IceStorm Learner's Documentation

Roland Lutz

This manual is work-in-progress. You can find the official IceStorm documentation here.

Table of Contents

Introduction

IceStorm has an excellent reference documentation of the 1K and 8K bitstreams as well as a few additional pages explaining the various tiles and the binary bitstream format. However, for a reader not already familiar with IceStorm or the iCE40 FPGA, it would be helpful if a more introduction-like documentation was available.

This document is intended to become this documentation for the IceStorm project and to contain all information which is useful for working with it. It follows the guidelines for GNU documentation (see Section "Documenting Programs" in *GNU Coding Standards*).

There's some important documentation which can't be included here for copyright reasons:

- Lattice iCE40 LP/HX Family Data Sheet
- Lattice iCE Technology Library
- iCE40 Programming and Configuration
- iCEstick Evaluation Kit User's Guide
- FT2232H Datasheet
- FTDI MPSSE Basics
- Command Processor for MPSSE and MCU Host Bus Emulation Modes
- AN2232C-01 Command Processor for MPSSE and MCU Host Bus Emulation Modes

1 Lattice iCE40 1K and 8K FPGAs

This chapter describes the architecture of the Lattice iCE40 1K and 8K FPGAs. ...

1.1 Wires

Wires are how an output of one tile is connected to an input of another tile. There are four different kinds of wires:

- The **carry output** of cell 7 of a logic tile is connected to the carry input of cell 0 of the logic tile directly above.
- Each cell can access the eight outputs of its **neighbouring tiles** in all eight directions directly, except for I/O tiles which aren't connected to other I/O tiles.

  What an "output" is depends on the tile: For logic tiles, it is the output of the eight logic cells. For I/O tiles, outputs 0 to 3 are the `D_IN_0` and `D_IN_1` [XXX] for the `io_0` and `io_1` [YYY], respectively, and outputs 4 to 7 are identical to outputs 0 to 3. For RAM tiles, [bottom: RDATA[7:0], top: RDATA[15:8]].

  As a special exception, the PLL outputs `LOCK` and `SDO` are accessible via otherwise unused diagonal neighbour connections of the corner logic tiles. For the first PLL, `LOCK` is connected to the bottom left neighbour connection 1 of the bottom left logic tile, and `SDO` is connected to the bottom right neighbour connection 3 of the bottom right logic tile. For the second PLL (only available on 4K), `LOCK` is connected to the top left neighbour connection 1 of the top left logic tile, and `SDO` is connected to the top right neighbour connection 1 of the top right logic tile.
- **Span wires** connect tiles across a limited distance, usually in a horizontal or vertical line.

  A span wire can be connected to the inputs and outputs of all the tiles it passes, except for those of the rightmost or bottom tile if that tile is a logic or RAM tile. Not accounting for some irregularities, any input on those tiles can be driven by any span wire, but each span wire can only be driven by a specific output of every second tile it passes.

  Span wires can be connected to each other in specific ways:

  - At their left and right/top and bottom tile, they can be connected to adjacent span wires of the same length, either in a straight line to form a longer span wire or orthogonally to form an L shape.
  - On its second tile from the left/top, each (non-peripheral) span-4 wire can be driven by a specific span-12 wire of the same orientation.
  - I/O tiles allow creating connections between the span-4 wires entering from the fabric: wires 1 and 7 of group 0 can be connected to the corresponding wires of group 2, and those of group 1 to the corresponding wires of group 3. These are the same wires that can be connected to the peripheral span-4 wires.

  There are five different kinds of span wires:

  - **Horizontal span-4 wires** connect up to 5 tiles in a row. Two adjacent tiles are connected by 4 groups of 12 wires each.

    The top and bottom I/O banks have special "peripheral" span-4 wires instead of the normal ones (see below).

- **Vertical span-4 wires** connect up to 9 tiles: 5 tiles in a column and, with the exception of the bottom tile and possible I/O tiles, each tile's left neighbour. Two adjacent tiles are connected by 4 groups of 12 wires each.

  In addition to the tiles to whose inputs and outputs they can normally be connected, vertical span-4 wires can also be connected to the inputs and outputs of these tiles' left neighbour tiles. However, except for the second neighbour tile from the top, only half of the inputs can be driven by any given span wire. (Output connections are not subject to such a restriction.)

  The left and right I/O banks have special "peripheral" span-4 wires instead of the normal ones (see below).
- **Peripheral span-4 wires** connect I/O tiles along the border of the fabric. Two adjacent I/O tiles are connected by 4 groups of 4 wires each.

  Along the edges, peripheral span-4 wires connect 5 tiles. At the bottom left and top right corner of the fabric, they are joined in the way one would expect and still connect 5 tiles.

  However, at the top left and bottom right corners, the groups' order is reversed when passing the corner. This means that the four groups of span wires passing these corners actually connect 2, 4, 6, and 8 tiles, respectively. The order of the signals inside the groups is not changed.

  Instead of the normal mechanism for connecting span wires orthogonally, each peripheral span-4 wire can be connected to one specific span-4 wire entering from the logic fabric at each of its endpoints.
- **Horizontal span-12 wires** connect up to 13 tiles in a row. Two adjacent tiles are connected by 12 groups of 2 wires each. The top and bottom I/O banks don't have span-12 wires.
- **Vertical span-12 wires** connect up to 13 tiles in a column. Two adjacent tiles are connected by 12 groups of 2 wires each. The left and right I/O banks don't have span-12 wires.

  Each time span wires pass from one non-I/O tile to another, they are pairwise crossed out: even wires' numbers increase by one and odd wires' numbers decrease by one.
- There are eight **global wires** which are connected to any tile. They can only be driven by certain I/O pins or by the special `fabout` signal of certain I/O tiles.

1.2 Local tracks

Except for the direct carry connection of vertically neighbouring logic tiles [and global wire special effects], outputs are routed to a tile's inputs via *local tracks*.

The number of local tracks depends on the tile: logic, top RAM, and bottom RAM tiles have 32 local tracks (4 groups @ 8 wires each) while I/O tiles have 16 local tracks (2 groups @ 8 wires each). The tile's own outputs, neighbouring tiles' outputs, and span wires can be connected to the local tracks, but only specific connections are allowed (see the Bitstream Reference for details). The local tracks can then be connected to the tile's inputs.

For global wires, an additional step is required: logic, top RAM, and bottom RAM tiles have 4 global-to-local nets (I/O tiles don't have global-to-local nets). Each of these nets can be connected to any of the eight global nets and to one specific local track.

1.3 Figures

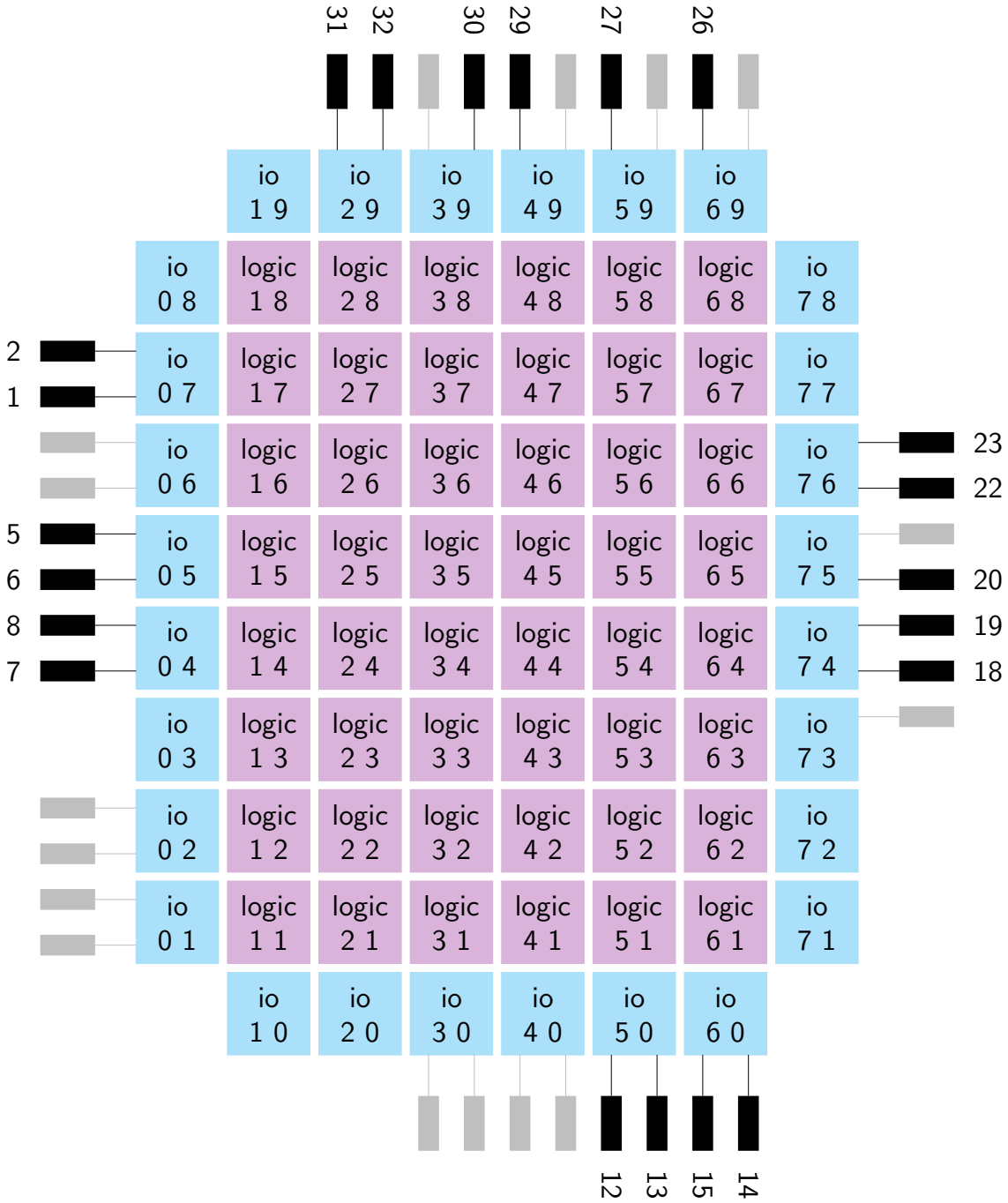

Figure 1.1: Tile fabric for the iCE40 LP384 FPGA. Pins on the left or bottom of an I/O tile are connected to I/O block 0 and pins on the right or top to I/O block 1. Pin numbers correspond to the 32-pin QFN package; pins shown in grey aren't connected in this package but are in other packages.

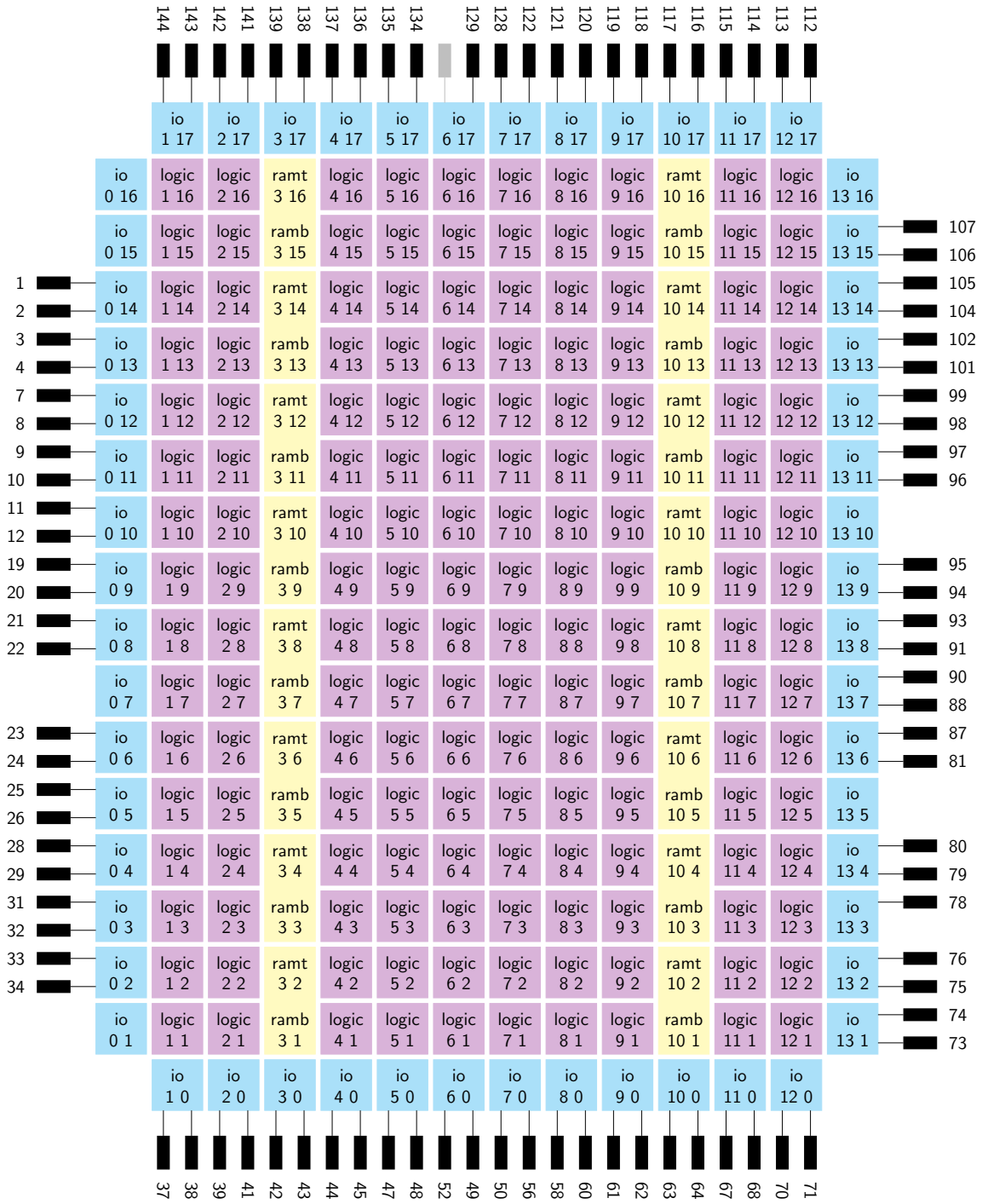

Figure 1.2: Tile fabric for the iCE40 xx1K FPGA. Pins on the left or bottom of an I/O tile are connected to I/O block 0 and pins on the right or top to I/O block 1. Pin numbers correspond to the 144-pin TQFP package; the pin shown in grey isn't connected in this package but is in other packages.

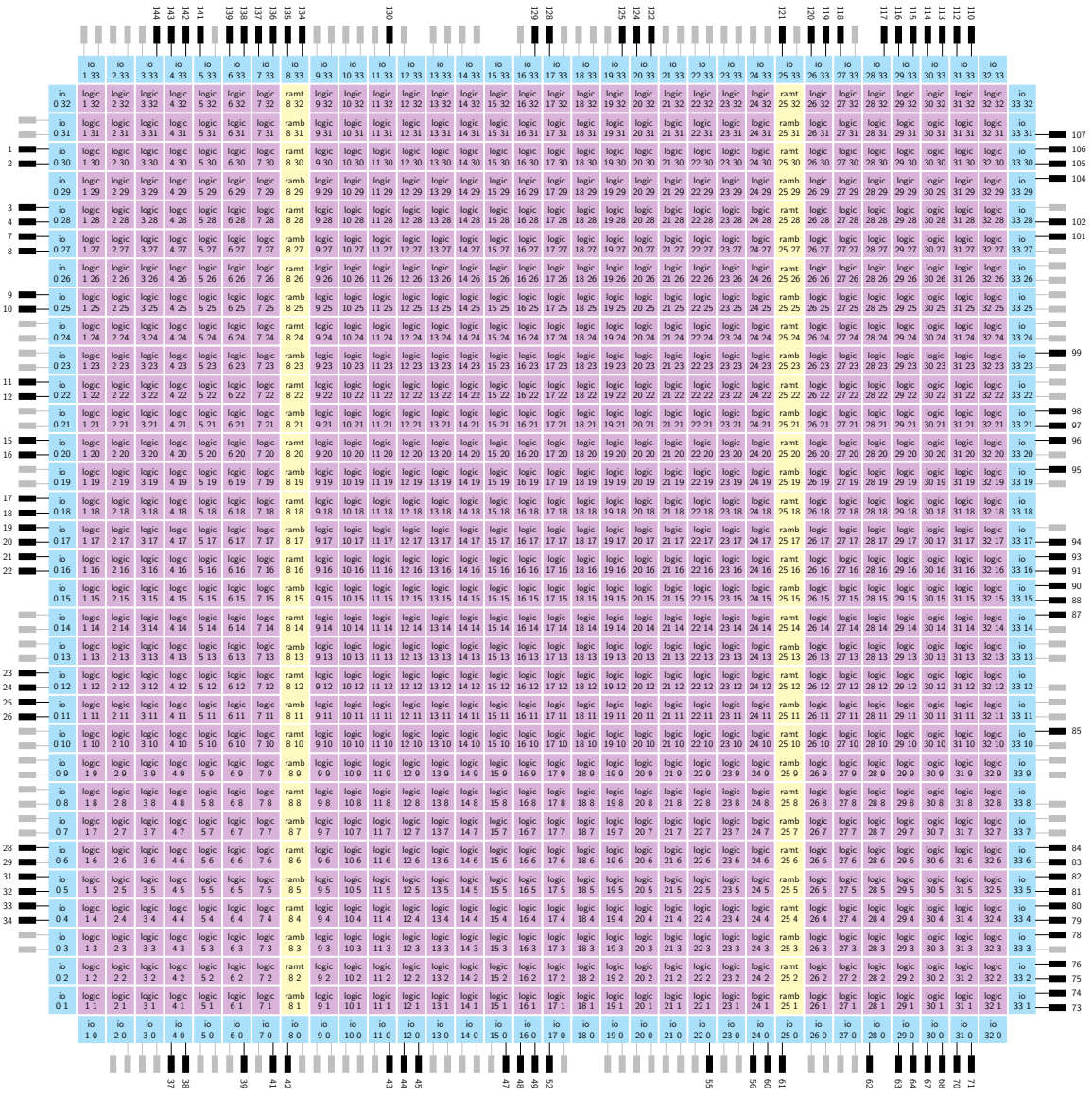

Figure 1.3: Tile fabric for the iCE40 xx4K and xx8K FPGA. Pins on the left or bottom of an I/O tile are connected to I/O block 0 and pins on the right or top to I/O block 1. Pin numbers correspond to the 144-pin TQFP package; pins shown in grey aren't connected in this package but are in other packages.

I suggest you spend a few minutes meditating over these pictures. This is the map of what you are working with; knowing it well will make things much easier.

2 Writing Designs for iCE40 FPGAs

2.1 Pin mappings

I/O Block	Pin	Description	in/out	
0 14 1	1	DCDn	out	
0 14 0	2	DSRn	out	
0 13 1	3	DTRn	in	
0 13 0	4	CTSn	out	
0 12 1	7	RTSn	in	
0 12 0	8	RS232_Tx_TTL	out	(actually, it's not RS232 but UART)
0 11 1	9	RS232_Rx_TTL	in	(actually, it's not RS232 but UART)
0 8 1	21	12.0000 MHz clock	in	
4 0 0	44	J3 10		
4 0 1	45	J3 9		
5 0 0	47	J3 8		
5 0 1	48	J3 7		
7 0 1	56	J3 6		
8 0 1	60	J3 5		
9 0 0	61	J3 4		
9 0 1	62	J3 3		
11 0 0	67	SPI MOSI		
11 0 1	68	SPI MISO		
12 0 0	70	SPI SCK		
12 0 1	71	SPI SS_B		
13 3 1	78	Pmod 1		
13 4 0	79	Pmod 2		
13 4 1	80	Pmod 3		
13 6 0	81	Pmod 4		
13 6 1	87	Pmod 7		
13 7 0	88	Pmod 8		
13 7 1	90	Pmod 9		
13 8 0	91	Pmod 10		
13 9 1	95	D5 (green)	out	
13 11 0	96	D4 (red)	out	
13 11 1	97	D3 (red)	out	
13 12 0	98	D2 (red)	out	
13 12 1	99	D1 (red)	out	
13 14 1	105	IR TXD		
13 15 0	106	IR RXD		
13 15 1	107	IR SD		
12 17 1	112	J1 3		
12 17 0	113	J1 4		
11 17 1	114	J1 5		
11 17 0	115	J1 6		
10 17 1	116	J1 7		
10 17 0	117	J1 8		
9 17 1	118	J1 9		
9 17 0	119	J1 10		

Table 2.1: Pin mapping for the Lattice iCEstick

2.2 USB communication

Un unmodified iCEstick has the following EEPROM configuration:

```
0001 0403 6010 0700 fa80 0000 1111 109a
3caa 0000 0000 0000 0056 0000 0000 0000
0000 0000 0000 0000 0000 0000 0000 0000
0000 0000 0000 0000 0000 0000 0000 0000
0000 0000 0000 0000 0000 0000 0000 0000
0000 0000 0000 0000 0000 0000 0000 0000
0000 0000 0000 0000 0000 0000 0000 0000
0000 0000 0000 0000 0000 0000 0000 0000
0000 0000 0000 0000 0000 0310 004c 0061
0074 0074 0069 0063 0065 033c 004c 0061
0074 0074 0069 0063 0065 0020 0046 0054
0055 0053 0042 0020 0049 006e 0074 0065
0072 0066 0061 0063 0065 0020 0043 0061
0062 006c 0065 0000 0000 0000 0000 0000
0000 0000 0000 0000 0000 0000 0000 165b
```

This translates to the following configuration parameters:

```
Channel A: FIFO (not a virtual COM port)
Channel B: UART (not a virtual COM port, don't suspend on DBUS7 low)

Vendor ID: 0403
Product ID: 6010
Release number: 0700

Power source: bus powered
Remote wakeup: disabled

Max power consumption: 500mA

In endpoint isn't isochronous
Out endpoint isn't isochronous
Don't pull down I/O pins during USB suspend mode
Don't use serial number string

Group AL drive: 8mA (no slow slew, no schmitt input)
Group AH drive: 8mA (no slow slew, no schmitt input)
Group BL drive: 8mA (no slow slew, no schmitt input)
Group BH drive: 8mA (no slow slew, no schmitt input)

Attached EEPROM: 93x56 (actually, it's a 93LC56)

Manufacturer: "Lattice"
Product: "Lattice FTUSB Interface Cable"
No serial string
```

3 Programing the FPGA

3.1 How the configuration is loaded

The configuration of the FPGA is stored in an internal SRAM. Since the SRAM contents are lost when the FPGA isn't powered, the configuration has to be loaded onto the FPGA when it is powered up. There are three different ways how this can be achieved:

- The configuration can be stored in an **external flash**. On power-up or after a reset, the FPGA loads the configuration automatically from the flash, acting as an SPI master. A connection to a computer or external microprocessor is only necessary when the contents of the flash should be updated. (The iCEstick works this way.)
- The configuration can be **written to the SRAM** of the FPGA acting as an SPI slave by a computer or an external microprocessor after each power-up or reset. (This is possible on the iCEstick but requires a hardware modification; see TODO:fig.)
- The configuration can be stored in the FPGA's internal **non-volatile configuration memory**. This allows the FPGA to configure itself instantly without the need for external hardware, mimicking the behavior of an ASIC; but the NVCM is only programmable once, and documentation on this is somewhat scarce. When programming the NVCM, the configuration is written to the FPGA acting as an SPI slave using an interface similar to a 25-series SPI flash (which, in addition to normal SPI slave configuration, also uses the `SPI_SO` pin). Alternatively, fast production NVCM programming via the `VPP_FAST` pin can be used. Using the NVCM is mainly useful for mass-produced hardware or, since read access to the NVCM can be disabled, for obscuring the configuration from the user.

Which of these configuration modes is used is determined by the value of the `SPI_SS_B` pin at the moment when the `CRESET_B` pin returns high. If `SPI_SS_B` is low, the FPGA waits to be configured by an external device. If `SPI_SS_B` is high, the FPGA configures itself from the external flash (or the internal NVCM, if enabled); in this case, the `SPI_SS_B` pin doubles as the slave select output for reading the flash.

When loading the configuration, the SRAM starts out as all zeroes which are gradually replaced with the incoming data. *The configuration bits immediately take effect as they are loaded.* During the configuration, the I/O pins are held in tristate mode; this is released afterwards. The four SPI pins used for configuration (`SPI_SO`, `SPI_SI`, `SPI_SS_B` and `SPI_SCK`) are released 49 clock cycles later.

[CRESET]

[CDONE – Once the configuration is finished, the CDONE pin is set to high.]

3.2 Invoking the `iceprog` Program

```
iceprog [-b|-n|-c] input-file
iceprog -r|-Rbytes output-file
iceprog -S input-file
iceprog -t
```

The `iceprog` program is a simple programming tool for FTDI-based Lattice iCE programmers which can read, write and erase the flash and write the SRAM of an FPGA. It is

typically invoked after the bitstream has been converted by `icepack` to the iCE40 `.bin` format as the last step of the build process to transfer the bitstream to the FPGA.

3.2.1 Operation mode

When no special option is given, `iceprog` erases all 64 kB sectors which would be touched by the written data, writes the data to the flash, and then reads it back and verifies it.

> **Please note:** If the data is not aligned to 64 kB, some data before (if `-o` is used) and after the written data may be erased as well.

The way the flash is erased can be changed with the following options:

'`-b`' Bulk erase the entire flash before writing. When using this option, `iceprog` can be invoked without an *input-file*; in this case, the flash is just bulk erased, and nothing is written.

'`-n`' Don't erase the flash before writing.

Instead of the default erase/write/verify, `iceprog` can perform some other operations:

'`-c`' Just read the data which would have been written from the flash and verify it ("check").

'`-r`' Read the first 256 kB from flash and write them to a file.

'`-R` *size-in-bytes*' Read the specified number of bytes from the flash and write them to a file. You can append `k` to the size to specify it in kilobytes and `M` to specify it in megabytes.

'`-S`' Perform SRAM programming.

'`-t`' Just read the flash ID sequence.

All of the above options are mutually exclusive.

3.2.2 General options

'`-d` *device-string*' Use the specified USB device instead of the default one (which is vendor ID `0x0403` and device ID `0x6010`). The supported notations for *device-string* are:

'`d:`*devicenode*' Path of the bus and device node within USB device tree (usually at `/proc/bus/usb/`). Example: '`d:002/005`'.

'`i:`*vendor*`:`*product*' First device with given vendor and product ID. IDs can be decimal, octal (preceded by '`0`'), or hex (preceded by '`0x`'). Example: '`i:0x0403:0x6010`'.

'`i:`*vendor*`:`*product*`:`*index*' Same as above, with index being the number of the device (starting with 0) if there is more than one device with this vendor and product ID. Example: '`i:0x0403:0x6010:0`'.

'`s:`*`vendor`*`:`*`product`*`:`*`serial-string`*'
First device with given vendor ID, product ID and serial string.

'`-I A|B|C|D`'
Connect to the specified interface on the FTDI chip. If this option is omitted, interface A is used.

'`-o` *`offset-in-bytes`*'
Start reading/writing at address *offset-in-bytes* instead of the beginning of the memory. You can append `k` to the offset to specify it in kilobytes and `M` to specify it in megabytes.

'`-v`' Write more verbose messages.

'`--help`' Display a help text and exit.

3.2.3 Exit status

`0` Success.

`1` A non-hardware error occurred (e.g., failure to read from or write to a file, or invoked with invalid options).

`2` Communication with the hardware failed (e.g., cannot find the iCE FTDI USB device).

`3` Verification of the data failed.

3.2.4 Notes for specific boards

3.2.4.1 Notes for the iCEstick (iCE40HX-1k development board)

An unmodified iCEstick can only be programmed via the serial flash. Direct programming of the SRAM is not supported. For direct SRAM programming, the flash chip and one zero ohm resistor must be desoldered, and the FT2232H SI pin must be connected to the iCE SPI_SI pin as shown in this picture.

3.2.4.2 Notes for the iCE40-HX8K Breakout Board

Make sure that the jumper settings on the board match the selected mode (SRAM or FLASH). See the iCE40-HX8K user manual for details.

4 Communicating with the FPGA

The FTDI IC on the iCEstick which handles USB communication with the computer provides two communication channels: channel A is used to program the configuration onto the flash via SPI while channel B is directly connected to the FPGA and can be used for arbitrary communication. (If the hardware has been modified for direct SRAM programming, channel A is available for communication via SPI as well.)

By default, channel B is configured as an UART (universal asynchronous receiver/transmitter) port, i.e., a serial terminal. To wires named Tx (transmit, FPGA→PC) and Rx (receive, PC→FPGA) are used to transmit data serially (i.e., one bit after another) in packages of 7 or 8 bits. In addition, there are five "modem control lines" which can be used as generic I/O pins (see Table 4.1).

I/O Block	TQFP Pin	Name	Direction	`ioctl` Bit	Valu
0 11 1	9	receive (FPGA) / transmit (PC)	PC→FPGA	—	—
0 12 0	8	transmit (FPGA) / receive (PC)	FPGA→PC	—	—
—	—	line enable	—	`TIOCM_LE`	0x00
0 13 1	3	data terminal ready	PC→FPGA	`TIOCM_DTR`	0x00
0 12 1	7	request to send	PC→FPGA	`TIOCM_RTS`	0x00
—	—	secondary transmit	—	`TIOCM_ST`	0x00
—	—	secondary receive	—	`TIOCM_SR`	0x01
0 13 0	4	clear to send	FPGA→PC	`TIOCM_CTS`	0x02
0 14 1	1	data carrier detect	FPGA→PC	`TIOCM_CAR` / `TIOCM_CD`	0x04
—	—	ring indicator	FPGA→PC	`TIOCM_RNG` / `TIOCM_RI`	0x08
0 14 0	2	data set ready	FPGA→PC	`TIOCM_DSR`	0x10

Table 4.1: UART and modem control lines

4.1 Using the modem control lines as GPIO

The easiest way to comminicate with the FPGA is to use the "modem control lines" DTR, RTS, CTS, (D)CD and DSR as general purpose I/O pins. You can totally disregard their historic meaning and just use them to communicate an arbitrary status from the computer to the FPGA (DTR, RTS) or from the FPGA to the computer (CTS, CD, DSR).

4.1.1 Accessing the modem control lines from the FPGA

The modem control lines are connected to pins 1–4 and 7 of the FPGA as shown in Table 4.1. In order to be able to use them, you have to add them to the `.pcf` file:

```
set_io DCD 1  # out
set_io DSR 2  # out
set_io DTR 3  # in
set_io CTS 4  # out
set_io RTS 7  # in
```

You can assign whatever name you want to them here; all that matters to the FPGA is which pins it should input from or output to. Just make sure not to confuse an input for an output as you may damage the board.

Now, you can add them to the top module declaration using your assigned names just like any I/O pin. Here's an example design with a binary counter which is shown on the middle three LEDs as well as signalled on the three output lines. The state of the two input lines is displayed on the two outer LEDs:

```
‘default_nettype none

module top(
        input clk,

        output DCD,
        output DSR,
        input DTR,
        output CTS,
        input RTS,

        output LED0,
        output LED1,
        output LED2,
        output LED3,
        output LED4);

    reg [25:0] counter;
    always @(posedge clk)
        counter <= counter + 1;

    assign {LED2, LED0} = ~{DTR, RTS};
    assign {LED3, LED4, LED1} = counter[25:23];
    assign {DCD, DSR, CTS} = ~{counter[25], counter[24], counter[23]};
endmodule
```

Note: The modem pins (but not Tx and Rx) on the FTDI chip are active low, so the levels have to be inverted when reading/writing them.

4.1.2 Accessing the modem control lines from the computer

You can examine and toggle the state of the modem control lines from the computer using a standard terminal program. I'll explain the process using `GtkTerm` as an example. After starting up the application, select Configuration → Port from the menu (or press *Ctrl+Shift+S*) to bring up the configuration dialog. Select `/dev/ttyUSB1` as a serial port (if you have multiple USB serial devices connected to your computer, the number may be higher) and press OK. The other settings don't matter for now.

You should then be able to see the indicators in the status bar flash between disabled and normal. To toggle the state of the lines from the computer to the FPGA, you can use Control signals → Toggle DTR and Control signals → Toggle RTS (*F7* and *F8*).

4.1.3 Writing a program which accesses the modem control lines

You can control the modem control lines from your own program by using the `ioctl(2)` system call; see `tty_ioctl(4)` for a list of commands which can be used on a serial port. To illustrate this, here's an example program which prints the state of the FPGA→computer lines and sets the computer→FPGA lines whose names have been given as command line arguments to high and the other lines to low:

```
#include <err.h>
#include <fcntl.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <sys/ioctl.h>
#include <unistd.h>

const char *tty_path = "/dev/ttyUSB1";

int main(int argc, char *argv[])
{
        int fd = open(tty_path, O_RDWR | O_NOCTTY);
        if (fd == -1)
                err(EXIT_FAILURE, "open: %s", tty_path);

        int bits = 0;
        if (ioctl(fd, TIOCMGET, &bits) == -1)
                err(EXIT_FAILURE, "ioctl: TIOCMGET");

        printf("CTS = %d, DCD = %d, DSR = %d\n", (bits & TIOCM_CTS) != 0,
                                                 (bits & TIOCM_CD) != 0,
                                                 (bits & TIOCM_DSR) != 0);

        bits &= ~TIOCM_DTR;
        bits &= ~TIOCM_RTS;

        int i;
        for (i = 1; i < argc; i++) {
                if (strcasecmp(argv[i], "DTR") == 0)
                        bits |= TIOCM_DTR;
                else if (strcasecmp(argv[i], "RTS") == 0)
                        bits |= TIOCM_RTS;
                else
                        errx(EXIT_FAILURE, "%s: invalid argument", argv[i]);
        }

        if (ioctl(fd, TIOCMSET, &bits) == -1)
                err(EXIT_FAILURE, "ioctl: TIOCMSET");
```

```
        if (close(fd) == -1)
                err(EXIT_FAILURE, "close");

        return EXIT_SUCCESS;
}
```

You can compile this program using the following command:

```
$ gcc -Wall -W -g -o mcl-example mcl-example.c
```

To run it, specify the bits to be set high on the command line:

```
$ ./mcl-example
$ ./mcl-example dtr
$ ./mcl-example rts
$ ./mcl-example dtr rts
```

Note: There's a serial line control mode flag called `HUPCL` which is set by default and causes the modem control lines to be lowered after the last process closes the port. If you want your changes to have a visible effect, you have to either add some more code to your program so it doesn't quit immediately, keep another program open on the terminal, or use the following command to reset the `HUPCL` flag:

```
$ stty -hupcl < /dev/ttyUSB1
```

This will take effect until you reboot you computer, after which you'll have to re-run the command. If you want to undo this for some reason, you can do so with the following command:

```
$ stty hupcl < /dev/ttyUSB1
```

4.2 Communication with the FPGA via UART

4.2.1 Sending and receiving bytes from the FPGA

4.2.2 Connecting to the FPGA from the computer

4.2.3 Writing a program which uses a serial port

4.3 Using the FTDI IC in FIFO mode

4.3.1 Configuring the FTDI IC to use FIFO mode

4.3.2 Sending and receiving data from the FPGA

4.3.3 Writing a program which accesses the FTDI IC in FIFO mode

4.3.4 Writing a program which accesses the FTDI IC in MPSSE mode

5 Using multiple configurations

5.1 Invoking the `icemulti` Program

`icemulti [`*option*`]... `*input-file*`...`

The `icemulti` program creates a multi-configuration image for Lattice iCE40 FPGAs from up to four individual configuration images. This is used with the coldboot and warmboot mechanisms which allow to select a configuration image to be loaded after a reset based on the values read from the `CBSEL0` and `CBSEL1` pins or on the values of the `fabout` nets of I/O tiles 13 1 and 13 2, respectively.

`icemulti` is typically invoked after the bitstream has been converted to the iCE40 `.bin` format by `icepack` and before programming the resulting image onto the configuration flash using `iceprog`.

5.1.1 Options

`-c`
`--coldboot`
: Enable coldboot mode: the image which is loaded on power-on or after a low pulse on `CRESET_B` is determined by the values read from the pins `CBSEL0` and `CBSEL1`.

`-p` *index*
: When not using coldboot mode, specify the image to be used on power-on or after a low pulse on `CRESET_B`. *index* must be in the range 0...3. This option can't be used in combination with `-c`.

`-a` *N*
: Align the images to 2^*N* bytes within the resulting bitstream. The resulting extra space is filled with `0xff`.

`-A` *N*
: Like `-a`*N*, but align the first image, too.

`-o` *output-file*
: Write the resulting bitstream to *output-file*. (If this option is not specified, the default is to print it to the standard output.)

`-v`
`--verbose`
: Print the image offsets to stderr.

5.1.2 Resulting bitstream

The bitstream generated by `icemulti` consists of five 32-byte headers followed by the actual configuration images. If the options `-a` or `-A` have been specified, the unused bytes between the images are filles with `0xff`.

Each header specifies the start offset of an image: the first header specifies the the offset of the image to be used on power-on or after a low pulse on `CRESET_B` unless coldboot is used, and the following headers specify the offsets of the four images available via the coldboot or warmboot mechanism. Unused headers point to the first configuration image. In addition, the first header contains a bit specifying whether the coldboot mechanism should be used.

5.1.3 Exit status

Returns ‘`0`’ on success and ‘`1`’ on failure.

Appendix A GNU Free Documentation License

Version 1.3, 3 November 2008

0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document *free* in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or non-commercially. Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released

under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

The "publisher" means any person or entity that distributes copies of the Document to the public.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

   If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

   If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

   If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

   It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

   You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

   A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any,

be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.

B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.

C. State on the Title page the name of the publisher of the Modified Version, as the publisher.

D. Preserve all the copyright notices of the Document.

E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.

F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.

G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.

H. Include an unaltered copy of this License.

I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.

J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.

K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.

L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.

M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.

N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.

O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their

titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties—for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

   You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

   The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

   In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements."

6. COLLECTIONS OF DOCUMENTS

   You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

   You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

   A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

   If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

   Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

   If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

   You may not copy, modify, sublicense, or distribute the Document except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, or distribute it is void, and will automatically terminate your rights under this License.

   However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

   Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

   Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, receipt of a copy of some or all of the same material does not give you any rights to use it.

10. FUTURE REVISIONS OF THIS LICENSE

    The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See http://www.gnu.org/copyleft/.

    Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation. If the Document specifies that a proxy can decide which future versions of this License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Document.

11. RELICENSING

    "Massive Multiauthor Collaboration Site" (or "MMC Site") means any World Wide Web server that publishes copyrightable works and also provides prominent facilities for anybody to edit those works. A public wiki that anybody can edit is an example of such a server. A "Massive Multiauthor Collaboration" (or "MMC") contained in the site means any set of copyrightable works thus published on the MMC site.

    "CC-BY-SA" means the Creative Commons Attribution-Share Alike 3.0 license published by Creative Commons Corporation, a not-for-profit corporation with a principal place of business in San Francisco, California, as well as future copyleft versions of that license published by that same organization.

    "Incorporate" means to publish or republish a Document, in whole or in part, as part of another Document.

    An MMC is "eligible for relicensing" if it is licensed under this License, and if all works that were first published under this License somewhere other than this MMC, and subsequently incorporated in whole or in part into the MMC, (1) had no cover texts or invariant sections, and (2) were thus incorporated prior to November 1, 2008.

    The operator of an MMC Site may republish an MMC contained in the site under CC-BY-SA on the same site at any time before August 1, 2009, provided the MMC is eligible for relicensing.

ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

```
Copyright (C)  year  your name.
Permission is granted to copy, distribute and/or modify this document
under the terms of the GNU Free Documentation License, Version 1.3
or any later version published by the Free Software Foundation;
with no Invariant Sections, no Front-Cover Texts, and no Back-Cover
Texts.  A copy of the license is included in the section entitled ``GNU
Free Documentation License''.
```

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the "with. . . Texts." line with this:

```
with the Invariant Sections being list their titles, with
the Front-Cover Texts being list, and with the Back-Cover Texts
being list.
```

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.

Index