

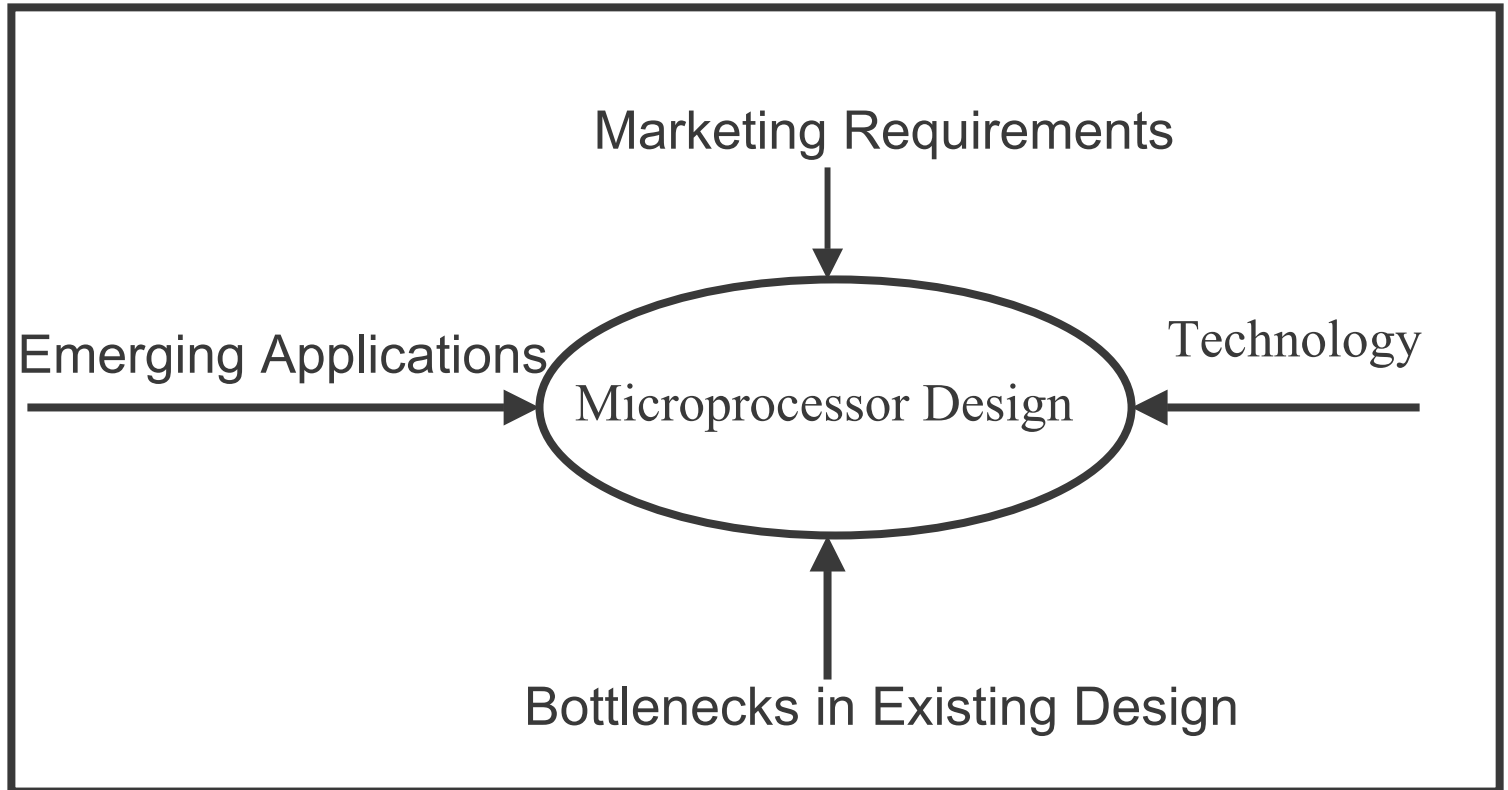


# **CHARACTERIZATION OF EMBEDDED WORKLOADS**

**Course Project**  
*Embedded Software Systems*

**Ajay Joshi**  
**University of Texas, Austin**  
**March 24, 2004**

# MOTIVATION



*“ I think there is a world market for maybe five computers ”*

- Thomas Watson, IBM Chairman, 1943

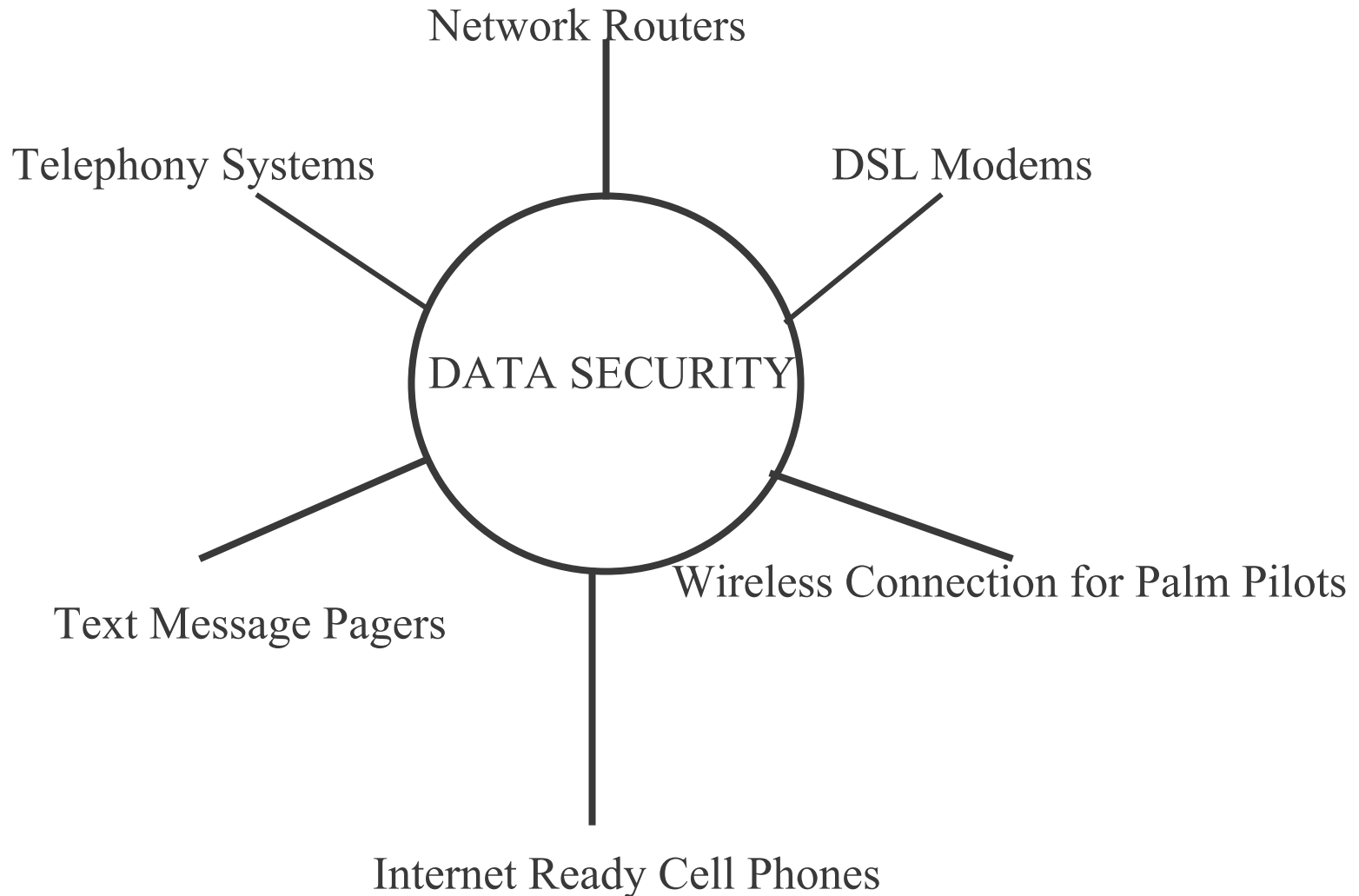
*“There is no reason for anyone to have a computer in his home”*

- Ken Olson, DEC President, 1977

# SECURITY APPLICATIONS

## An Emerging Embedded Workload

---



# PROJECT CONTRIBUTIONS

## What is the deal?

---

### Problem

- ◆ Characterize security applications & create a program behavior model

### Deliverables

- ◆ Inherent parallelism in security applications
- ◆ Micro-architecture independent characteristics of security applications
- ◆ Analysis of applications characteristics and its impact on architecture design



# PREVIOUS WORK – I

---

## Parallelism Limit Studies

- **Dataflow Graphs** [Arvind et al., 1988]
  - \* Results not biased by implementation constraints
  - \* Parallelism = Inherent + Specific Implementation
- **Instruction Trace** [Wall, 1991]
  - \* Speculative Execution, Memory Disambiguation etc.
  - \* Speedup of 7-60 observed
- **Abstract Machines** [Wilson & Lam, 1992]
  - \* Seven Abstract Machines
  - \* Control flow is complex and highly data dependent
  - \* Evaluates three techniques useful to relax constraints



# PREVIOUS WORK - II

## ◆ General Purpose Workloads

\* Scientific, Commercial, Web, Multimedia, and Graphics

## ◆ Embedded Workloads

\* Network, Multimedia, and Telecommunication

## ◆ Workload Characteristics

\* Micro-Architecture Dependent & Independent

Instruction Mix	Static & Dynamic Size	Basic Block Size
Parallelism	Computation : Communication	Locality
Branch Predictability	Operating System Vs User Code	Dependency Distance
Cache Miss Rate	Branch Prediction Accuracy	Cycles Per Instruction
Throughput	Instructions Per Cycle	Power



# METHODOLOGY

---

## Selecting Representative Workload

- ◆ Encryption / Decryption algorithms are at the heart of security applications
  - Pretty Good Privacy
  - Blowfish
  - Rijndael
  - Secure Hash Algorithm

## Inherent Parallelism

- ◆ Abstract application from specific implementation
- ◆ Model algorithms using SDF or BDF in Ptolemy

## Micro-Architecture Independent Characteristics

- ◆ Synthesize Programs / Compile programs for ARM ISA
- ◆ Instrument *Strong-ARM* cycle accurate simulator







# KEY REFERENCES

---

## Measuring Parallelism in Programs

- [1] Arvind, D. Culler, and G. Maa, "Accessing the benefits of Fine-Grain Parallelism in Dataflow Programs", *The International Journal of Supercomputer Applications*, 2(3), November 1988.
- [2] D. Wall, "Limits of Instruction Level Parallelism", In *Proceedings of ASPLOS-4*, volume 26, pp. 176-189, April 1991.
- [3] M. Lam, and R. Wilson, "Limits of Control Flow on Parallelism", In *Proceedings of the 19<sup>th</sup> Annual International Symposium on Computer Architecture*, Gold Coast, Australia, May 19-21, 1992, pp. 46-57.
- [4] Ramakrishna Rau, J. Fischer, "Instruction Level Parallel Processing: History, Overview, and Perspective", *The Journal of Supercomputing* 7(1-2), 1993.

## Workload Characterization

- [5] D.J. Kuck, Y. Muraoka, and S.C. Chen, "On the Number of Operations Simultaneously Executable in FORTRAN like Programs and their Resulting Speedup", *ACM Transactions on Computers*, C-21: 1293-1310, December 1972.
- [6] K. Agarwala, R.M. Bryant, J. M. Mohr, " An approach to the workload characterization problem", *Computer* 9(6), pp. 18-32, June 1976.
- [7] L. John, P. Vasudevan, and J. Sabarinathan, " Workload Characterization: Motivation, Goals, Methodology and Methodology", In the *Workload Characterization: Methodology and Case Studies*, IEEE Computer Society, edited by Lizy John and A.M.G. Maynard, 1999, ISBN 0-7695-0452-3.
- [8] J. Burke, J. McDonald, T. Austin, "Architectural support for fast symmetric-key cryptography", *Proceedings of the 9<sup>th</sup> International Conference on Architectural Support for Programming Languages and Operating Systems*, pp 178-189, December 2000.





**Thank You.**

**Q & A Time**