# September 15, 2003

- Lab Access
  - Key to A-205 available at main gate and department office.
  - One Design Kit available in department office.
- Problems with first lab assignment?
- This week's laboratory: Write a Handel-C program from scratch.

# C/C++ Compilation Model

- Preprocessor (*cpp*)
  - Makes textual substitution to source code.
  - Preprocessor directives have a '#' in column 1.
    - #include
    - #define
    - #ifdef
- Compiler (*c1*)
  - Translates output of *cpp* into assembly language.
  - Text in, text out.
- Assembler (*as*)
  - Converts assembly language to object code.
  - Text in, binary object module out.
- Linker (*ld*)
  - Combines object modules to produce an executable file.
  - Binary object modules in, binary executable out.

# Handel-C Compilation Model

- GNU C/C++ preprocessor (*cpp*)
- Compiler (*handelc*)
  - Expanded source code from *cpp* in.
  - VHDL, Verilog, or EDIF out
    - Command line option for which output format
  - Text in, text out
- Other vendor tools process compiler output

# Xilinx Tools (from edifmake.bat)

```
:NODIR
ngdbuild %SDPATH% -dd . %1.edf %1.ngd
if ERRORLEVEL 1 goto ERROR
goto MAP
:UCF_EDIF
ngdbuild %SDPATH% -dd . -uc %1.ucf %1.edF %1.ngd
goto PAR
:MAP
map %1.ngd
if ERRORLEVEL 1 goto ERROR
:PAR
par -ol 4 -w %1.ncd %1.ncd
if ERRORLEVEL 1 goto ERROR
bitgen -g LCK_cycle:2 -g compress -g HswapenPin:Pulldown -g M0Pin:Pulldown -g CclkPin:PullNone -g
      ProgPin:PullNone -g DriveDone:Yes -g TdiPin:PullNone -d -w %1.ncd %1.bit
if ERRORLEVEL 1 goto ERROR
goto DONE
```

# NGDBuild (Quote from Xilinx Website)

NGDBuild performs all the steps necessary to read a netlist file in XNF, EDIF, or NGC format and create an NGD file describing the logical design (a logical design is in terms of logic elements such as AND gates, OR gates, decoders, flip-flops, and RAMs). The NGD file resulting from an NGDBuild run contains both a logical description of the design reduced to Xilinx Native Generic Database (NGD) primitives and a description in terms of the original hierarchy expressed in the input netlist. The output NGD file can be mapped to the desired device family.

# Map, PAR, and Bitgen

- Map converts a generic database file to a family-specific format (Virtex-II family, for example)
- PAR (Place and Route) maps a design to specific slices of the target device.
  - Constraint and optimization tools available for tweaking the design for speed or space.
- Bitgen generates a .bit file for downloading to the target device.
  - FTU2.exe from Celoxica can do the download to an RC200.

# Handel-C Concepts

- I/O Models
  - Platform Stream Layer (PSL)
    - RC200PSLReference.pdf
  - Platform Abstraction Layer (PAL)
    - PALUserManual.pdf, etc.
- Ports
  - For connection to FPGA pins
  - For connections to simulated I/O
- Control Structures
  - Loops
  - Functions, Macros
- Data
  - unsigned
  - RAM, ROM, WOM

# Sample Code

```
// simple.hcc
/*
 *    This is a simple Handel-C application that compiles,
 *    but optimizes down to nothing.
 */
set clock = external "C11";
set family = XilinxVirtexII;
set part = "XC2V1000-4FG456";
void
main( void )
{

}
```

# Commands for Sample Code

C:> handelc –edif sample.hcc

C:> edifmake.bat sample

- Can also be run from DK
- Change edif to verilog or vhdl to see what Handel-C generates for those HDLs.