The Hobbit Microprocessor for Personal Communicators

Robert J. Scavuzzo

AT&T Bell Laboratories, Inc.
Allentown, Pennsylvania

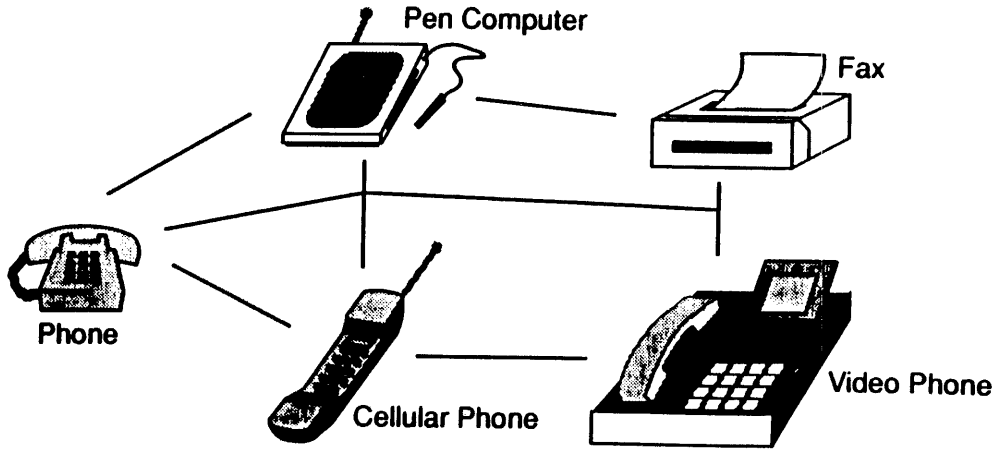
Hot Chips Symposium
August 10, 1992

Agenda

- Personal Communications
- Why Hobbit
- Hobbit History
- CRISP/Hobbit Comparison
- Hobbit Code Density
- Hobbit Performance
- Power Reduction

AT&T

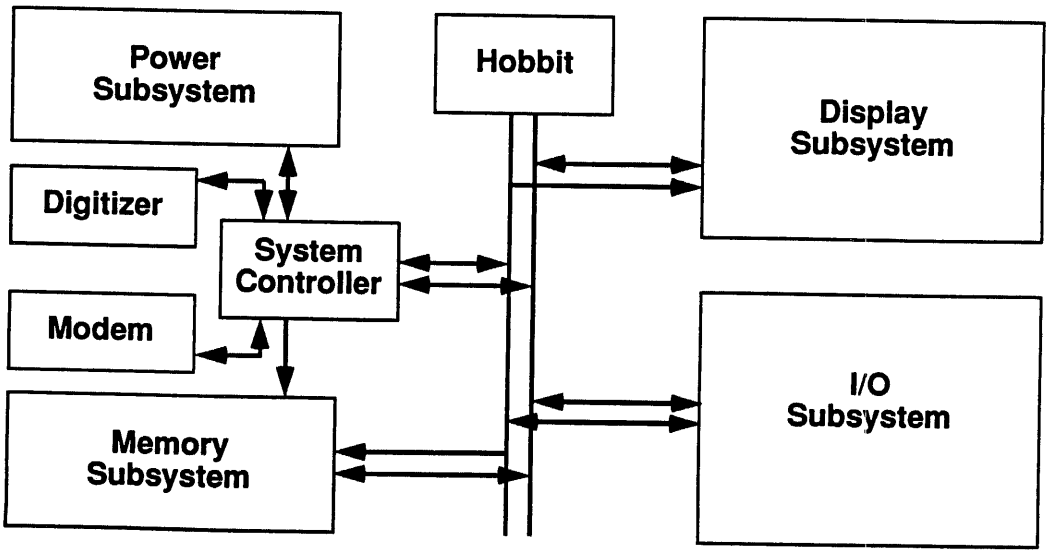
The AT&T Personal Communications Vision



Anytime, Anyplace Access to Voice, Data, Fax, Video

AT&T

A Personal Communicator Design



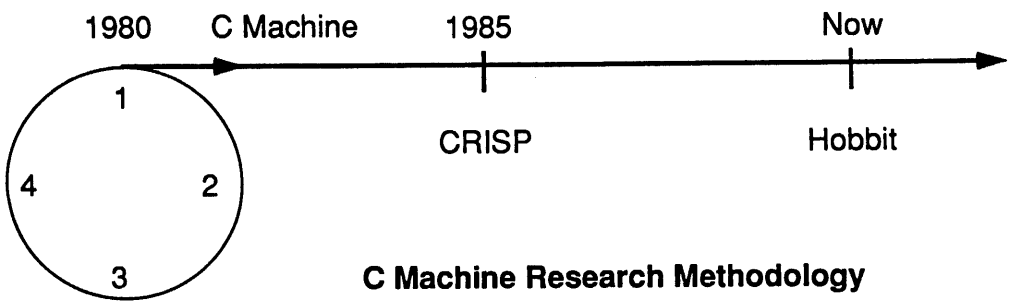
AT&T

Microprocessor Attributes Needed for PCS

- **High Code Density**
- **High Performance**
- **High Integration Level**
- **Low Power Dissipation**
- **Low Cost**

AT&T

Hobbit Evolution



C Machine Research Methodology

1) Propose an Architecture
2) Create C Compiler
3) Develop Simulator
4) Evaluate Performance

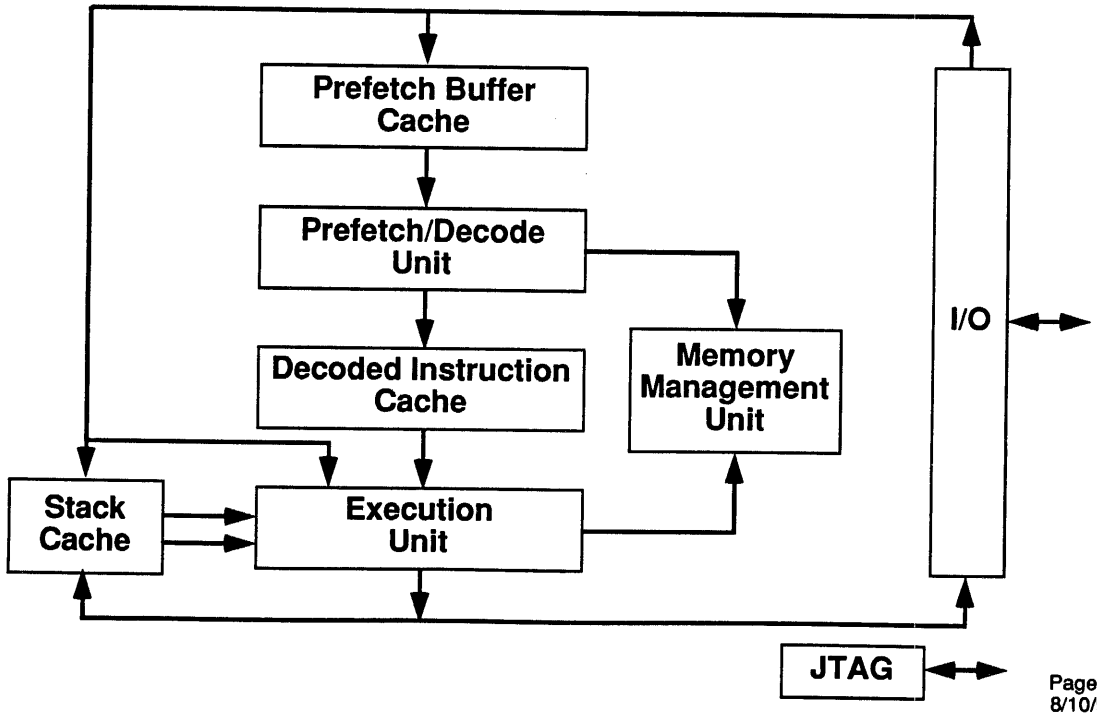
CRISP/Hobbit Architecture
High Level Attributes

- CISC Like
  - Memory-to-Memory Architecture
  - Variable Instructions
  - High Code Density
- RISC Like
  - Small Instruction Set
  - Pipelined Architecture
  - Single-Cycle Execution
  - High Performance

3-STAGE DECODE
3-STAGE EXECUTE

The CRISP/Hobbit Architecture Combines the Best of CISC and RISC

Hobbit
– Functional Block Diagram –



AT&T

Hobbit Microphotograph

AT&T

CRISP ⟶ Hobbit Changes

Technology/Silicon Design:

CRISP	Hobbit
• 5 V	• 2 V to 5.0 V
• 12 MHz, Selected	• 8 MHz to 30 MHz Operation
• No Standby Mode	• Low Power Standby Mode

Architecture:

• Big Endian Data Byte Ordering	• Big/Little Endian Data Byte Ordering
• 512 Bytes, Direct Mapped, Encoded Instruction Cache	• 3K Byte, 3-way Set Associative, Encoded Instruction Cache
• 128 Byte Stack Cache	• 256 Byte Stack Cache
• No MMU/TLBs	• MMU/TLBs
• No Data Loop Back	• Data Loop Back

AT&T

Code Density

- High Code Density
  - Memory-to-Memory Architecture
  - Variable Length Instructions
  - Carefully Crafted Instruction Set

AT&T

Code Density
Distribution

Instruction Size Distribution for PCC

2-Byte	15,000	68%
6-Byte	6,000	27%
10-Byte	1,000	5%

AT&T

Code Density
PCC Comparisons

Size Comparisons of Cross Compiled Hobbit PCC

Machine	Bytes	Relative
VAX	72K	1.0
Hobbit	74K	1.0
M68000	86K	1.2
R3000	128K	1.8
IBM 370	134K	1.9
SPARC	141K	1.9

AT&T

CRISP/Hobbit
Performance Related Features
– 1 –

- **High Code Density**
- **Synchronous but Independent Prefetch Decode and Execution Units**
- **Single Cycle Instruction Execution**
- **Encoded and Decoded ICs**
- Stack Cache
- Fast Procedure Call
- Branch Folding/Branch Prediction
- Read Cancelling with Two Stage Bypassing
- MMU and TLBs
- Graceful Performance Degradation with Increasing Memory Wait States

AT&T

CRISP/Hobbit
Performance Related Features
– 2 –

- High Code Density
- Synchronous but Independent Prefetch Decode and Execution Units
- Single Cycle Instruction Execution
- Encoded and Decoded ICs
- **Stack Cache**
- **Fast Procedure Call**
- Branch Folding/Branch Prediction
- Read Cancelling with Two Stage Bypassing
- MMU and TLBs
- Graceful Performance Degradation with Increasing Memory Wait States

AT&T

Stack Cache

- **A Different Approach to Register Allocation**
- **Automatically Maps the Stack onto a Circular Buffer of Registers**
  - **Looks like Registers to Hardware**
  - **Looks like Cache to Software**
- **Buffer Maintained by Head and Tail Pointers**
  - **Stack Pointer**
  - **Maximum Stack Pointer**
- **Stack Cache Hit if SP ≤ Address < MSP**
- **Low Order Virtual Address Bits Used as Index**

AT&T

Stack Cache Results

- Captures 80% of Data References
- Three Times Fewer References than VAX PCC
- Best of Registers
  - Fast Access
  - No Tags
  - No Tag Compares
- Best of Cache
  - Better Allocation than Software
  - Simplifies Compiler Design
  - Can Hold Strings, Arrays
  - Transparent of Software ⇒ Scalable

AT&T

Fast Procedure Call

- Four Instructions Used in Procedure Call
  - Call: (1 Clock Tick)
    Places Return Address in SC and
    Branches to Target Address
  - Enter: (1 Clock Tick Minimum)
    Allocates Space for New Procedure
  - Return: (2 Clock Ticks)
    Deallocates Current Stack Frame
  - Catch: (1 Clock Tick Minimum)
    Guarantees SC is Filled
- Procedure Calls are Typically 5 Clock Ticks

AT&T

CRISP/Hobbit
Performance Related Features
– 3 –

- High Code Density
- Synchronous but Independent Prefetch Decode and Execution Units
- Single Cycle Instruction Execution
- Encoded and Decoded ICs
- Stack Cache
- Fast Procedure Call
- **Branch Folding/Branch Prediction**
- Read Cancelling with Two Stage Bypassing
- MMU and TLBs
- Graceful Performance Degradation with Increasing Memory Wait States

AT&T

Branch Problems

- **Tend to "Break" Pipeline**
- **RISC Machines Typically use Delayed Branch**
  - **1 Cycle for Branch**
  - **1 - 2 Cycles for nop's**
  - **Problems with short Basic Blocks**

AT&T

Branch Folding

- ~ 1/3 of Instructions are Branches
- Branch Prediction is ~ 90% Accurate
- PDU Can Simultaneously Decode up to Two Instructions
- In Branch Folding, Target Address of Leading Instruction can be replaced by Target Address of Following Branch Instruction
- Branch Instruction is Executed in "Zero" Time
- Conditional Branches Provided for
  - Alternate Next-Address Field
  - Branch Prediction
  - Branch Spreading
- Reduces Number of Instructions Needed to Execute Program
- MIP Rate > Clock Rate

AT&T

CRISP/Hobbit Performance Related Features
– 4 –

- High Code Density
- Synchronous but Independent Prefetch Decode and Execution Units
- Single Cycle Instruction Execution
- Encoded and Decoded ICs
- Stack Cache
- Fast Procedure Call
- Branch Folding/Branch Prediction
- **Read Cancelling with Two Stage Bypassing**
- MMU and TLBs
- Graceful Performance Degradation with Increasing Memory Wait States

AT&T

Read Cancelling

- Dependences Cause Problems for Pipelines

  a ⟵ b + 1;

  c ⟵ a + 3;

- Hazards are Resolved by Bypassing
- Read is Cancelled with Bypass Occurs
- Hobbit Performs Better with Hazards

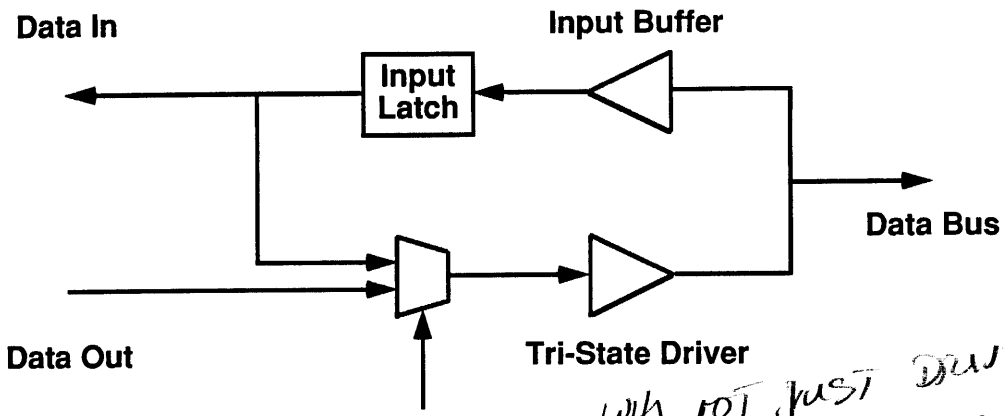
AT&T

Low Power Dissipation

- Low Voltage Design
- DC Standby Mode
- Integrated Resources
  - Caches
  - MMU/TLBs
- Reduced I/O
  - Caches
  - Read Cancelling
  - Stack Cache Architecture
- High Code Density
  - Fewer Text I/O Transactions
  - Less System RAM
- Data Loop-Back
  - Resistor-less System Bus

AT&T

Data Loop-Back



WHY NOT JUST DRIVE DATA GOING OUT

AT&T

Hobbit for Personal Communicators

- High Code Density
  - Less RAM
  - Lower Power Consumption
  - Lower System Cost
- High Performance
  - Faster System Response
  - Enables Use of CPU Intensive Applications
- High Integration
  - Lower Power Consumption
  - Lower System Cost
  - Longer Battery Life
  - Lighter/Smaller Systems
- Low Power
  - Longer Battery Life
  - Lighter/Smaller Systems

"The Right Choice"