Appendix A

Moving from the CM-2 to the CM-5

This appendix is for CM-2 and CM-200 users who want to port their programs to the CM-5.

A.1 Updating the Program

A.1.1 CM Fortran

Programs written in the CM Fortran language run on both the CM-5 and the CM-2/200. You need not make any changes in the use of language features to port a program from one platform to another. Some other system features, however, are platform-dependent. For example, send addresses on the CM-5 are 8-byte integers. Arrays that contain send addresses, therefore, should be declared (on any CM) as **`DOUBLE PRECISION`** or **`REAL*8`**. Send address arrays declared in this way are portable across all CM platforms. Also, some compiler switches apply to only one platform. Optimizations are the same for the CM-5 and the slicewise model on the CM-2 and CM-200; Paris optimizations do not apply to the CM-5.

Assemblers also differ: the Paris instruction set is not supported on the CM-5, so calls to Paris must be removed from CM-2/200 programs before they are ported to the CM-5. Such calls are typically replaced by calls to the utility library. Other library calls may also be nonportable:

- CMMD, CMX11, and CM/AVS are supported only on the CM-5.
- CM-2/200 visualization libraries are supported only on the CM-2 and CM-200.

Copyright © 1993–1994 Thinking Machines Corporation

- The CM Fortran Utility Library and CMSSL are supported on all CM systems.

A.1.2 C*

Most CM-2/200 C* programs should port without difficulty to the CM-5. You must recompile and relink using the CM-5 C* compiler. This list summarizes the changes that you must make (when applicable) to ensure portability:

- Remove all calls to libraries (like Paris) not supported on the CM-5.
- Remove all include files not supported on the CM-5 (for example, **<cm/paris.h>**).
- If you express lengths in terms of bits in a function (for example, in the overloaded versions of the grid communication functions or the **get** or **send** function), rewrite the code to express the size with **boolsizeof** and the appropriate parallel type.
- Change calls to **allocate_detailed_shape** to use the new format for CM-5 C*.
- The CM-5 C* compiler disallows casts between scalar types and pointers to parallel variables. If you call **palloc()** in a CM-2/200 C* program without including **<stdlib.h>** (which properly declares its return type) and cast the result, the code won't compile on the CM-5. Thus, this code won't work:

```
/* No included stdlib.h file */

int:current *p = (int:current *)palloc(current,
              boolsizeof(int:current));
```

Change it to this so that it will work in CM-5 C*:

```
#include <stdlib.h>

int:current *p = palloc(current,
                  boolsizeof(int:current));
```


Copyright © 1993–1994 Thinking Machines Corporation

A.2 Compiling and Linking

A.2.1 CM Fortran

In compiling CM Fortran programs, note that the CM-2/200 options **`-paris`** and **`-slicewise`** are not supported on the CM-5; the CM-5 supports only the options **`-sparc`** and **`-vu`**.

On the CM-5, **`cmf`** does not invoke the linker **`ld`** directly. If you link as a separate step, we recommend reinvoking **`cmf`**.

A.2.2 C*

You compile and link CM-5 C* programs as you do CM-2/200 C* programs. CM-5 C* supports most CM-2/200 compiler options. Exceptions are

- **`-noline`**
- **`-release`**
- **`-ucode`**
- **`-pg`**
- **`-keep c`**, since the compiler does not generate C code

A.3 Executing

When executing a CM Fortran or C* program, note that you do not issue the commands **`cmattach`** or **`cmcoldboot`**. A CM-5 partition manager is always "attached" to the nodes it controls. In addition, the **`cmsetsafety`** command is not available on the CM-5.

The CM-2/200 checkpointing facility is not currently supported on the CM-5.

Copyright © 1993–1994 Thinking Machines Corporation

A.4 *Lisp

CM-5 supports the *Lisp interpreter. Most CM-2/200 *Lisp code will port without any changes. However, operations that are not part of *Lisp, such as calls to Paris or other low-level facilities, either will not work on the CM-5 or will require significant revision. For complete information, see the manual *Porting to CM-5 *Lisp*.

CMOST Version 7.3, April 1994
Copyright © 1993–1994 Thinking Machines Corporation