can cause permanent damage to the device. These are absolute stress ratings only. Functional operation of the device is not implied at these or any other conditions in excess of those given in the operations sections of this data sheet. Exposure to absolute maximum ratings for extended periods can adversely affect device reliability.

Parameter	Symbol	Min	Max	Unit
Ambient Operating Temperature Range	T_A	0	70	°C
Storage Temperature	T_{stg}	-40	125	°C
Voltage on Any Pin with Respect to Ground	—	V_{SS} – 0.3	V_{DD5} + 0.3	V
Power Supply Voltage with Respect to Ground	V_{DD}	—	4.6	V
Power Supply Voltage with Respect to Ground	V_{DD5}	—	*5.5*	V

Table 47 – Absolute Maximum Ratings

Electrical Characteristics

Parameter	Symbol	Test Conditions	Min	Typ	Max	Unit
Input Voltage: Low High	 $\mathbf{V}_{IL}$ $\mathbf{V}_{IH}$	 — —	 — 2.0	 — —	 0.8 —	 V V
Output Voltage: Low High	 $\mathbf{V}_{OL}$ $\mathbf{V}_{OH}$	 — —	 — 2.4	 — —	 0.4 —	 V V
Power Supply Voltage	$\mathbf{V}_{DD}$, $\mathbf{V}_{DDA}$	—	3	3.3	3.6	V
	$\mathbf{V}_{DD5}$	5 V environment 3 V environment	4.375 3	5 3.3	5.5 3.6	V V

Note: (T_A = 0 °C, V_{DD} = 3.3 V ± 0.3 V, V_{SS} = 0 V)

Table 48 – DC Characteristics

	Operating	Suspend
Power supply current (I_{DD})	35 mA (typ)	350 µA (typ)

Note: (T_A = 0 °C, V_{DD} = 3.3 V ± 0.3 V, V_{SS} = 0 V)

Table 49 – Power Supply Current

Timing Characteristics

I²C Memory Device Interface Timing

The I²C memory device interface supports the I²C "fast mode." Timing specifics are given below.

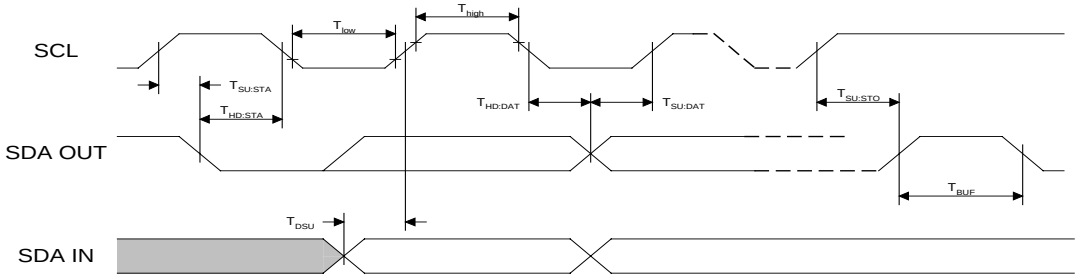

Figure 6 – I²C Memory Device Interface Timing

Parameter	Symbol	Value
Clock high time	T_{high}	667 ± 20 ns
Clock low time	T_{low}	1,333 ± 20 ns
Start condition hold time	$T_{HD:STA}$	667 ± 20 ns
Start condition setup time	$T_{SU:STA}$	667 ± 20 ns
Data output hold time	$T_{HD:DAT}$	667 ± 20 ns
Data output setup time	$T_{SU:DAT}$	667 ± 20 ns
Stop condition setup time	$T_{SU:STO}$	667 ± 20 ns
Required data valid before clock	T_{DSU}	84 ns
Bus free time	T_{BUF}	1,333 ± 20 ns

Table 50 – I²C Memory Device Interface Timing

USB Transceiver Timing Characteristics

The USS-725 USB transceiver complies with the timing and electrical requirements of the *Universal Serial Bus Specification version 1.1*.

Parallel Port Timing Characteristics

All input signals on the USS-725 parallel port are considered to be asynchronous, and are synchronized to the chip's internal system clock. All output signals are clocked using the chip's internal system clock, for which there is no external reference. The output signals should therefore be considered to be asynchronous. All IEEE 1284 protocol transactions comply with *IEEE 1284-1994*.

Clock

	Frequency	Duty Cycle
external crystal	12 MHz ± 0.25%	n/a
external signal (CLKSEL0 = CLKSEL1 = 0)	12 MHz ± 0.25%	10% – 90%
external signal (CLKSEL0 = CLKSEL1 = 1)	48 MHz ± 0.25%	10% – 90%

Note: Clock signal frequency is measured at $V_{DD}/2$ point. Rise and fall times should be 2 ns or less.

Table 51 – Clock Requirements

Reset

The USS-725 includes an on-chip power-up reset block. If the external reset signal is used to provide a hardware power-on reset, the RESET pin should be asserted high for a least 300 µs while the clock is running.

Physical Diagram

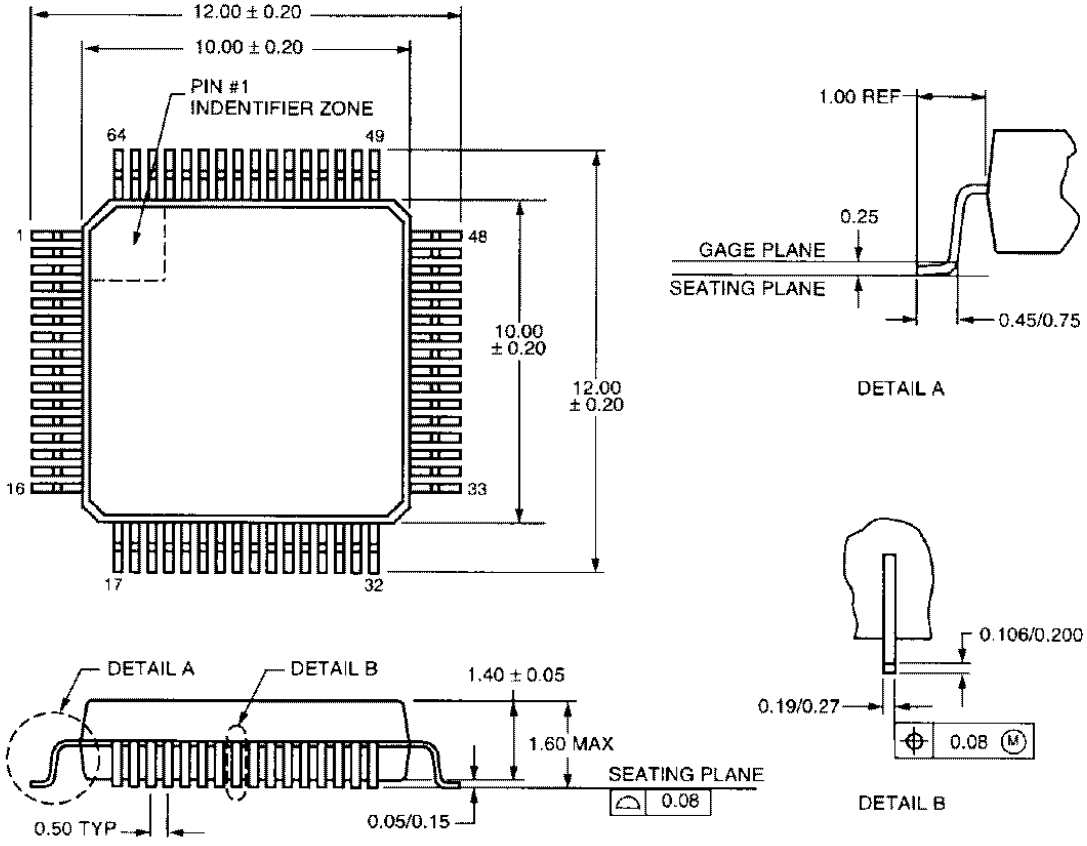

Figure 7 – Package Outline Diagram