# Program Your “Stupid” Brain to learn or do (Make) Anything?

Praseed Pai K.T.

Presented in “Maker Party” Kochi event held @ CUSAT on 21/09/2014

# A Proverb to keep us sane

- "**A science is any discipline in which the fool of this generation can go beyond the point reached by the genius of the last generation.**" - Max Gluckman, a South African Anthropologist

# What you need to “re-program” your brain?

- Perils of Bottom up Learning
- Learning Blocks

# PERILS OF BOTTOM UP LEARNING

# Bottom up Learning

- Learn Fundamentals, go to the lowest level of abstraction, iteratively learn more and more advanced topics to reach “Knowledge Bliss”
- Excellent for learning things, when you do not have people around who knows what you are learning
- A method apt suited for creating future “Scientists”, where as most of the people are “Craftsmen” (Artists and Engineers)
- Satisfies the need for “Cognitive Closure” and a Self satisficing exercise
- Was good during the pre-internet era and schools and colleges emphasize it

# Bottom up learning of Compilers - Perils

- Compiler Construction – started with Dragon Book first edition
- Learned how a Lexical Analyzer works
- Learned how Table Driven Parser works
- Learned Lex and Yacc, wrote scores of small programs before attemting a real compiler for a toy language
- Took three years to learn these things
- Using my “SLANGFORDOTNET” ( http://slangfordotnet.codeplex.com) compiler, in a matter of four months, a guy learned more than what I learned in three years and later went back to the theory to complete the loop

# Bottom up Learning of Computer Graphics - Perils

- Learned about Pixels and Image synthesis
- Color Models, VGA hardware
- Bresenham’s Line algorithm, Circle, Fill algorithms, wrote co-ordinate transformation routines, Projections,Shading etc by writing code from the scratch, by consulting various books
- Spent close to four years before I could write some really good graphics program
- All that can be learned during a weekend by writing a Clock and a Atlas of the world mapped on to a “Sphere” to create a globe

# Bottom up Learning of Electronics - Perils

- Purchased “Art of Electronics” and started reading it
- Tried to read every article in EFY
- Worked for Electronic CAD software company
- Purchased scores of book on Electronics and reached no where in fifteen years
- Times Change. In 2013,Found out a “guru” who was willing to teach me in a “exchange program”. He is a bottom up electronics guy and wanted to progress towards Embedded Micro processor based system from Microcontrollers. I taught him GCC/C++ under Linux and he taught me Electronic components and circuits
- In a matter of some days, I could create some basic circuits and got enough courage to go and shop in a Eleftronic component shops.
- This had a dramatic impact (discrete electronic circuit construction) and things began to fall in place for me. Now, I can reason about most electronics/embedded systems from Discrete component circuits and PCB, MicroController based Embedded Systems, Micro Processor based Embedded Systems, Hardware concurrency (FPGA) and ASIC etc.

# LEARNING BLOCKS

# Learning Blocks

- Mental Blocks are there for everyone!
- Framing, Fundamental Attribution Errors, Loss Aversion etc.
- Religious and Cultural Bias
- Aversion to certain things because of past experiences
- Lack of a “Stupidity Ring”
- Lack of Peer Pressure

# SOLUTION

TOP DOWN LEARNING (AKA LEARN BY DOING)

# Teaching Java to people who are specialized in Humanities

- Commerce graduates can deal “Human centric Systems” well.
- Top Down learning suites them. In a bottom up learning model, they loose interest
- They want “tolerance” for errors
- Started with Java SWING, JDBC,XML and made them implement Inventory systems, Expense management, Financial Accounting Systems

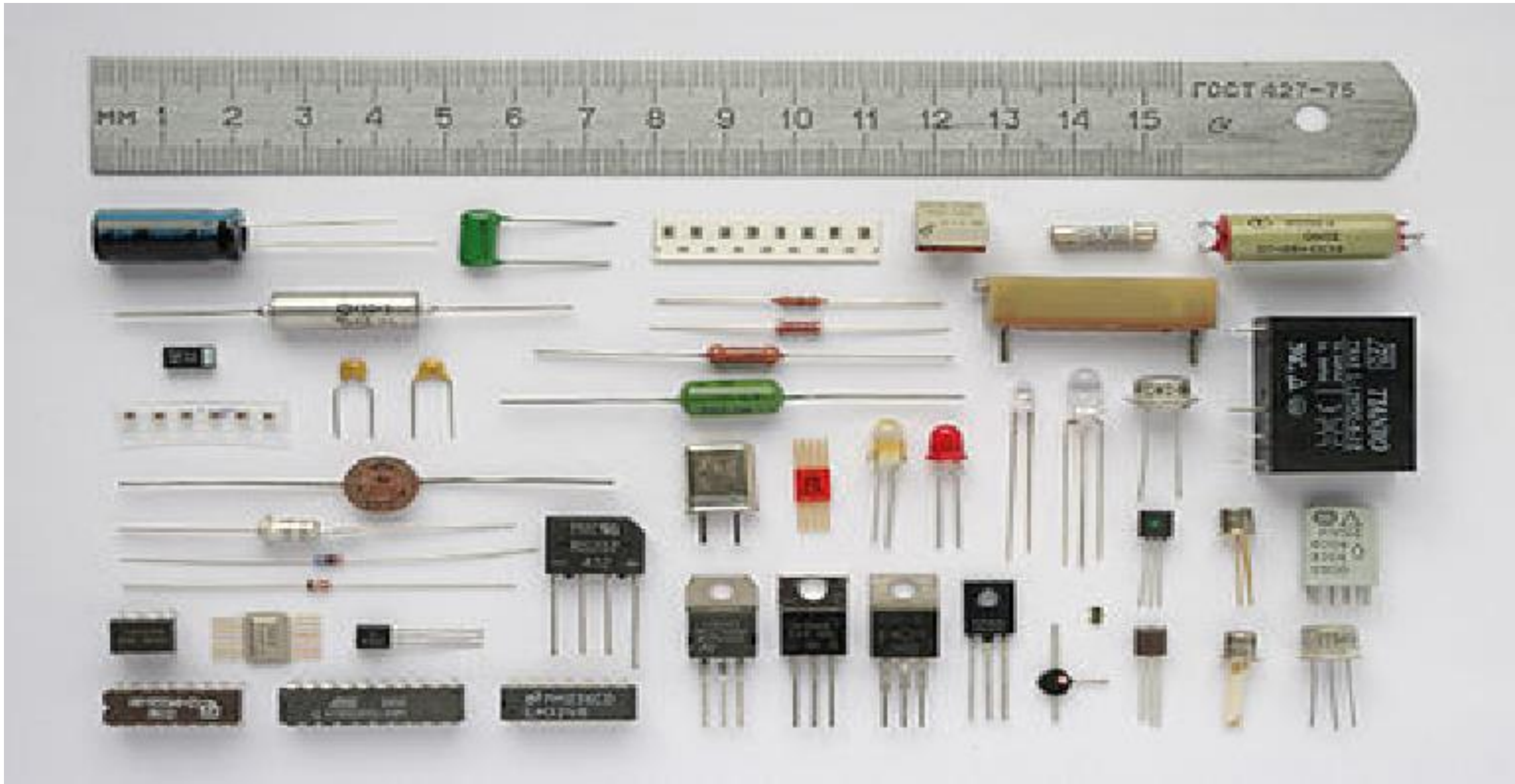
# Making good programmers out of "Smart Dropouts"

- Give them "Loaded" system building tasks
- Encourage Comparative Learning
- Help them to avoid "fixation" with things by unsettling their comfort zone
- Ask them to speak about what they know!
- Encourage them to Blog

# A “Total”Electronics/Embedded Self Learning Conceptual Toolkit

- Five Levels of Embedded Systems Competency
- Level 0 - Discrete Components
- Level 1 – Discrete Circuits/PCB systems
- Level 2 - Discrete Circuits/Mup systems
- Level 3 - Micro Processor based Systems
- Level 4 - VHDL/FPGA
- Level 5 - VHDL/ASIC – Custom Fabrication

# Level 0 – Discrete Components



# Level 1 – Discrete Circuits/PCB

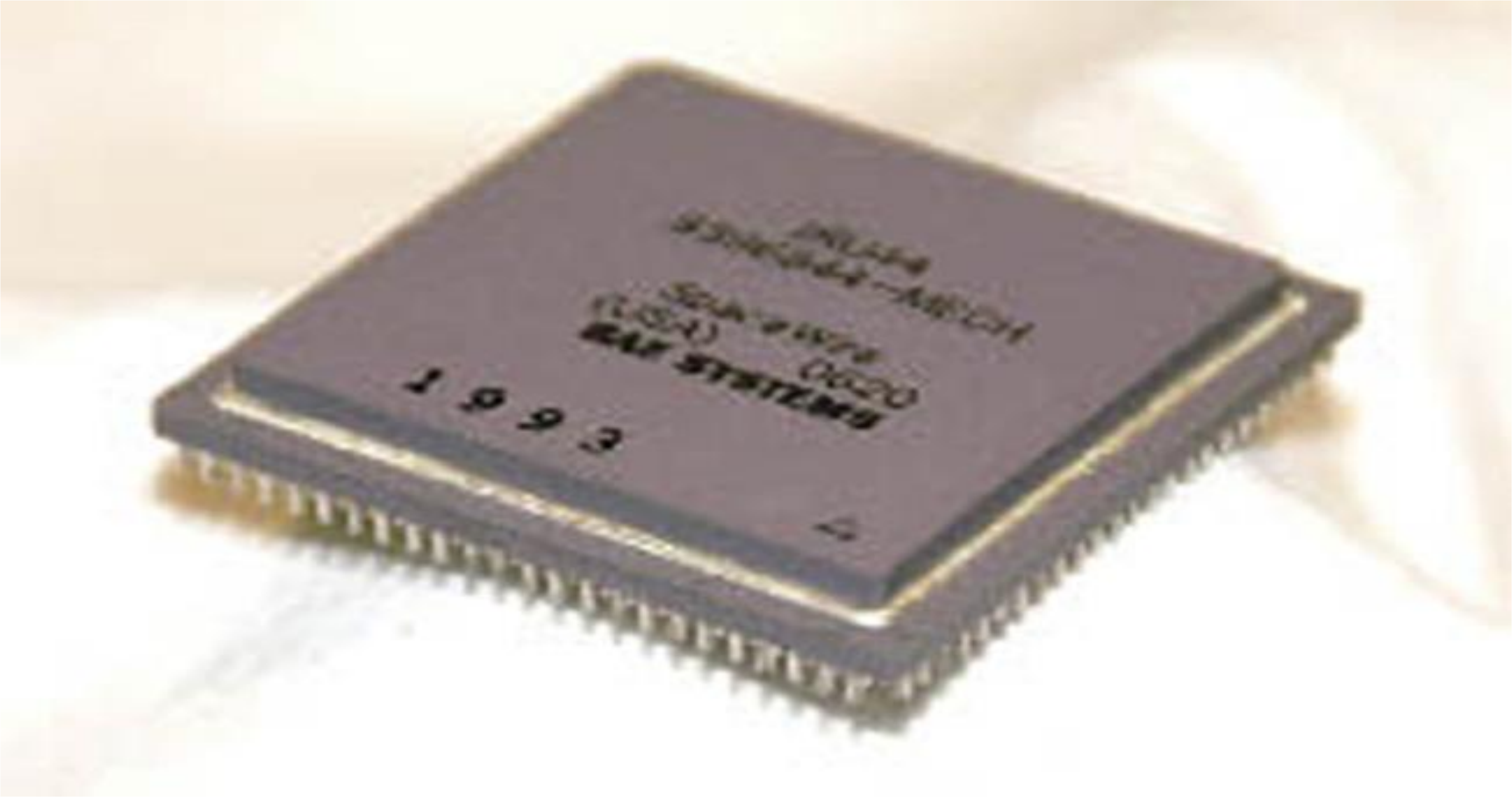
# Level 2 – MicroController based Embedded Systems (AVR/PIC/8051/Ardiuno)

# Level 3 -Microprocesser based Embedded Systems

- ARM , x86 and Rasberry Pi

# Level 4 – VHDL/FPGA systems

# Level 5 – VHDL/ASIC



# Questions?

- praseedp@gmail.com
- http://praseedp.blogspot.com