Embedded Computing

Krste Asanovic
Laboratory for Computer Science
Massachusetts Institute of Technology
How many microprocessors do you own?

- Average individual in developed country owns around 100 microprocessors
  - *Almost all are embedded*
- Maybe 10,000 processors/person by 2012! (according to Moore’s Law)
Future Computing Infrastructure

- MegaWatt Server Farms
- Wireless Networks
- Wireless Sensor Networks
- PDAs, Cameras, Cellphones, Laptops, GPS, Set-tops, 0.1-10 Watt Clients
- Base Stations
- The Internet
- Routers
- MegaWatt Server Farms
What is an Embedded Computer?

- A computer not used to run general-purpose programs, but instead used as a component of a system. Usually, user cannot change the computer program (except for minor upgrades).

- Example applications:
  - Toasters
  - Cellphone
  - Digital camera (some have several processors)
  - Games machines
  - Set-top boxes (DVD players, personal video recorders, ...)
  - Televisions
  - Dishwashers
  - Car (some have dozens of processors)
  - Router
  - Cellphone basestation
  - .... many more
Early Embedded Computing Examples

• MIT Whirlwind, 1946-51
  – developed for real-time flight simulator

• Intel 4004, 1971
  – developed for Busicom 141-PF printing calculator
Important Parameters for Embedded Computers

- **Real-time performance**
  - *hard real-time*: if deadline missed system has failed (car brakes!)
  - *soft real-time*: missing deadline degrades performance (skipping frames on DVD playback)

- **Real-world I/O performance**
  - sensor and actuators require continuous I/O (can’t batch process)

- **Cost**
  - includes cost of supporting structures, particularly memory
  - static code size very important (cost of ROM/RAM)
  - often ship millions of copies (worth engineer time to optimize cost down)

- **Power**
  - expensive package and cooling affects cost, system size, weight
What is Performance?

- **Latency (or response time or execution time)**
  - time to complete one task

- **Bandwidth (or throughput)**
  - tasks completed per unit time
Performance Measurement

Execution Rate

Average Rate:  \( A > B > C \)
Worst-case Rate:  \( A < B < C \)

Which is best for desktop performance? _______
Which is best for hard real-time task? _______
Future Computing Infrastructure

- MegaWatt Server Farms
- Wireless Networks
- Processors defined by Watts not MIPS!
- PDAs, Cameras, Cellphones, Laptops, GPS, Set-tops, 0.1-10 Watt Clients
Physics Review

- Energy measured in Joules
- Power is rate of energy consumption measured in Watts (Joules/second)
- Instantaneous power is Vdd * Idd

Battery Capacity Measured in Joules
- 720 Joules/gram for Lithium-Ion batteries
- 1 instruction on Intel XScale takes ~1nJ
Power versus Energy

- System A has higher peak power, but lower total energy
- System B has lower peak power, but higher total energy
Impacts on Computer System

• Energy consumed per task determines battery life
  – Second order effect is that higher current draws decrease effective battery energy capacity (higher power also lowers battery life)

• Current draw causes IR drops in power supply voltage
  – Requires more power/ground pins to reduce resistance R
  – Requires thick&wide on-chip metal wires or dedicated metal layers

• Switching current \((dl/dt)\) causes inductive power supply voltage bounce \(\propto Ldl/dt\)
  – Requires more pins/shorter pins to reduce inductance L
  – Requires on-chip/on-package decoupling capacitance to help bypass pins during switching transients

• Power dissipated as heat, higher temps reduce speed and reliability
  – Requires more expensive packaging and cooling systems
  – Fan noise
  – Laptop temperature
Power Dissipation in CMOS

Primary Components:

- **Capacitor Charging (85-90% of active power)**
  - Energy is $\frac{1}{2} CV^2$ per transition

- **Short-Circuit Current (10-15% of active power)**
  - When both p and n transistors turn on during signal transition

- **Subthreshold Leakage (dominates when inactive)**
  - Transistors don’t turn off completely
  - Becoming more significant part of active power with scaling

- **Diode Leakage (negligible)**
  - Parasitic source and drain diodes leak to substrate
Reducing Switching Power

Power $\propto$ activity $\times \frac{1}{2} CV^2 \times$ frequency

- Reduce activity
- Reduce switched capacitance C
- Reduce supply voltage V
- Reduce frequency
Reducing Activity

Clock Gating
- don’t clock flip-flop if not needed
- avoids transitioning downstream logic
- Pentium-4 has hundreds of gated clocks

Bus Encodings
- choose encodings that minimize transitions on average (e.g., Gray code for address bus)
- compression schemes (move fewer bits)

Remove Glitches
- balance logic paths to avoid glitches during settling
- use monotonic logic (domino)
Reducing Switched Capacitance

Reduce switched capacitance $C$
- Different logic styles (logic, pass transistor, dynamic)
- Careful transistor sizing
- Tighter layout
- Segmented structures

![Diagram showing bus and switches](image)

**Shared bus driven by A or B when sending values to C**

**Insert switch to isolate bus segment when B sending to C**
Reducing Supply Voltage

Quadratic savings in energy per transition – *BIG* effect

- Circuit speed is reduced
- Must lower clock frequency to maintain correctness
Reducing Frequency

• Doesn’t save energy, just reduces rate at which it is consumed
  – Some saving in battery life from reduction in rate of discharge
Voltage Scaling for Reduced Energy

- Reducing supply voltage by 0.5 improves energy per transition by 0.25
- Performance is reduced – need to use slower clock
- Can regain performance with parallel architecture

- Alternatively, can trade surplus performance for lower energy by reducing supply voltage until “just enough” performance

*Dynamic Voltage Scaling*
Parallel Architectures Reduce Energy at Constant Throughput

• 8-bit adder/comparator
  – 40MHz at 5V, area = 530 k\(\mu\)\(^2\)
  – Base power Pref

• Two parallel interleaved adder/compare units
  – 20MHz at 2.9V, area = 1,800 k\(\mu\)\(^2\) (3.4x)
  – Power = 0.36 Pref

• One pipelined adder/compare unit
  – 40MHz at 2.9V, area = 690 k\(\mu\)\(^2\) (1.3x)
  – Power = 0.39 Pref

• Pipelined and parallel
  – 20MHz at 2.0V, area = 1,961 k\(\mu\)\(^2\) (3.7x)
  – Power = 0.2 Pref

Chandrakasan et. al. “Low-Power CMOS Digital Design”,
IEEE JSSC 27(4), April 1992
“Just Enough” Performance

- Save energy by reducing frequency and voltage to minimum necessary (usually done in O.S.)

- Run fast then stop

- Run slower and just meet deadline

- Time

- Frequency

- $t=0$

- $t=\text{deadline}$
# Voltage Scaling on Transmeta Crusoe TM5400

<table>
<thead>
<tr>
<th>Frequency (MHz)</th>
<th>Relative Performance (%)</th>
<th>Voltage (V)</th>
<th>Relative Energy (%)</th>
<th>Relative Power (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>700</td>
<td>100.0</td>
<td>1.65</td>
<td>100.0</td>
<td>100.0</td>
</tr>
<tr>
<td>600</td>
<td>85.7</td>
<td>1.60</td>
<td>94.0</td>
<td>80.6</td>
</tr>
<tr>
<td>500</td>
<td>71.4</td>
<td>1.50</td>
<td>82.6</td>
<td>59.0</td>
</tr>
<tr>
<td>400</td>
<td>57.1</td>
<td>1.40</td>
<td>72.0</td>
<td>41.4</td>
</tr>
<tr>
<td>300</td>
<td>42.9</td>
<td>1.25</td>
<td>57.4</td>
<td>24.6</td>
</tr>
<tr>
<td>200</td>
<td>28.6</td>
<td>1.10</td>
<td>44.4</td>
<td>12.7</td>
</tr>
</tbody>
</table>
Types of Embedded Computer

• General Purpose Processors
  - often too expensive, too hot, too unpredictable, and require too much support logic for embedded applications

• Microcontroller
  - emphasizes bit-level operations and control-flow intensive operations (a programmable state machine)
  - usually includes on-chip memories and I/O devices

• DSP (Digital Signal Processor)
  - organized around a multiply-accumulate engine for digital signal processing applications

• FPGA (Field Programmable Gate Array)
  - reconfigurable logic can replace processors/DSPs for some applications
New Forms of Domain-Specific Processor

- **Network processor**
  - arrays of 8-16 simple multithreaded processor cores on a single chip used to process Internet packets
  - used in high-end routers

- **Media processor**
  - conventional RISC or VLIW engine extended with media processing instructions (SIMD or Vector)
  - used in set-top boxes, DVD players, digital cameras
Single 32-bit DSP instruction:
\[
\text{AccA} \leftarrow (\text{AR1}++) \times (\text{AR2}++)
\]

Equivalent to one multiply, three adds, and two loads in RISC ISA!
Network Processors

RISC Control Processor

MicroEngine 0

MicroEngine 1

MicroEngine 15

Microcode RAM

Register File

ALU

Scratchpad Data RAM

Buffer RAM

PC0

PC1

PC7

Eight threads per microengine

16 Multithreaded microengines

Network 10Gb/s

DRAM0

DRAM1

DRAM2

SRAM0

SRAM1

SRAM2

SRAM3

Register File

ALU

Scratchpad Data RAM

Buffer RAM

PC0

PC1

PC7

Eight threads per microengine

16 Multithreaded microengines

Network Processor

MicroEngine 1

MicroEngine 0

Microcode RAM

Register File

ALU

Scratchpad Data RAM

Buffer RAM

PC0

PC1

PC7

Eight threads per microengine

16 Multithreaded microengines

Network Processor

MicroEngine 1

MicroEngine 0

Microcode RAM

Register File

ALU

Scratchpad Data RAM

Buffer RAM

PC0

PC1

PC7

Eight threads per microengine

16 Multithreaded microengines

Network Processor

MicroEngine 1

MicroEngine 0

Microcode RAM

Register File

ALU

Scratchpad Data RAM

Buffer RAM

PC0

PC1

PC7

Eight threads per microengine

16 Multithreaded microengines

Network Processor

MicroEngine 1

MicroEngine 0

Microcode RAM

Register File

ALU

Scratchpad Data RAM

Buffer RAM

PC0

PC1

PC7

Eight threads per microengine

16 Multithreaded microengines

Network Processor

MicroEngine 1

MicroEngine 0

Microcode RAM

Register File

ALU

Scratchpad Data RAM

Buffer RAM

PC0

PC1

PC7

Eight threads per microengine

16 Multithreaded microengines

Network Processor

MicroEngine 1

MicroEngine 0

Microcode RAM

Register File

ALU

Scratchpad Data RAM

Buffer RAM

PC0

PC1

PC7

Eight threads per microengine

16 Multithreaded microengines

Network Processor

MicroEngine 1

MicroEngine 0

Microcode RAM

Register File

ALU

Scratchpad Data RAM

Buffer RAM

PC0

PC1

PC7

Eight threads per microengine

16 Multithreaded microengines

Network Processor

MicroEngine 1

MicroEngine 0

Microcode RAM

Register File

ALU

Scratchpad Data RAM

Buffer RAM

PC0

PC1

PC7

Eight threads per microengine

16 Multithreaded microengines

Network Processor

MicroEngine 1

MicroEngine 0

Microcode RAM

Register File

ALU

Scratchpad Data RAM

Buffer RAM

PC0

PC1

PC7

Eight threads per microengine

16 Multithreaded microengines

Network Processor

MicroEngine 1

MicroEngine 0

Microcode RAM

Register File

ALU

Scratchpad Data RAM

Buffer RAM

PC0

PC1

PC7

Eight threads per microengine

16 Multithreaded microengines

Network Processor

MicroEngine 1

MicroEngine 0

Microcode RAM

Register File

ALU

Scratchpad Data RAM

Buffer RAM

PC0

PC1

PC7

Eight threads per microengine

16 Multithreaded microengines

Network Processor

MicroEngine 1

MicroEngine 0

Microcode RAM

Register File

ALU

Scratchpad Data RAM

Buffer RAM

PC0

PC1

PC7

Eight threads per microengine

16 Multithreaded microengines

Network Processor

MicroEngine 1

MicroEngine 0

Microcode RAM

Register File

ALU

Scratchpad Data RAM

Buffer RAM

PC0

PC1

PC7

Eight threads per microengine

16 Multithreaded microengines

Network Processor

MicroEngine 1

MicroEngine 0

Microcode RAM

Register File

ALU

Scratchpad Data RAM

Buffer RAM

PC0

PC1

PC7

Eight threads per microengine

16 Multithreaded microengines

Network Processor

MicroEngine 1

MicroEngine 0

Microcode RAM

Register File

ALU

Scratchpad Data RAM

Buffer RAM

PC0

PC1

PC7

Eight threads per microengine

16 Multithreaded microengines

Network Processor

MicroEngine 1

MicroEngine 0

Microcode RAM

Register File

ALU

Scratchpad Data RAM

Buffer RAM

PC0

PC1

PC7

Eight threads per microengine

16 Multithreaded microengines

Network Processor
Programming Embedded Computers

- Microcontrollers, DSPs, network processors, media processors usually have complex, non-orthogonal instruction sets with specialized instructions and special memory structures
  - poor compiled code quality (% peak with compiled code)
  - high static code efficiency
  - high MIPS/$ and MIPS/W
  - usually assembly-coded in critical loops

- Worth one engineer year in code development to save $1 on system that will ship 1,000,000 units

- Assembly coding easier than ASIC chip design

- But room for improvement...