

***Teaching and Teaching Computer Architecture:  
Two Very different topics  
(Some Opinions about each)***

***Yale N. Patt  
teacher, The University of Texas at Austin***

***WCAE  
San Diego, June 8, 2003***

## ***Visions (Re: Education)***

- \* Distance Learning Produces  
Better Education, Not Cheaper Education***
- \* We Pay Teachers Enough That  
Those Who Would Opt For This Career  
Don't Opt For Medical School Instead***
- \* We Teach High School English Teachers  
Enough English That  
Students At The University Can Write  
Two Consecutive Coherent Sentences***
- \* We Get Past This Insane Preoccupation  
With Political Correctness,  
So We Can Get On With The Business  
Of Teaching and Learning.***
- \* We Stop Canonizing  
The Use Of High Tech In Education.  
Bad Pedagogy Is NOT Good Pedagogy  
if Draped in Technology.***
- \* We Stop Rewarding Memorization Ability,  
So Maybe Students Will Learn To Think,  
. . . And Perhaps Understand***

# ***My Ten Commandments of Good Teaching***

- \* Know the Material***
- \* Want to teach***
- \* Genuinely respect your students and show it***
- \* Set the bar high; students will measure up***
- \* Emphasize understanding; de-emphasize memorization***
- \* Take responsibility for what is covered***
- \* Don't even try to cover the material***
- \* Encourage interruptions; don't be afraid to digress***
- \* Don't forget those three little words***
- \* Reserved for future use***

## ***What is Important?***

- \* Top-down design,  
Bottom-up learning for understanding***
- \* Abstraction is vital, but...***
- \* Not bottom-up,  
but "motivated" bottom-up***
- \* Engineering is about DESIGN,  
first understand the components***
- \* From Concrete to Abstract  
(Dijkstra notwithstanding)***
- \* Cut through protective layers***
- \* Memorizing is not understanding***
- \* Students do better working in groups***

# **"Motivated" Bottom-Up**

## ***Solving Problems with Computers***

---

### **1. Overview**

---

#### **11. Overview of HLL Programming**

---

#### **16. Linked lists**

#### **15. Pointers**

#### **14. Recursion**

#### **13. Functions**

#### **12. Variables, Operators, Control**

---

#### **6. Programming/Debugging**

---

#### **10. Data Structures (Stack)**

#### **9. Subroutines, Traps**

#### **8. Physical I/O**

#### **7. Assembly Language, Assembler**

---

#### **2. Bits, Datatypes, Arithmetic, Logic**

---

#### **5. LC-2**

#### **4. Von-Neumann Model**

#### **3. Transistor, Gate, Memory, Finite State Control**

---

### ***Electronic Circuits and Below***

# **High Tech**

## **Some Uses**

- \* **Email**
- \* **Web site**
- \* **Power Point**
- \* **Document Reader**
- \* **Animations**
- \* **Plato, vintage 2003**
- \* **Clever attendance mechanism**
- \* **Other bookkeeping**
- \* **Text+Voice**  
**(WOW Factor, see Shriver's CDROM)[3]**

## **Some Caveats**

- \* **Baseline Power Point**
- \* **Cost**
- \* **Extemporaneous Effect**
- \* **Visual/voice disconnect**
- \* **Attendance vs. Participation**

# ***Some Fundamentals of Computer Architecture***

- \* The transformation hierarchy***
- \* Three parts of a Microarchitecture***
- \* The DSI***
- \* IPC vs. cycle time***
- \* Partitioning***

***Problem***

---

***Algorithm***

---

***Program***

---

***Instruction Set Architecture (ISA)***

---

***Microarchitecture***

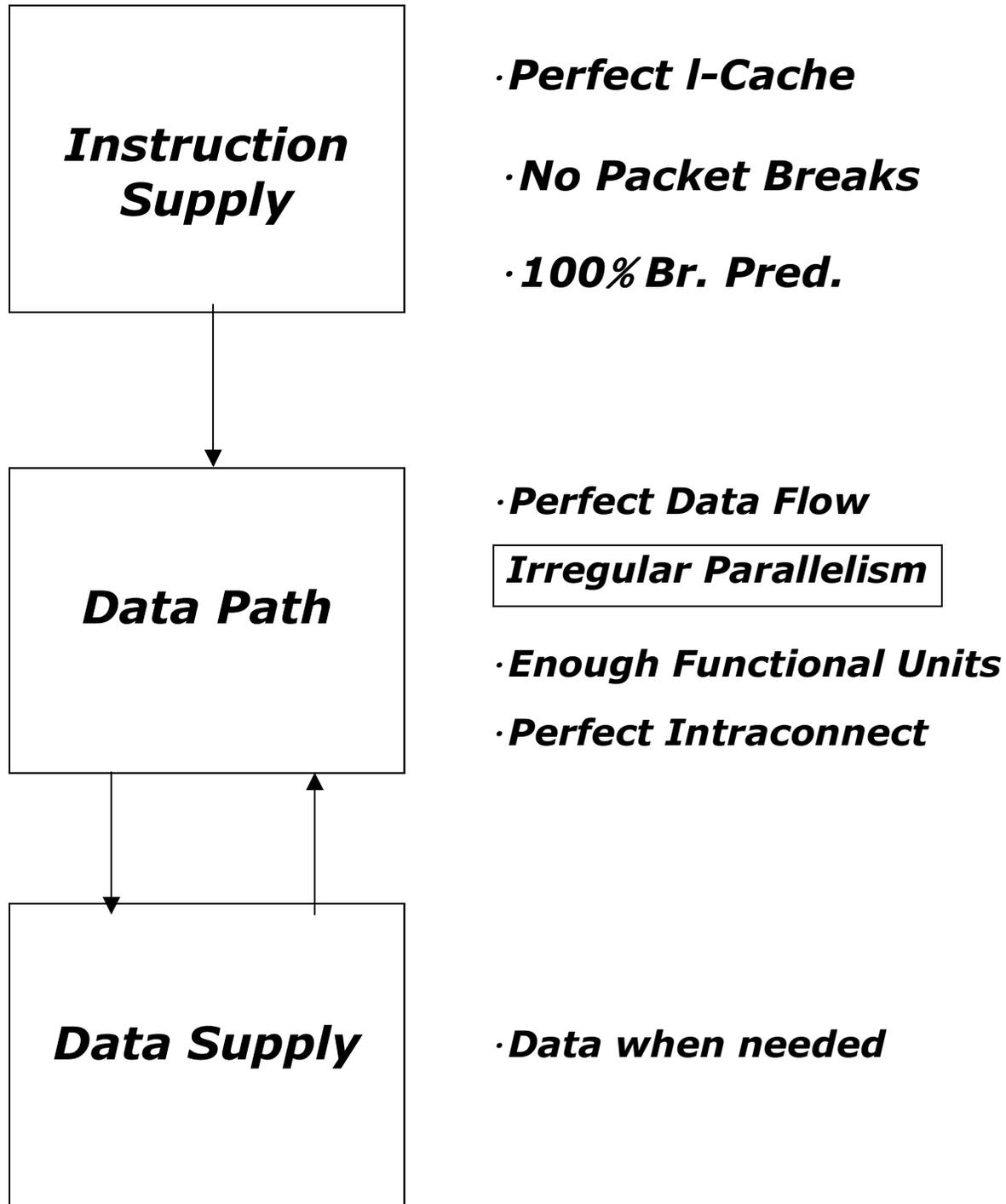
---

***Circuits***

---

***Electrons***

# ***Microarchitecture (The Requirement)***



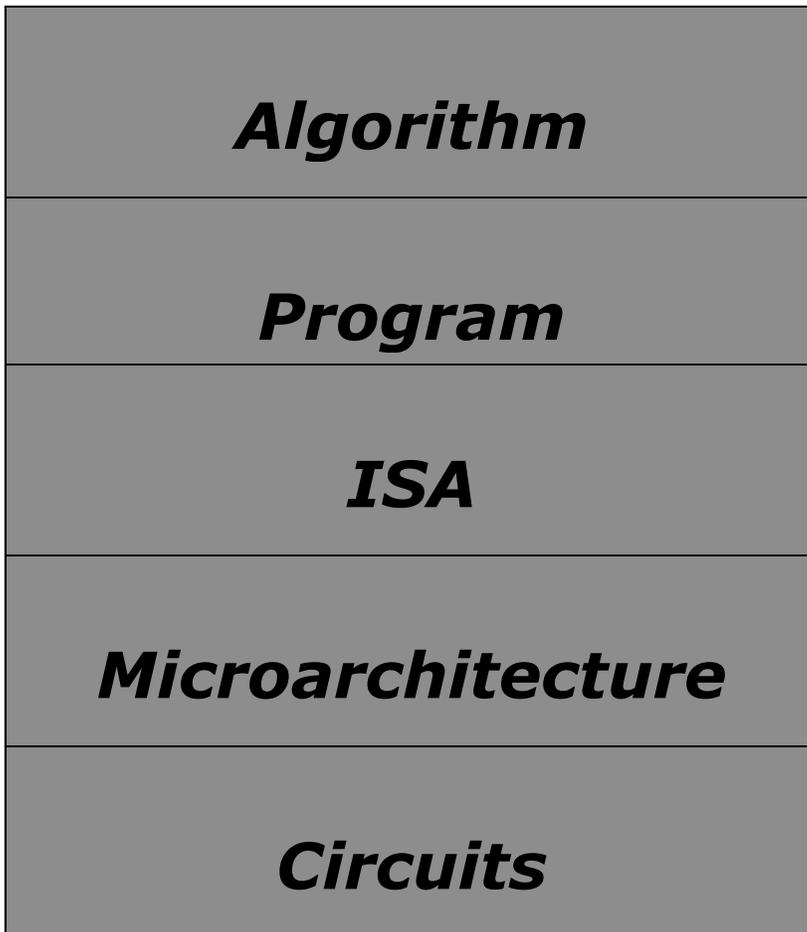
## ***Some Concerns***

- \* Focus on Measurements***
- \* Use of Simulations***
- \* Real ISA vs. Concocted ISA***

## ***The Microprocessor Ten Years From Now (perhaps)***

- \* the new data path***
- \* internal fault tolerance***
- \* asynch and synch co-existing***
- \* different cycle times for different functions***
- \* SSMT (aka helper threads)***
- \* Block-structured ISA***
- \* uarch support for CAD***
- \* greater use of microcode***
- \* greater impact of the compiler***
- \* compiler/uarch communication***

***Problem***



***Electrons***