

# bumbleBEE and Hornet: An Apian Progress Report

Gregory Wright  
Berkeley Wireless Research Center



# Outline

- Cast of Characters
- The BEE and bumbleBEE
- Hardware for the bumbleBEE
- Progress on the BEE:
  - Software for mapping circuits onto the FPGA array
  - Playing nice with Simulink
- An integrated platform for signal processing research:  
the Hornet



# Cast of Characters

- Bob Brodersen, BWRC professor
- Gary Kelson, BWRC professional staff
- Hayden So, graduate student researcher
- Kea Hunt, EECS undergraduate
- Gregory Wright, BWRC member company researcher



# The BEE Project

The BEE is a system on a chip emulator built out of Field Programmable Gate Arrays (FPGAs).

This idea isn't new, but FPGA densities have been going up fast enough to allow us to build a system that can emulate state of the art chips.

We're also helped by the fact that the design times for very complex chips (18 months to 2 years or even more!) make emulation an attractive intermediate step.



# BEE and bumbleBEE

BEE = Biggascale Emulation Engine

Because Bigga is bigga than Giga

bumbleBEE = a kinder, gentler BEE

Still bigga than Giga.

Designed to use off the shelf components as much as possible.

Still has plenty of sting.



# bumbleBEE

An array of 18 1 to 2 million gate FPGAs  
4 programmable crossbar switches (> 2000  
programmable global interconnections)

Several MB of fast SRAM

High speed board to board interconnections using  
LVDS point to point links – multiple bumbleBEEs  
can be used together to solve bigger problems.



# The bumbleBEE Hardware

The bumbleBEE is a circuit board containing:

18 FPGAs,

4 programmable crossbar switches (I-Cube MSX532),

2 or 4 11 Gbps parallel optical links (Infineon PAROLI),

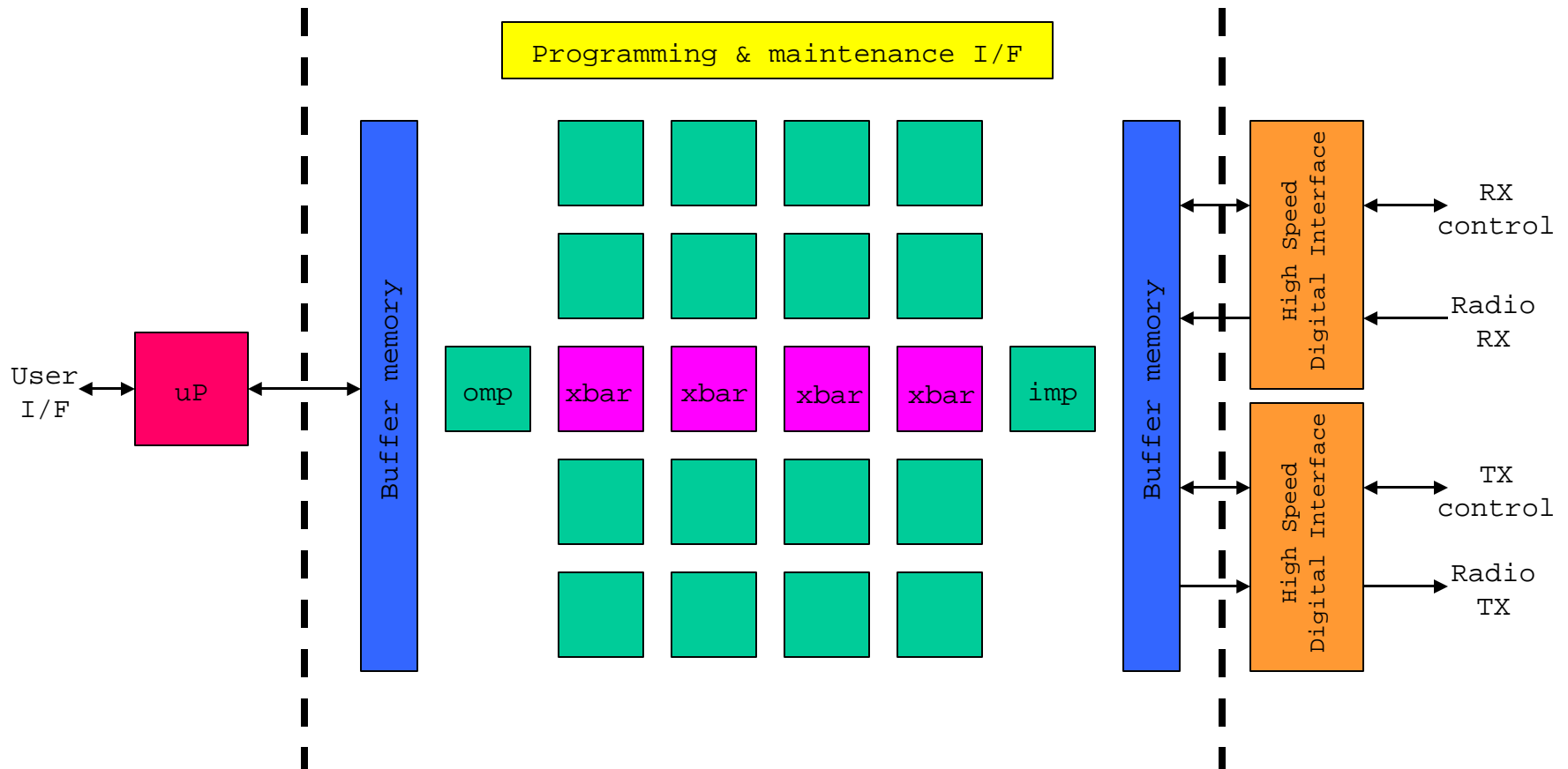
A maintenance processor running a full featured OS  
(StrongARM SA1100 under Linux),

A VXI bus interface (Interface Technology),

Blinky lights to amuse the passersby.



# The bumbleBEE Hardware, Continued





# bumbleBEE Engineering Issues

## Physical Design: Power and Cooling

The initial hardware platform will be on 9U VXI cards.

- This solves the much of the cooling problem for us, since a 9U VXI slot can handle a power dissipation in excess of 200 W, depending on the card cage.
- We are limited in the number of FPGAs per board by the total +5V power available per slot, 12 A. After conversion to 1.8 V, we have 26 A available. If more current is needed, 48 W of -5.2 V is available per slot for conversion to 1.8 V. This would provide an additional 20 A.
- For typical clock rates (50 MHz) and utilization (70 %), this limits us to about 20 FPGA chips per board (5 V supply) or 36 (5 V and -5.2 V supplies).



# bumbleBEE Engineering Issues

## Configuration and Monitoring

The VXI bus can be linked to a standard PC using an IEEE-1394 (FireWire) interface. The PC will be used to configure and debug the FPGA array.

- We can use our PC infrastructure in the lab to work with the bumbleBEE.
- Self-test (using the JTAG ports on the FPGAs and crossbar chips) is still a big issue.



# bumbleBEE Engineering Issues: Interconnection

A key question for the design of the BEE is how to connect the FPGAs to minimize the use of global routing resources (i.e, crossbar switch ports).

This is important because we need to be certain that the place and route software software can in fact map almost every circuit we're interested in to the BEE.



# bumbleBEE Engineering Issues: Interconnection

The place and route problem is naturally viewed as a graph embedding problem:

The goal is to embed the graph representing the circuit in the graph representing the emulator. The emulator graph is weighted by the cost of interconnections. We seek a minimum cost embedding.

This problem is hard.

Our approach has been to look at graph drawing algorithms, in the hope that graphs that look good represent low cost embeddings.



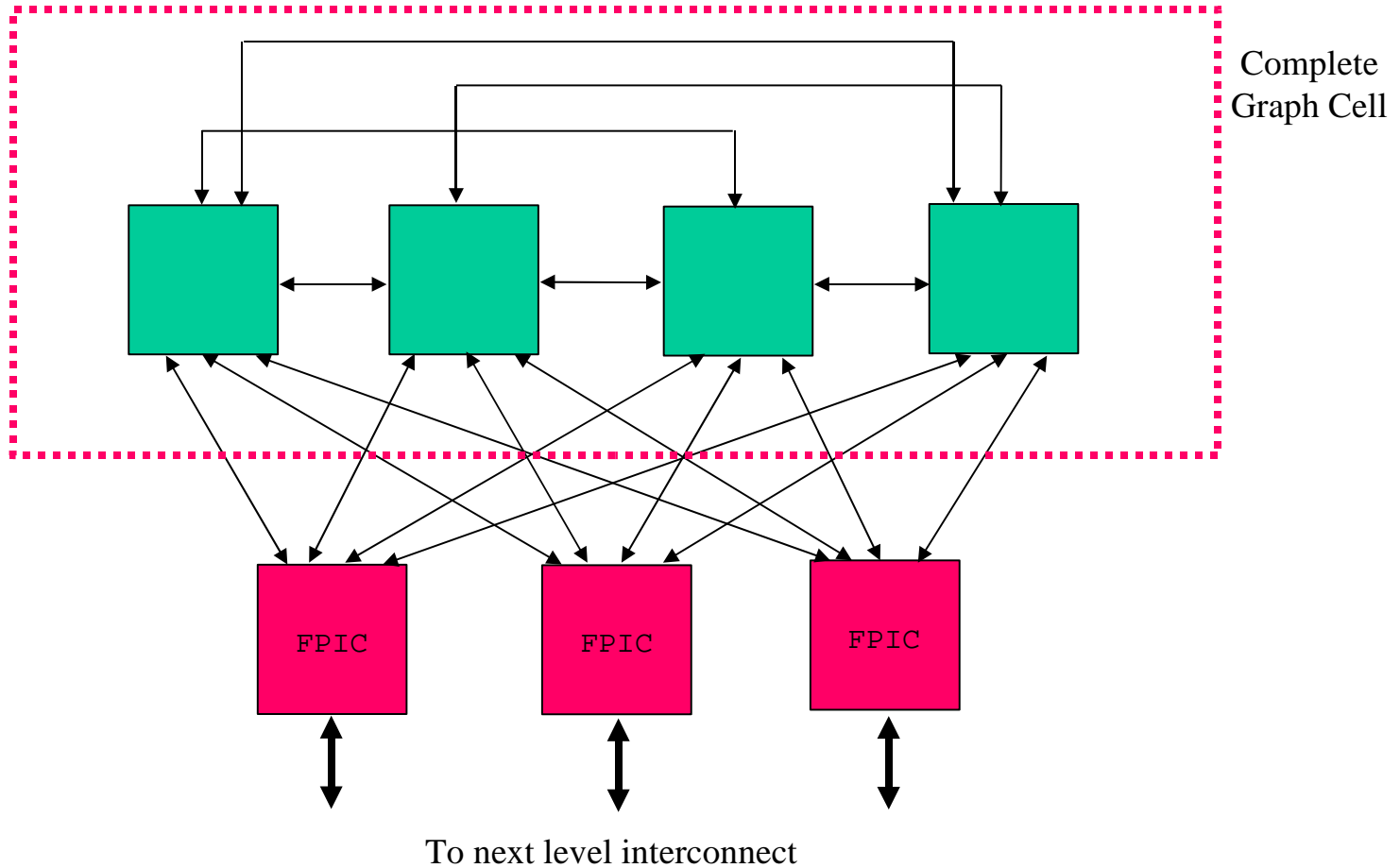
# bumbleBEE Engineering Issues: Interconnection

We had originally believed that an architecture similar to that used in FPGA based test vector accelerators would be best (a locally complete graph + partial crossbar architecture).

Work by Hayden So indicates that a somewhat different architecture (“bow interconnect”) is better for the typical communication IC.



# Local Complete Graph with Partial Crossbar



# bumbleBEE Engineering Issues

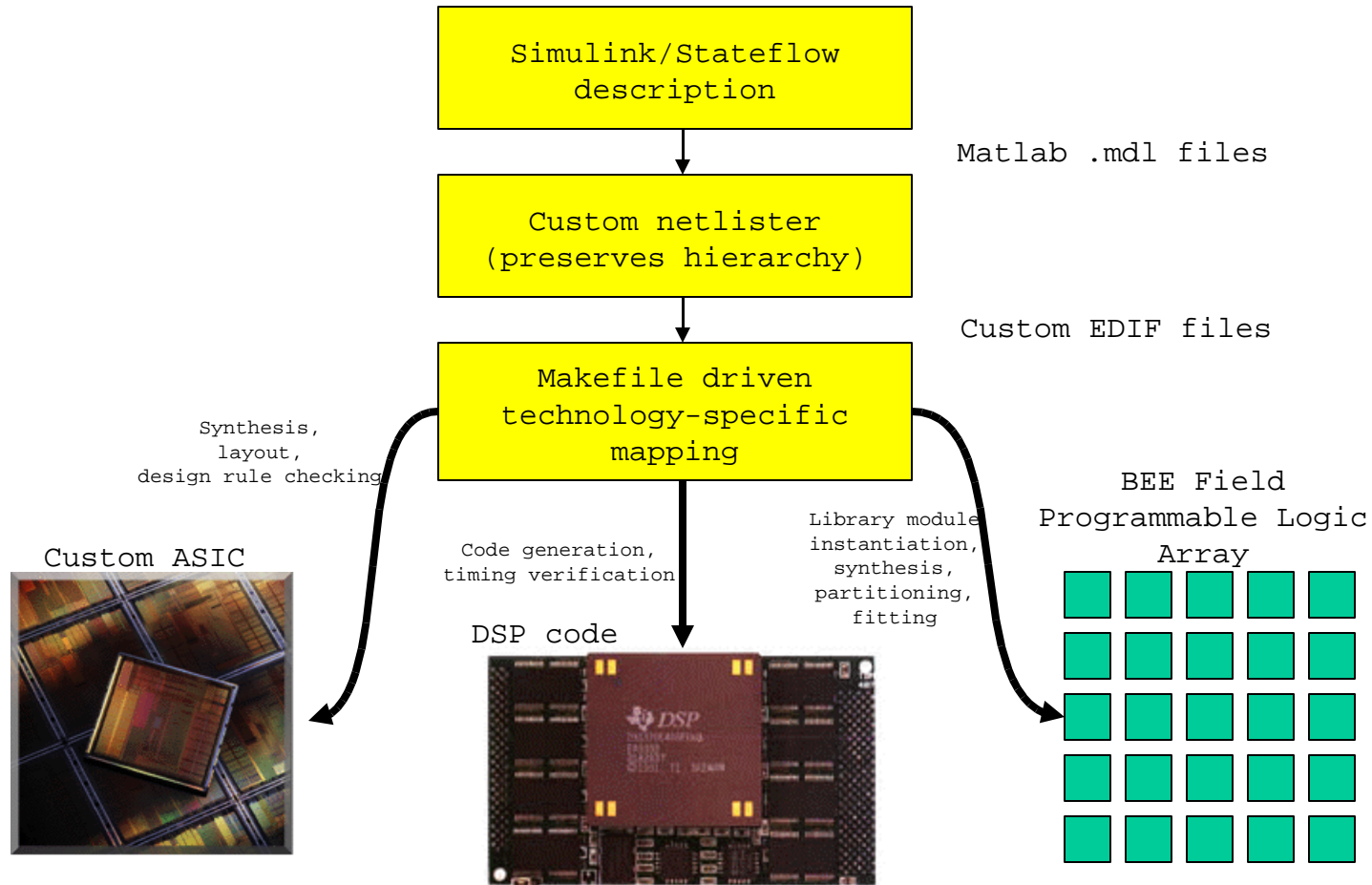
## The BEE as a circuit design tool

Our goal is to make the BEE fit into the existing BWRC full custom ASIC design flow. Notable features of this design flow are:

- Simulink/Stateflow high level design and system level simulation,
- a custom hierarchy preserving netlister, and
- an automated design processing control tool, *icmake*, which controls the invocation of the appropriate synthesis and layout tools.

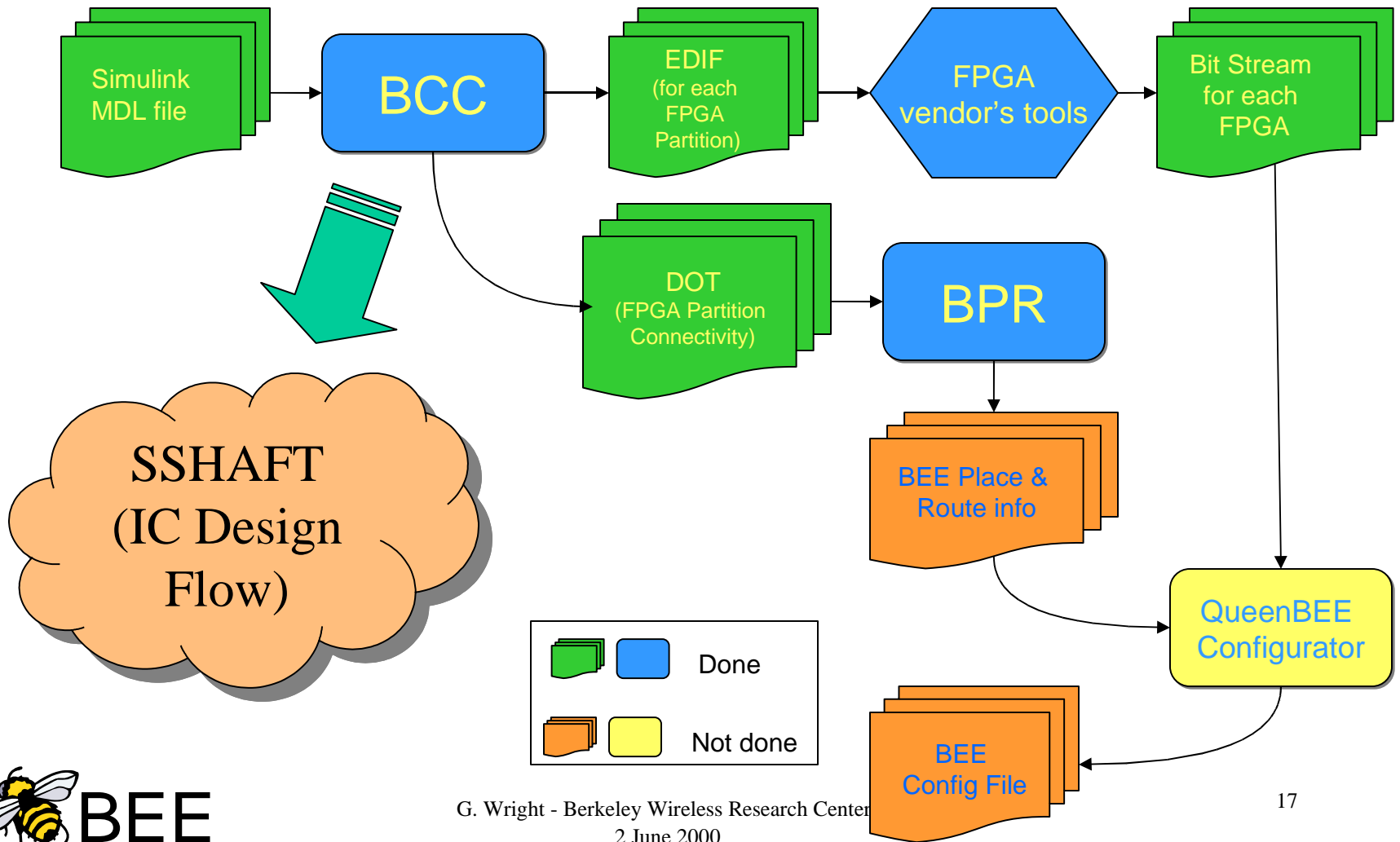


# BEE and the BWRC Design Flow





# Design Flow – What's working and what's not



# BEE and the BWRC Design Flow

- To obtain adequate performance we need a good library of signal processing components, optimized for our target FPGA architecture.
  - Good progress has been made in this area. Kea Hunt has written code for the Xilinx CoreGen program so it will build modules corresponding to the Simulink primitives used in the IC design flow.



# A BEE for the USSR: the Hornet

- The bumbleBEE is designed to be the development platform for the Universal Spectrum Sharing Radio (USSR) project.
- Lucent Technologies has donated a wide dynamic range radio design (thanks Paul!) to be used as the front end.
- bumbleBEE + Radio = more buzz = Hornet





# Hornet Status

- Have working drawings, schematics, BOM.
- Need to add A/D and D/A interfaces to radio boards to allow a digital connection to the bumbleBEE. This will require a 2<sup>nd</sup> downconversion to a lower IF.
- Still a question of whether to use an optical or electrical digital interface to the bumbleBEE (a cost versus reliability issue).
- New radio board will probably be used to try out a new vendor for board fabrication and assembly.



# BEE Project Status, revisited

- Initially targeted to support the Universal Radio project in the BWRC.
  - *Still true.*
- We are looking at the design flow first and assembling all of the software components we need before committing to hardware.
  - *A good choice. We have learned a lot about how to interconnect the FPGAs to support data flow designs. We have more confidence that the BEE will be an useful tool for the communication signal processing problems we want to solve.*
- Goal is a workable design flow (integrated into the BWRC IC flow) in 2Q2000 and hardware in 3Q2000.
  - *A month or so behind schedule.*



# Summary

Good progress is being made, especially on understanding how to integrate the BEE into the BWRC custom IC design flow. There is still plenty to be done, but the results to date are giving us confidence of success.

We will be busy working on the BEE!

