LibXil Standard C Libraries

EDK 10.1, January 14, 2008

Summary

This document describes the software libraries available for the embedded processors. The document contains the following sections:

- "Overview"
- "Additional Resources"
- "Standard C Library (libc.a)"
- "Xilinx C Library (libxil.a)"
- "Input/Output Functions"
- "Memory Management Functions"
- "Arithmetic Operations"
- "Thread Safety"

Overview

The Xilinx® Embedded Development Kit (EDK) libraries and device drivers provide standard C library functions, as well as functions to access peripherals. The EDK libraries are automatically configured by Libgen for every project based on the Microprocessor Software Specification (MSS) file. These libraries and include files are saved in the current project `lib` and `include` directories, respectively. The **-I** and **-L** options of `mb-gcc` are used to add these directories to its library search paths.

Additional Resources

- *MicroBlaze Processor Reference Guide*
  http://www.xilinx.com/ise/embedded/mb_ref_guide.pdf
- *Embedded System Tools Reference Manual*
  http://www.xilinx.com/ise/embedded_design_prod/platform_studio.htm

Standard C Library (libc.a)

The standard C library, `libc.a`, contains the standard C functions compiled for the MicroBlaze™ processor or the PowerPC™ processor. You can find the header files corresponding to these C standard functions in <*XILINX_EDK*>/gnu/<*processor*>/<*platform*>/<*processor-lib*>/include, where:

- <*XILINX_EDK*> is the <*Installation directory*>
- <*processor*> is `powerpc-eabi` or `microblaze`
- <*platform*> is `sol`, `nt`, or `lin`
- <*processor-lib*> is `powerpc-eabi` or `microblaze-xilinx-elf`

The `lib.c` directories and functions are:

_ansi.h	fastmath.h	machine/	reent.h	stdlib.h	utime.h
_syslist.h	fcntl.h	malloc.h	regdef.h	string.h	utmp.h
ar.h	float.h	math.h	setjmp.h	sys/	
assert.h	grp.h	paths.h	signal.h	termios.h	
ctype.h	ieeefp.h	process.h	stdarg.h	time.h	
dirent.h	limits.h	pthread.h	stddef.h	unctrl.h	
errno.h	locale.h	pwd.h	stdio.h	unistd.h	

Programs accessing standard C library functions must be compiled as follows:

For MicroBlaze processor:

```
mb-gcc <C files>
```

For PowerPC processors:

```
powerpc-eabi-gcc <C files>
```

The `libc` library is included automatically.

For programs that access `libm` math functions, specify the **`-lm`** option.

Refer to “MicroBlaze Application Binary Interface (ABI)” section in the *MicroBlaze Processor Reference Guide* for information on the C Runtime Library. The “Additional Resources,” page 1 contains a link to the document.

Xilinx C Library (libxil.a)

The Xilinx C library, `libxil.a`, contains the following object files for the MicroBlaze processor embedded processor:

```
_exception_handler.o
_interrupt_handler.o
_program_clean.o
_program_init.o
```

Default exception and interrupt handlers are provided. The `libxil.a` library is included automatically.

Programs accessing Xilinx C library functions must be compiled as follows:

```
mb-gcc <C files>
```

Input/Output Functions

The EDK libraries contains standard C functions for I/O, such as `printf` and `scanf`. These functions are large and might not be suitable for embedded processors.

The prototypes for these functions are in `stdio.h`.

Note: The C standard I/O routines such as printf, scanf, vfprintf are, by default, line buffered. To change the buffering scheme to no buffering, you must call setvbuf appropriately. For example:

```
setvbuf (stdout, NULL, _IONBF, 0);
```

These Input/Output routines require that a newline is terminated with both a CR and LF. Ensure that your terminal CR/LF behavior corresponds to this requirement.

Refer to the "Microprocessor Software Specification (MSS)" chapter in the *Embedded System Tools Reference Manual* for information on setting the standard input and standard output devices for a system. The "Additional Resources," page 1 contains a link to the document.

In addition to the standard C functions, the EDK processors (MicroBlaze processor and PowerPC 405 processor) library provides the following smaller I/O functions:

`void` **`print`** `(char *)`

This function prints a string to the peripheral designated as standard output in the Microprocessor Software Specification (MSS) file. This function outputs the passed string as is and there is no interpretation of the string passed. For example, a `"\n"` passed is interpreted as a new line character and not as a carriage return and a new line as is the case with ANSI C printf function.

`void` **`putnum`** `(int)`

This function converts an integer to a hexadecimal string and prints it to the peripheral designated as standard output in the MSS file.

`void` **`xil_printf`** `(const *char ctrl1,...)`

This function is similar to `printf` but much smaller in size (only 1 kB). It does not have support for floating point numbers. `xil_printf` also does not support printing of long (such as 64-bit numbers).

Memory Management Functions

The MicroBlaze processor and PowerPC processor C libraries support the standard memory management functions such as malloc( ), calloc( ), free( ). Dynamic memory allocation provides memory from the program heap. The heap pointer starts at low memory and grows toward high memory. The size of the heap cannot be increased at runtime. Therefore an appropriate value must be provided for the heap size at compile time. The malloc( ) function requires the heap to be at least 128 bytes in size to be able to allocate memory dynamically (even if the dynamic requirement is less than 128 bytes). The return value of `malloc` must always be checked to ensure that it could actually allocate the memory requested.

Arithmetic Operations

Software implementations of integer and floating point arithmetic is available as library routines in `libgcc.a` for both processors. The compiler for both the processors inserts calls to these routines in the code produced, in case the hardware does not support the arithmetic primitive with an instruction.

MicroBlaze Processor

Integer Arithmetic

By default, integer multiplication is done in software using the library function `__mulsi3`. Integer multiplication is done in hardware if the mb-gcc option, **`-mno-xl-soft-mul`**, is specified.

Integer divide and mod operations are done in software using the library functions `__divsi3` and `__modsi3`. The MicroBlaze processor can also be customized to use a hard divider, in which case the div instruction is used in place of the `__divsi3` library routine.

Double precision multiplication, division and mod functions are carried out by the library functions `__muldi3`, `__divdi3`, and `__moddi3`, respectively.

The unsigned version of these operations correspond to the signed versions described above, but are prefixed with an `__u` instead of __.

Floating Point Arithmetic

All floating point addition, subtraction, multiplication, division, and conversions are implemented using software functions in the C library.

PowerPC Processor

Integer Arithmetic

Integer addition and subtraction operations are provided in hardware. Hence no specific software library is available for the PowerPC processor.

Floating Point Arithmetic

The PowerPC processor supports all floating point arithmetic implemented in the standard C library.

Thread Safety

The standard C library provided with EDK is not built for a multi-threaded environment. STDIO functions like printf(), scanf() and memory management functions like malloc() and free() are common examples of functions that are not thread-safe. When using the C library in a multi-threaded enviroment proper mutual exclusion techniques must be used to protect thread unsafe functions.