C++ compiler characteristics

- "Cross compiler": target ≠ host
  - Target – where generated code is run
  - Host – where code is generated
- GNU C/C++ 3.3.5 (gcc)
  - Supports C (.c) and C++ (.cpp)
  - Most C libraries available (*e.g.*, <math.h>, <stdio.h>, …)

1

C++ data types available

- **`byte/word`** aliases from <msoe/common.h>
- [[**`un`**]**`signed`**] **`char`**
  - **`unsigned`** a.k.a. **`byte`** (8-bit)
- [[**`un`**]**`signed`**] **`short`** [**`int`**] (16-bit)
  - **`unsigned`** a.k.a. **`word`**
- [[**`un`**]**`signed`**] **`int`**
  - 32-bit by default, but we use -mshort (16-bit)
- [[**`un`**]**`signed`**] **`long`** [**`long`**] [**`int`**] (32/64-bit)
- **`bool`**
- User-defined **`class`**es
- Pointers

2

About floating point numbers

- **`float`** and **`double`** are supported
- HC11 has no floating point instructions, so linker must add support subroutines
- Very expensive in time and space!

3

Floating point size example (GCC 3.0.4)

Description	Prog. Size
Short string output, compares **`int`**s	0x01F1
`float` instead of **`int`**	0x0654
`double` instead of **`float`**	0x0906
Divide **`double`** by a **`const double`**	0x1405
Largest block of RAM on Fox11	0x7C00

4

Useful operators

- ~ ! Invert (bitwise / logical {bool})
- | || OR (bitwise / logical)
- & && AND (bitwise / logical)
- ^ XOR (bitwise)
- << left shift (e.g., k = j<<3;)
- >> right shift
- All binary operators have '=' forms
  - *e.g.*, |=, <<=

5

`volatile` qualifier

- "The purpose of **`volatile`** is to force an implementation to suppress optimization that could otherwise occur. For example, for a machine with memory-mapped input/output, a pointer to a device register might be declared as a pointer to **`volatile`**, in order to prevent the compiler from removing apparently redundant references through the pointer."
  - – p. 211 of The *C Programming Language*, 2 ed., by Kernighan and Ritchie

6

.h files available (1/3)

- #include <msoe/ports.h> (standard ports)
  - declares all ports volatile (0x1000-0x103F)
  - values included when linking with –lmsoe
- #include "ports_fox11.h" **(new on website)**
  - Defines fox11_port{b,c,f}
- #include <msoe/common.h>
  - defines:
    - byte = unsigned char
    - word = unsigned short
    - size_t = long unsigned

7

.h files available (2/3)

- #include <msoe/time.h>
  - `void wait1ms(void);`
  - `void wait(int msec);`
- #include <msoe/display.h>
  - `char digit2ascii(byte digit);`
    `// Hex digit to ASCII`
  - `Wookie / Briefcase Display`
    - `void showstring(const char* sin);`
    - `void showchar(int pos, char ch);`
    - `// positions are 1234 from left`

8

.h files available (3/3)

- #include <msoe/string.h>
  - `char *itos(int value, char *string, int format);`
  - `char *strcpy(char *dest, const char *src);`
  - `void *memcpy(void *dest, const void *src, size_t count);`
  - All return pointers to destination (sometimes convenient, often not used).

9

C/C++ compiler directories

- /usr/… (C:\usr\...)
  - bin, m6811-elf/bin – executables
  - man – documentation
  - lib – libraries, including startup, float support, MSOE library support, etc.

10

Using the compiler, single command

- Bring up a Command Prompt and run gccvars in the normal way
- gcc
  - `-g` debugging information
  - `-mshort` int is 16-bit
  - `-Wall` useful warnings (really)
  - `-Wl,-T` ignore default linker config.
  - `-Wl,../msoe.x` send msoe.x to as
  - `-fno-exceptions` disable C++ exceptions
  - `-O2` optimize
  - `hello.cpp` file(s) to compile / asm
  - `-lmsoe` libraries to link against
  - `-o hello.elf` name of the output

11

Using the compiler, step-by-step

- For each C++ module:
  - gcc -g -mshort -Wall -fno-exceptions -O2 -S name.cpp
  - as -o name.o name.s
- For each ASM module:
  - as -o name.o name.s
- Linking
  - ld
    - -T ..\msoe.x
    - -relax *(enable certain addressing optimizations)*
    - -o main.elf
    - C:\usr\lib\gcc-lib\m6811-elf\3.3.5-m68hc1x-20050129\mshort\crt1.o *(pre-main set up)*
    - **[list of .o files or *.o]**
    - -L\usr\lib\gcc-lib\m6811-elf\3.3.5-m68hc1x-20050129\mshort -L\usr\m6811-elf\lib\mshort *(places to search for support libraries)*
    - -lmsoe -lgcc *(math and library support, as needed, etc.)*

12