ASC Programming

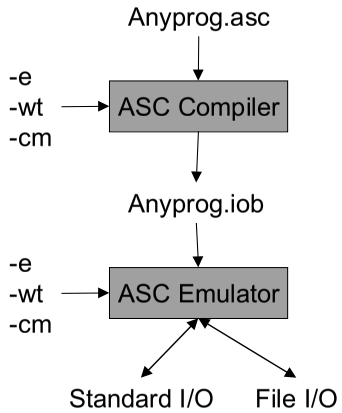
Michael C. Scherger
Department of Computer Science
Kent State University
September 27, 2002

Contents

- Software Location and Installation
- Basic Program Structure
- Data Types and Variables
- Associations and Setscope
- Input / Output
- Control Structures
- Looping
- Performance Monitor

Software

- Compiler and Emulator
  - DOS/Windows, UNIX (Linux)
  - WaveTracer
  - Connection Machine
- http://www.cs.kent.edu/~mscherge/Parallel/Software/software.htm
- Use any text editor.
  - Careful moving files between DOS and UNIX!



Software

- Example:
  - To compile your program…
    - % asc1.exe –e shapes.asc
  - To execute your program…
    - % asc2.exe –e shapes.iob
    - % asc2.exe –e shapes.iob < shapes.dat
    - % asc2.exe –e shapes.iob < shapes.dat > shapes.out

Basic Program Structure

- Main program_name
  - Constants;
  - Varaibles;
  - Associations;
  - Body;
- End;

Basic Program Structure

- Example:
  - Consider an ASC Program that computes the area of various simple shapes (circle, rectangle, triangle).
- Here is an example program…shapes.asc
- Here is the data file…shapes.dat
- Here is the output…shapes.out.txt

Data Types and Variables

- ASC has eight data types…
  - int, real, hex, oct, bin, card, char, logical, index.
- Variables can either be *scalar* or *parallel*.
  - Scalar variables are in the IS. Parallel variables are in the cells.
  - Parallel variables have the suffix "[$]" at the end of the identifier.
  - Can specify the length (in bits) of parallel variables. Default length works fine for most programs.

Data Types and Variables

```
int scalar a, b, c;
int parallel p[$], q[$],
r[$], pp[$,4];
index parallel xx[$], yy[$];
```

a+b	scalar
a+p[$]	parallel
a+p[xx]	scalar
pp[xx,b]*3+q[$]	parallel
p[xx]+pp[yy,b]	scalar
q[$]+r[$]	parallel
a+pp[xx,b]*3	scalar

- More Examples on page 9-10

Associations and Setscope

- There are no “structs” or “classes” in ASC.
  - Create an association between parallel variables and a parallel logical variables.
    - Example on page 8.
- Setscope
  - Selects or “marks” a set of active cells.
    - Example on page 15

Input / Output

- READ
  - Page 12-13.
  - Must have an ASSOCIATE statement.
  - Input file must have blank line at end of data set.
  - Read from “standard input” or from data file re-directed from command line.

Input / Output

- PRINT
  - Page 13
  - Must have an ASSOCIATE statement.
  - Does not output user specified strings.
    - I.e. User text messages.
    - Only outputs the values of parallel variables.

Input / Output

- MSG
  - Page 13-14
  - Used to display user text messages.
  - Used to display values of scalar variables.
  - Used to display a dump of the parallel variables.

Control Structures

- IF-THEN-ELSE
  - Scalar version similar to other sequential programming languages.
    - Either executes the body of the IF, OR executes the body of the ELSE.
  - Parallel version is more “sequence-like”.
    - Executes body of the IF followed by body of ELSE.
    - Masking operation.

Control Structures

- Relational operators on page 40.
- Example on page 16.
- IF-NOT-ANY on page 16-17.
- ANY on page 17-19.

Looping

- Sequential looping
  - Loop-Until
    - Example on page 20.
- Two types of parallel looping constructs:
  - Parallel For-Loop
    - Conditional is evaluated only once.
    - Example on page 21.
  - Parallel While-Loop
    - Conditional is evaluated every iteration.
    - Example on page 21-22.

Performance Monitor

- Keeps track of number of scalar and parallel operations.
  - PERFORM = 1;
  - PERFORM = 0;
  - MSG PA_PERFORM;
  - MSG SC_PERFORM;