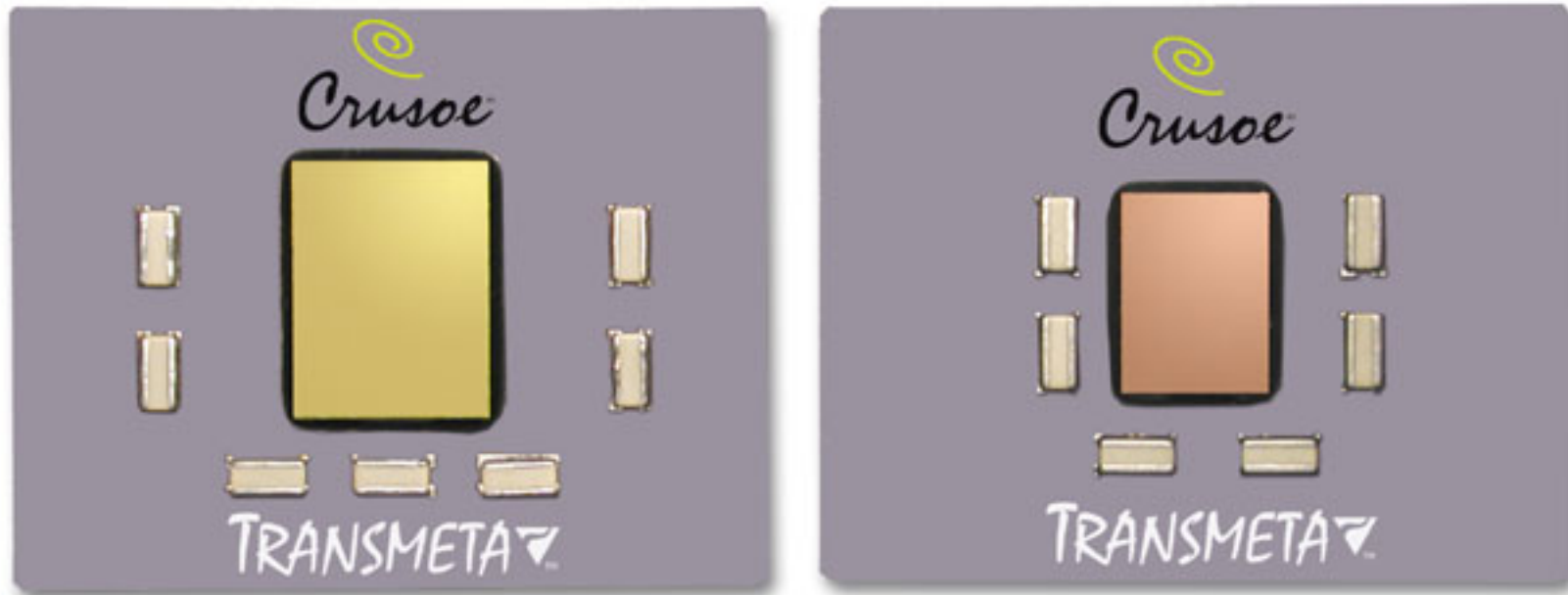


The Transmeta Crusoe - A Smart Microprocessor for Mobile Internet Computing

Sascha Opletal



Outline

- Introduction: There is a need for a new processor design
- The Hardware: The Processor Core etc.
- The Software: The Code Morphing Technology
- Results: Benchmarks / Battery-life etc.
- Summary

The market situation for mobile processors

- High demand for mobile computing devices
 - Small systems with high performance
 - Long battery life to be useful
- Mobile functionality limited:
 - Unchanged desktop designs (e.g. mobile Pentium/Athlon)
 - Specialized solutions (e.g. ARM/Dragonball)

The Plan

A processor providing:

- Low power consumption
- Flexible Internet connection at low cost
- Reasonable speed
- Small system sizes

The Realization

Design Ideas:

- Large parts of the processor are implemented in software to reduce power consumption
- x86 compatibility for cost efficient Internet connection
- High performing processor core/software combination
- Integration of normally separate components onto the processor die

High performing CPU architectures

- Superscalar CISC processors
 - Complex commands / parallel execution in the CPU
- RISC processors
 - Simple commands / fast execution
- VLIW processors
 - Combination of the above

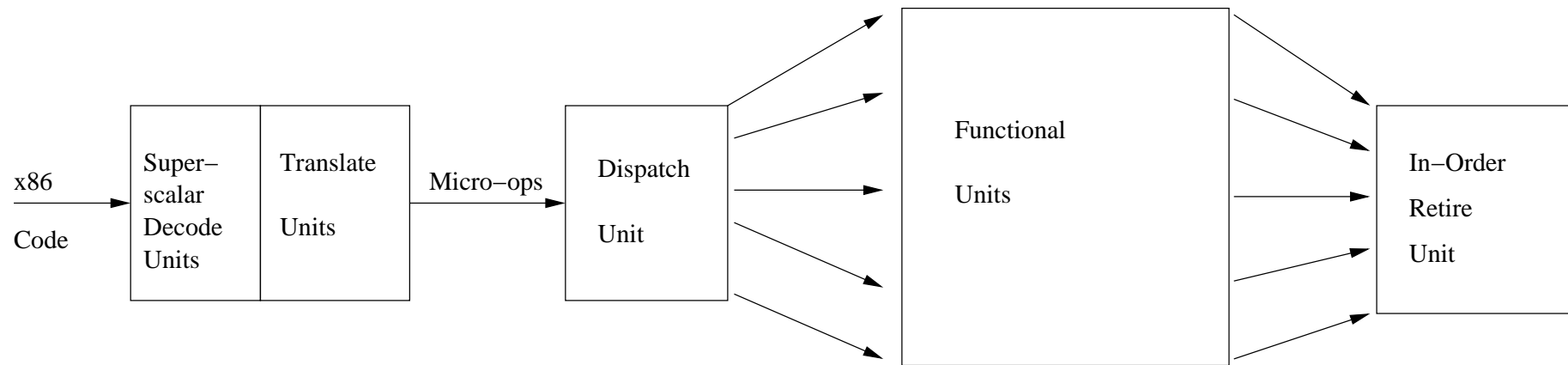
An x86 code Example

x86 code chunk:

- A. `addl %eax, (%esp) // load data from stack, add to %eax`
- B. `addl %ebx, (%esp) // load data from stack, add to %ebx`
- C. `movl %esi, (%ebp) // load %esi from memory`
- D. `subl %ecx, 5 // subtract 5 from %ecx register`

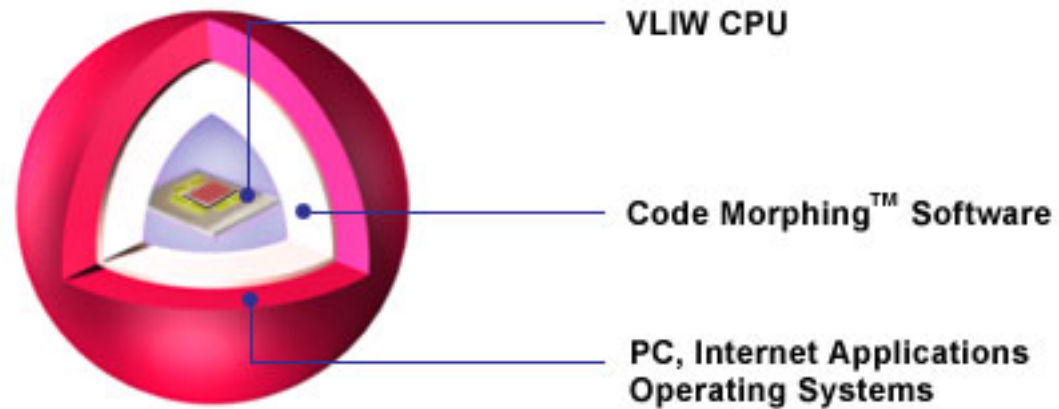
Conventional out-of-order execution

Execution of RISC-like micro-ops in conventional x86 superscalar processors such as the Pentium II and Pentium III



Structure of the VLIW system

x86 compatibility will be achieved by code translation



Outline

- Introduction: There is a need for a new processor design
- **The Hardware: The Processor Core etc.**
- The Software: The Code Morphing Technology
- Results: Benchmarks / Battery-life etc.
- Summary

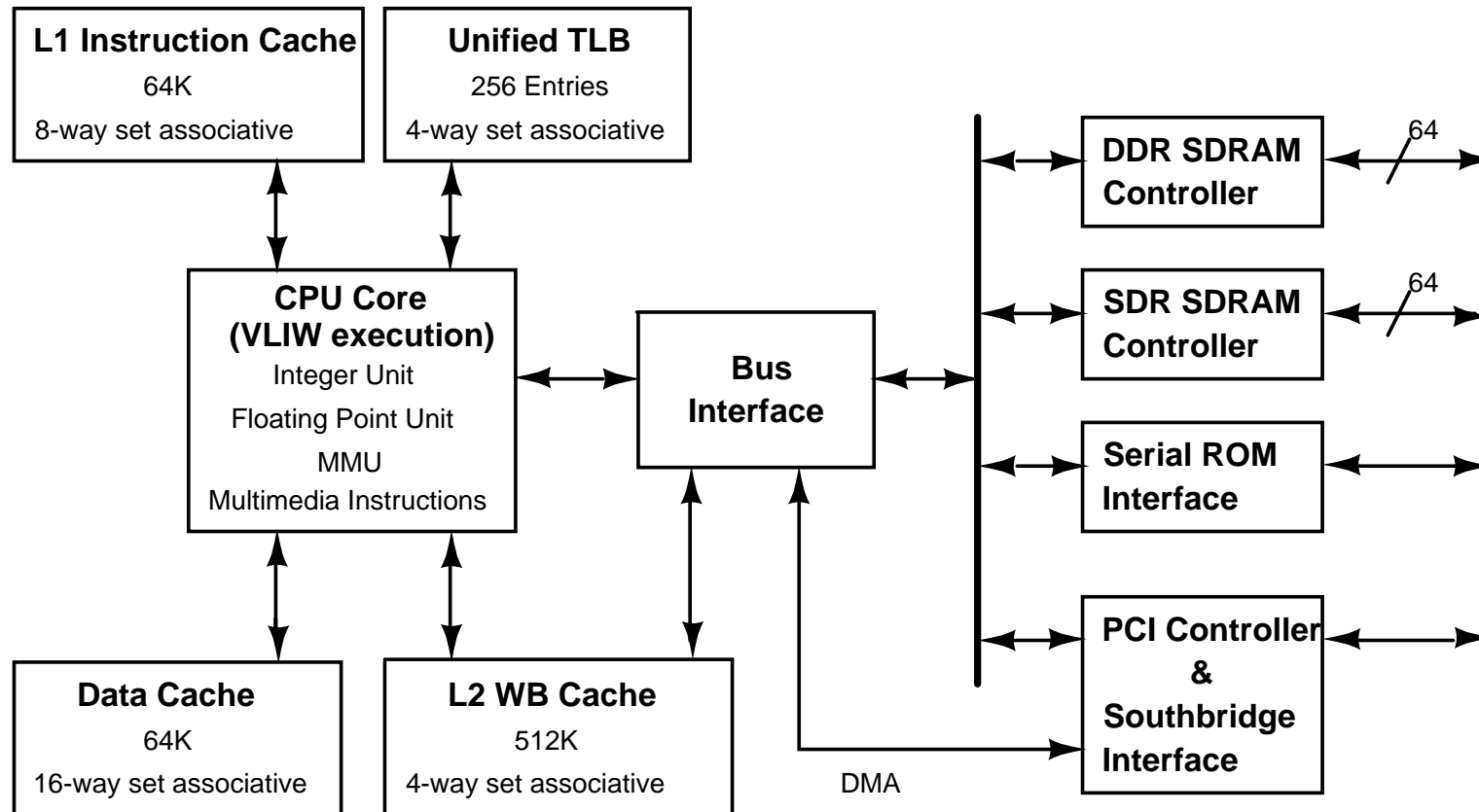
Power reduction on the hardware level

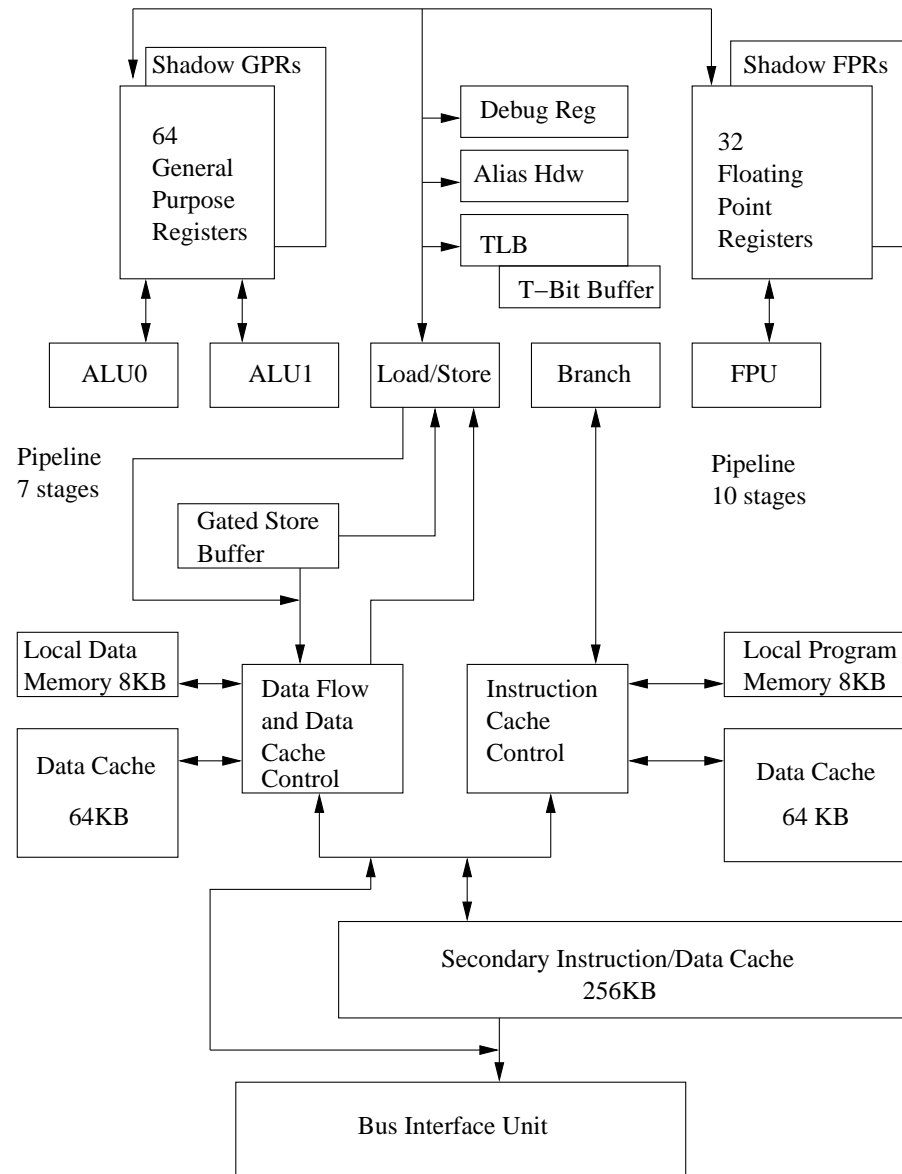
Power consumption of a system can be reduced by:

- Replacing processor logic with software \Rightarrow fewer transistors
- Using new manufacturing processes \Rightarrow smaller transistors
- Integrating external components onto the die \Rightarrow reducing resistance

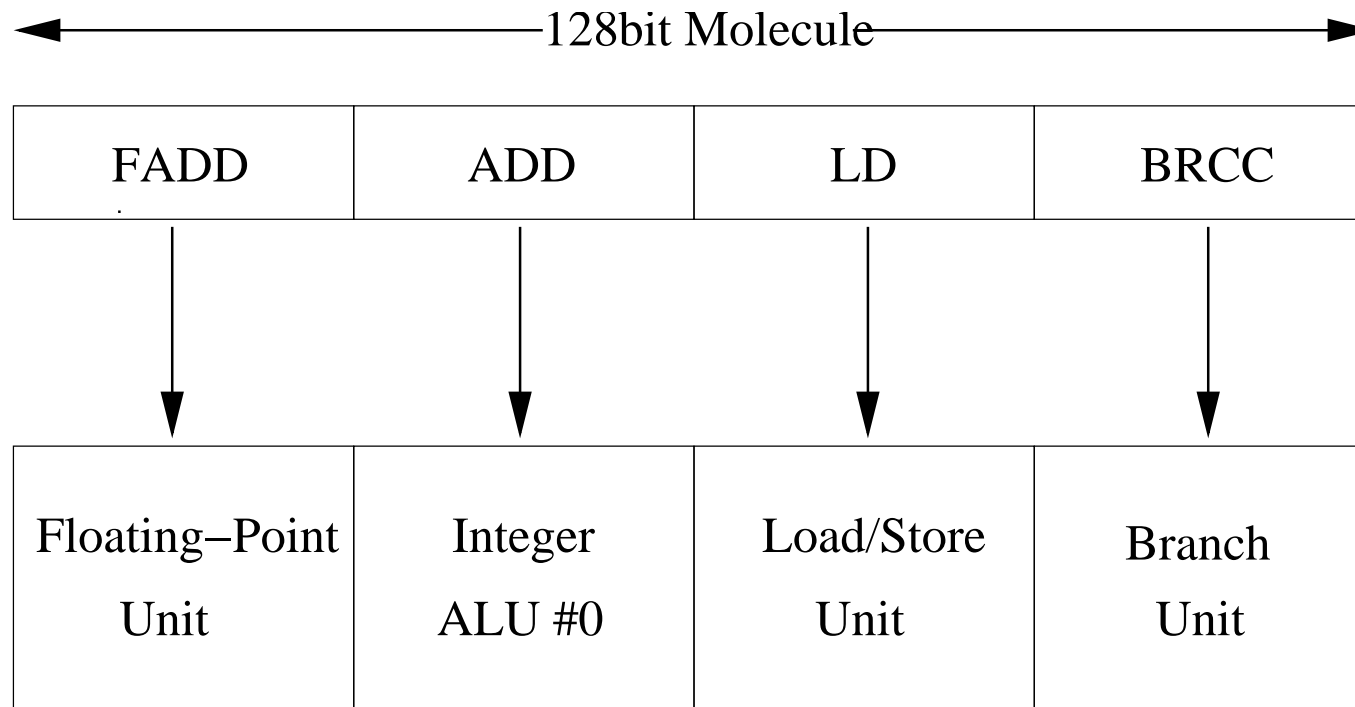
$$Power = \frac{Total\ Capacitance * Frequency * Voltage^2}{2}$$

Processor structure (1)





The Instruction Word (VLIW)



Die sizes of selected mobile processors

The streamlined VLIW processor core makes the needed die-size smaller

| | Mobile PIII | TM5400 | ARM920T Macrocell |
|-------------------------|-----------------------------|-----------------------------|-----------------------------|
| Process | <i>.18μm</i> | <i>.18μm</i> | <i>.18μm</i> |
| On-chip L1 Cache | 32KB | 128KB | 16KB |
| On-chip L2 Cache | 256KB | 256KB | - |
| Die Size | 106mm ² | 73mm ² | 11.8mm ² |
| Power cons. | 0.3-20 W | 1-3 W | 160mW |

The Model Range

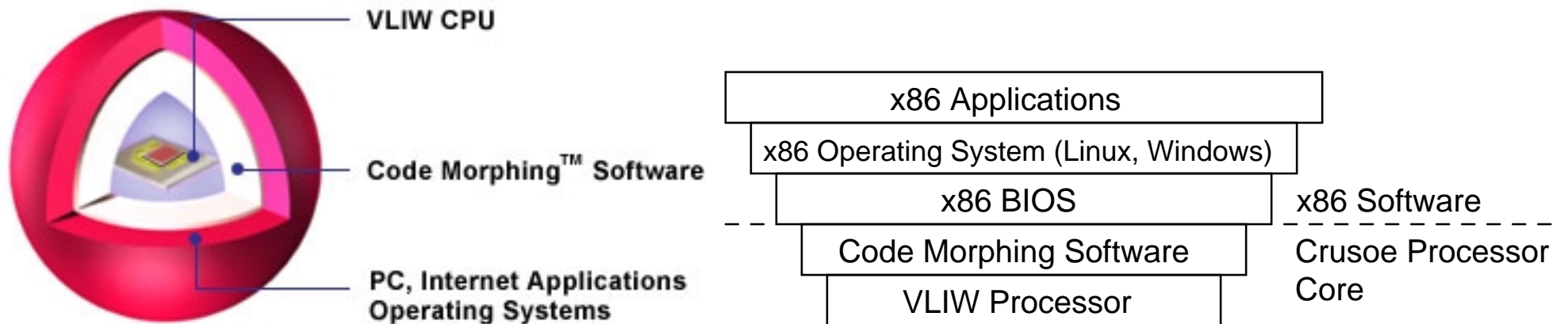
| | | | | |
|-------------------|--------------|--------------|--------------|--------------|
| Processor | TM5400 | TM5500 | TM5600 | TM5800 |
| Clock Mhz | 500-700 | 667-800 | 500-700 | 667-800 |
| L1-Cache | 128KB | 128KB | 128KB | 128KB |
| L2-Cache | 256KB | 256KB | 512KB | 512KB |
| Core voltage | 1,2-1,6V | 0,9-1,3V | 1,2-1,6V | 0,9-1,3V |
| Power consumption | 1-3 W | 0.4-1.0 W | 1-2 W | 0,4-1.0 W |
| Processtechnology | 0,18 μm | 0,13 μm | 0,18 μm | 0,13 μm |
| Case | 474 BGA | 474 BGA | 474 BGA | 474 BGA |
| Production | now | July 2001 | now | July 2001 |

Outline

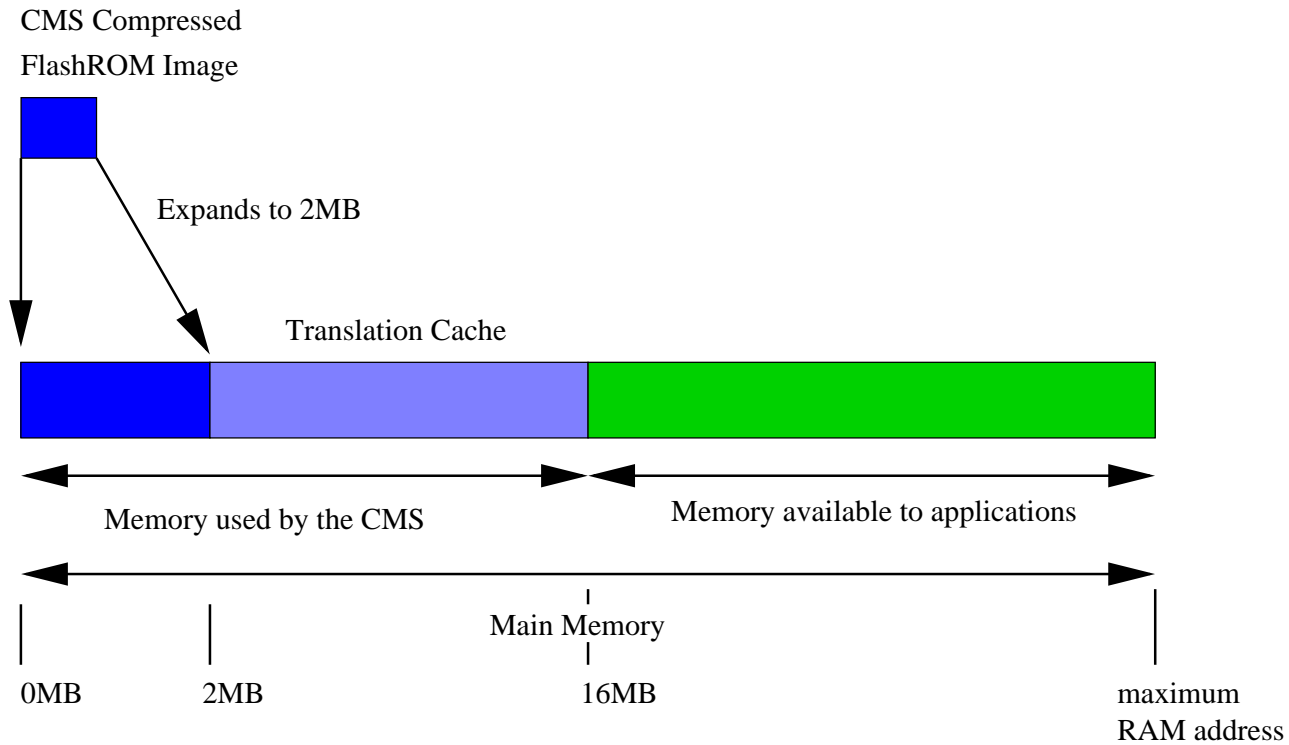
- Introduction: There is a need for a new processor design
- The Hardware: The Processor Core etc.
- **The Software: The Code Morphing Technology**
- Results: Benchmarks / Battery-life etc.
- Summary

Executing x86 Applications

The combination of both hardware and software leads to a fully x86 compatible execution environment



Memory layout of a running system



Code Morphing Example (1)

x86 code chunk:

```
A. addl %eax, (%esp) // load data from stack, add to %eax
B. addl %ebx, (%esp) // load data from stack, add to %ebx
C. movl %esi, (%ebp) // load %esi from memory
D. subl %ecx, 5      // subtract 5 from %ecx register
```

Translation for the target architecture:

```
ld %r30, [%esp]      // load from stack into temporary register
add.c %eax, %eax, %r30 // add to %eax, set condition codes
ld %r31, [%esp]      // load from stack into temporary register
add.c %ebx, %ebx, %r31 // add to %ebx, set condition codes
ld %esi, [%ebp]      // load %esi from memory
sub.c %ecx, %ecx, 5   // subtract 5 from %ecx register
```

Code Morphing Example (2)

Optimization:

```
ld %r30, [%esp]           // load from stack only once
add %eax,%eax,%r30
add %ebx,%ebx,%r30        // reuse data loaded earlier
ld %esi, [%ebp]
sub.c %ecx,%ecx,5         // only this last condition code needed
```

Atom grouping and scheduling:

1. `ld %r30, [%esp]; sub.c %ecx,%ecx,5`
2. `ld %esi, [%ebp]; add %eax,%eax,%r30; add %ebx,%ebx,%r30`

Comparison of the two execution methods

| Traditional x86 Processors | Crusoe Processor with Code Morphing Software |
|---|--|
| Translates single instructions one after one | Translates an entire group of x86 instructions at once |
| Translates each x86 instruction everytime it is encountered | Translates instructions once, saving the translation in a cache for re-use |
| Static compiled programs | Just-in-Time compilation Optimization possible |

LongRunTM Powermanagement Technology

| Power Management Technologies | Features |
|--------------------------------------|--|
| Legacy Power Management technology | Old and primitive Processor on or off Sleep mode |
| Intel's SpeedStep technology | Powersource based approach Two distinct frequencies |
| Transmeta's LongRun technology | Processor activity based Fine adjustment of frequency and voltage |

$$Power = \frac{Total\ Capacitance * Frequency * Voltage^2}{2}$$

Outline

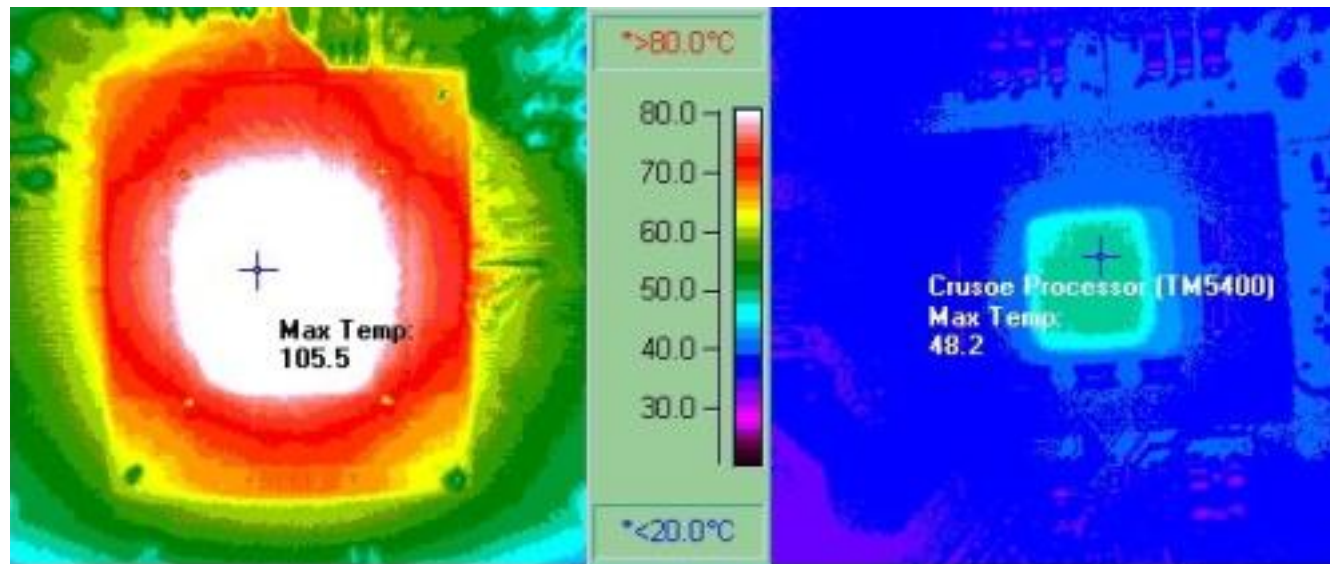
- Introduction: There is a need for a new processor design
- The Hardware: The Processor Core etc.
- The Software: The Code Morphing Technology
- **Results: Benchmarks / Battery-life etc.**
- Summary

Experimental Results of the Design Ideas

- Full x86 compatible execution environment
- Long battery life
- Low operating temperature
- Possibility to build smaller systems
- Enough computing power for most tasks

Operating temperatures

Operating temperatures of a Pentium III compared to a Crusoe TM5400 while running a software DVD player



Battery runtime on different applications

| Notebook | Processor | Media Player | Winamp |
|---------------------------|--------------------------|---------------------|--------------------|
| IBM X-Series ThinkPad X20 | Intel Pentium III 500MHz | 2:58 hours | 6:24 hours |
| IBM X-Series ThinkPad X20 | Intel Pentium III 300MHz | 3:50 hours | 6:46 hours |
| NEC Versa DayLite | Transmeta TM5600-600MHz | 7:14 hours | 12:04 hours |

| Notebook | Processor | WinDVD - The Matrix |
|----------------------------|--------------------------|----------------------------|
| IBM I-Series ThinkPad 1124 | Intel Pentium III 300MHz | N/A |
| IBM X-Series ThinkPad X20 | Intel Pentium III 500MHz | 1:59 hours |
| NEC Versa DayLite | Transmeta TM5600-600MHz | 4:04 hours |

Benchmarks

| Notebook | Processor | CPUMark99 Score |
|-------------------------------|----------------------------|------------------------|
| Dell Latitude L400 | Pentium III 500MHz | 47.8 |
| IBM X-Series ThinkPad X20 | Pentium III 500MHz | 47.5 |
| IBM I-Series ThinkPad 1124 | Pentium III 300MHz | 29.8 |
| NEC Versa UltraLite | Transmeta TM5600-600MHz | 39.3 |
| NEC Versa DayLite | Transmeta TM5600-600MHz | 39.5 |

Summary

The Crusoe is well suited for mobile systems:

- Low development cost by reusing x86 software
- Low power consumption
- Low heat and noise production
- Ability to build smaller systems
- Highly reconfigurable
 - FlashROM
 - CMS

