Embedded System Hardware - Processing -

Jian-Jia Chen
(Slides are based on Peter Marwedel)
Informatik 12
TU Dortmund
Germany

2017年 11 月 15 日

These slides use Microsoft clip arts. Microsoft copyright restrictions apply.
Embedded System Hardware

Embedded system hardware is frequently used in a loop ("hardware in a loop"): 

- Sensors
- A/D converter
- Sample-and-hold
- Information processing
- Display
- D/A converter
- Actuators

Cyber-physical systems
Not So Seriously

• Dr. Michio Kaku's states in one of his recent books: "Today, your cell phone has more computer power than all of NASA back in 1969, when it placed two astronauts on the moon."

• Seems hard to believe, we know, but it is actually true – a hand-held apparatus on which we fling birds at pigs has greater computational capabilities than the arsenal of machines used for guiding crafts through outer space some 45 years ago.
Efficiency: slide from lecture 1 applied to processing

- CPS & ES must be **efficient**
  - Code-size efficient (especially for systems on a chip)
  - Run-time efficient
  - Weight efficient
  - Cost efficient
  - Energy/power efficient
Key requirement #1: Code-size efficiency

- Overview: http://www-perso.iro.umontreal.ca/~latendre/codeCompression/codeCompression/node1.html

- Compression techniques: key idea
Code-size efficiency

- Compression techniques (continued):
  - 2nd instruction set, e.g. ARM Thumb instruction set:

  16-bit Thumb instr.
  ADD Rd #constant

<table>
<thead>
<tr>
<th></th>
<th></th>
<th>Rd</th>
<th>Constant</th>
</tr>
</thead>
<tbody>
<tr>
<td>001</td>
<td>10</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

  major opcode

  minor opcode

  source = destination

  zero extended

  1110 001 01001 0 Rd 0 Rd 0000 Constant

  - Reduction to 65-70 % of original code size
  - 130% of ARM performance with 8/16 bit memory
  - 85% of ARM performance with 32-bit memory

  Same approach for LSI TinyRisc, …
  Requires support by compiler, assembler etc.

[ARM, R. Gupta]
Dictionary approach, two level control store (indirect addressing of instructions)

“Dictionary-based coding schemes cover a wide range of various coders and compressors.

Their common feature is that the methods use some kind of a dictionary that contains parts of the input sequence which frequently appear.

The encoded sequence in turn contains references to the dictionary elements rather than containing these over and over.”

Key idea (for $d$ bit instructions)

Uncompressed storage of a $d$-bit-wide instructions requires $a \times d$ bits.

In compressed code, each instruction pattern is stored only once.

Hopefully, $axb + cxd < axd$.

Called nanoprogramming in the Motorola 68000.
Key requirement #2: Run-time efficiency
- Domain-oriented architectures -

Example: Filtering in Digital signal processing (DSP)

\[ x_S = \sum_{k=0}^{n-1} w_{S-k} \ast a_k \]

Signal at \( t=t_s \) (sampling points)
Key requirement #3: energy-efficient and power efficient

<table>
<thead>
<tr>
<th>Execution platform</th>
<th>Relevant during use?</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Plugged</td>
</tr>
<tr>
<td>E.g.</td>
<td>Factory</td>
</tr>
<tr>
<td>Global warming</td>
<td>✓</td>
</tr>
<tr>
<td>Cost of energy</td>
<td>✓</td>
</tr>
<tr>
<td>Increasing performance</td>
<td>✓</td>
</tr>
<tr>
<td>Problems with cooling, avoiding hot spots</td>
<td>✓</td>
</tr>
<tr>
<td>Avoiding high currents &amp; metal migration</td>
<td>✓</td>
</tr>
<tr>
<td>Reliability</td>
<td>✓</td>
</tr>
<tr>
<td>Energy a very scarce resource</td>
<td>☐</td>
</tr>
</tbody>
</table>

- Global warming and cost of energy are relevant during use (plug/unplug).
- Problems with cooling and reliability are relevant during both plug and unplug.
- Avoiding high currents and metal migration are relevant during plug.
- Energy scarcity is relevant during unplug.

Power efficiency is crucial for long-term sustainability.
Should we care about energy consumption or about power consumption?

\[ E = \int P(t) \, dt \]

Both are closely related, but still different.
Should we care about energy consumption or about power consumption (2)?

- Minimizing **power consumption** important for
  - design of the power supply & regulators
  - dimensioning of interconnect, short term cooling

- Minimizing **energy consumption** important due to
  - restricted availability of energy (mobile systems)
  - cooling: high costs, limited space
  - thermal effects
  - dependability, long lifetimes

☞ **In general, we need to care about both**
PCs: Problem: Power density increasing

Surpassed hot-plate power density in 0.5μ
Not too long to reach nuclear reactor

© Intel
M. Pollack,
Micro-32
PCs: Just adding transistors would have resulted in this:

**Reuters: December 9, 2004:** Men should keep their laptops off their laps because they could damage fertility, an expert said on Thursday. Laptops, which reach **high internal operating temperatures**, can heat up the scrotum which could affect the quality and quantity of men’s sperm. “The increase in scrotal temperature is significant enough to cause changes in sperm parameters,” said Dr Yefim Sheynkin, an associate professor of urology at the State University of New York at Stony Brook.
How do We Now Cook
PCs: Surpassed hot (kitchen) plate ...? Why not use it?

Strictly speaking, energy is not “consumed”, but converted from electrical energy into heat energy.

http://www.phys.ncku.edu.tw/~htsu/humor/fry_egg.html
Cooling Matters

• Thermoelectric cooling
• Liquid cooling
• Refrigeration cooling
• etc.
Keep it simple, stupid (KISS)

Mobile phones: Increasing performance requirements

C.H. van Berkel: Multi-Core for Mobile Phones, DATE, 2009;

Many more instances of the power/energy problem
Mobile phones: Where does the power go?

Source: Siemens

[O. Vargas: Minimum power consumption in mobile-phone memory subsystems; Pennwell Portable Design - September 2005;]
Prerequisite: CMOS Circuits
Static and dynamic power consumption

- Dynamic power consumption: Power consumption caused by charging capacitors when logic levels are switched.

\[ P = \alpha C_L V_{dd}^2 s \]

\( \alpha \): switching activity
\( C_L \): load capacitance
\( V_{dd} \): supply voltage
\( s \): clock frequency

\[ \text{Decreasing } V_{dd} \text{ reduces } P \text{ quadratically} \]

- Static power consumption (caused by leakage current): power consumed in the absence of clock signals
- Leakage becoming more important due to smaller devices
How to make systems energy efficient: Fundamentals of dynamic voltage scaling (DVS)

Power consumption of CMOS circuits (ignoring leakage):

\[ P = \alpha \ C_L \ V_{dd}^2 \ s \] with

\( \alpha \) : switching activity
\( C_L \) : load capacitance
\( V_{dd} \) : supply voltage
\( s \) : clock frequency

\[ Delay \ for \ CMOS \ circuits: \]

\( \tau = k \ C_L \ \frac{V_{dd}}{(V_{dd} - V_t)^2} \]

\( V_t \) : threshold voltage
\( (V_t \ < \ than \ V_{dd}) \)

\( \Downarrow \) Decreasing \( V_{dd} \) reduces \( P \) quadratically, while the run-time of algorithms is only linearly increased.
How to make systems energy efficient: 
Fundamentals of dynamic voltage/frequency scaling (DVFS)

Power consumption of CMOS circuits (ignoring leakage):

\[ P = \alpha C_L V_{dd}^2 s \]

- \( \alpha \): switching activity
- \( C_L \): load capacitance
- \( V_{dd} \): supply voltage
- \( s \): clock frequency

\( \tau = k C_L \frac{V_{dd}}{(V_{dd} - V_t)^2} \)

- \( \tau \): delay for CMOS circuits
- \( k \): constant
- \( V_t \): threshold voltage

\( V_t < \) than \( V_{dd} \)

\( \exists \) Decreasing \( V_{dd} \) and frequency together reduces \( P \) cubically, while the run-time of algorithms is only linearly increased
Voltage/Frequency scaling: Example

Maximum Clock Frequency

Energy Consumption

[Courtesy, Yasuura, 2000]
Abstract Power Model

CMOS-core Power Model

\[ P(s) = P_{\text{dynamic}}(s) + P_{\text{static}} \]

Considering that:

\[ P_{\text{dynamic}}(s) = C_{\text{eff}} V_{dd}^2 s \]

\[ s \propto \frac{(V_{dd} - V_t)^2}{V_{dd}} \]

We can approximate to:

\[ P(s) = \alpha s^\gamma + \beta \]

**Figure:** \( \alpha = 1.76 \frac{\text{Watts}}{\text{GHz}^3} \), \( \gamma = 3 \) and \( \beta = 0.5 \text{ Watts} \)
Abstract Energy Model

Energy Consumption

\[ E(s) = (\alpha s^\gamma + \beta) \frac{\Delta c}{s} \]

Critical Frequency:

\[ s_{\text{crit}} = \sqrt[\gamma - 1]{\frac{\beta}{\alpha}} \]

Figure: \( \alpha = 1.76 \text{ Watts/GHz}^3 \), \( \gamma = 3 \), \( \beta = 0.5 \text{ Watts} \) and \( \Delta c = 10^9 \) cycles
Dynamic power management (DPM)

Example: STRONGARM SA1100

**RUN**: operational

**IDLE**: a SW routine may stop the CPU when not in use, while monitoring interrupts

**SLEEP**: Shutdown of on-chip activity

- **RUN**: 400mW
- **IDLE**: 50mW
- **SLEEP**: 160µW

- **10µs** from **RUN** to **IDLE**
- **10µs** from **IDLE** to **SLEEP**
- **90µs** from **SLEEP** to **RUN**
- **160µs** from **SLEEP** to power fault signal
Low voltage, parallel operation more efficient than high voltage, sequential operation

Basic equations

Power:
\[ P \sim V_{DD}^2 s, \]

Maximum clock frequency:
\[ s \sim V_{DD}, \]

Energy to run a program:
\[ E = P \times t, \text{ with: } t = \text{runtime} \]
\[ t \sim 1/s \]

Time to run a program:

Changes due to parallel processing, with \( M \) operations per clock:

Clock frequency reduced to:
\[ s' = s / M, \]

Voltage can be reduced to:
\[ V_{DD}' = V_{DD} / M, \]

Power for parallel processing:
\[ P' = P / M^3 \text{ per operation,} \]
\[ P' = M \times P' = P / M^2, \]

Time to run a program is still:
\[ t' = t, \]

Energy required to run program:
\[ E' = P' \times t = E / M^2 \]

_argument in favour of voltage scaling, and parallel processing_

Rough approximations!
Where is the energy consumed? Target for the mobile phones

- Current trends violation of 0.5-1 W constraint for small mobiles; large mobiles: ~ 7 W
Energy-efficient architectures: Heterogeneous processors

(2) Telephony (W-CDMA)

<table>
<thead>
<tr>
<th>Baseband part</th>
<th>Control</th>
<th>ON</th>
</tr>
</thead>
<tbody>
<tr>
<td>W-CDMA</td>
<td>ON</td>
<td></td>
</tr>
<tr>
<td>GSM</td>
<td>ON / OFF</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Application part</th>
<th>System-domain</th>
<th>ON</th>
</tr>
</thead>
<tbody>
<tr>
<td>Realtime-domain</td>
<td>OFF</td>
<td></td>
</tr>
</tbody>
</table>

| Measured Leakage Current (@ Room Temp, 1.2V) | 407 μA |

Power on: Red
Power off: Black

ARM’s big.LITTLE as an example

Used in Samsung S4
Thermal Modeling: A Single Power Source

- Thermal conduction
  - Fourier’s Law of Cooling: the temperate change is proportional to the different of the chip and the ambient temperature (or the heat sink temperature)
    - If the chip is hotter, the temperature change drops more
    - If the chip is cooler, the temperature change drops less
  - Heating generation is proportional to the power consumption
    - If the power consumption is larger, the temperature change increases more
    - If the power consumption is smaller, the temperature change increases less
  - Therefore, \( T'(t) = uP(t) - v(T(t) - T_{amb}) \)
    - \( T(t) \) is the temperature of the power source at time \( t \)
    - \( P(t) \) is the power consumption of the power source at time \( t \)
    - \( T_{amb} \) is the ambient temperature. I will simple use it as 0. Why?
    - \( u \) and \( v \) are both hardware-dependent constants.
Solving Ordinary Differential Equation (ODE):

\[ T'(t) = uP(t) - \nu T(t) \]

It is a standard linear ODE, where \( u \) and \( \nu \) are constants:

\[
d \frac{T(t)e^{\nu t}}{dt} = e^{\nu t}d \frac{T(t)}{dt} + T(t) \cdot \nu e^{\nu t} = e^{\nu t}(uP(t) - \nu T(t)) + T(t) \cdot \nu e^{\nu t} = e^{\nu t}uP(t).
\]

\[
\int_{t_0}^{t} d \frac{T(t)e^{\nu t}}{dt} = \int_{t_0}^{t} e^{\nu t}uP(t) \Rightarrow T(t)e^{\nu t} - T(t_0)e^{\nu t_0} = \int_{t_0}^{t} e^{\nu x}uP(x)dx
\]

\[
\Rightarrow T(t) - T(t_0)e^{-\nu(t-t_0)} = \int_{t_0}^{t} e^{\nu(x-t)}uP(x)dx
\]

\[
\Rightarrow T(t) = T(t_0)e^{-\nu(t-t_0)} + \int_{t_0}^{t} e^{\nu(x-t)}uP(x)dx
\]

- The temperature effect at time \( t_0 \) decreases exponentially by \( T(t_0)e^{-\nu(t-t_0)} \).
- The power consumption effect at time \( x \) decreases exponentially by \( T(t_0)e^{\nu(x-t)} \), since \( \nu > 0 \) and \( x - t \leq 0 \) for \( x \leq t \).
Different Traces versus Temperature

a) 

b) 

T [K]

360

350

340

1 1.1 1.2 1.3 1.4 1.5

1 1.1 1.2 1.3 1.4 1.5

active idle
Thermal-Dependent Leakage Power Consumption

\[ I_{\text{avg}}(T, V_{dd}) = I(T_0, V_0) \left( A T^2 e^{\left( \frac{q_1 \cdot V_{dd} + q_2}{T} \right)} + B e^{\left( \gamma \cdot V_{dd} + \delta \right)} \right), \]

However, the term \( e^{(1/T)} \) does not provide significant role in the accuracy. It is possible to use a simpler formula to formulate the leakage current.

\[ \hat{I}_{\text{avg}}(T) = \hat{A} T^2 + \hat{B}, \]

Chuan-Yue Yang, Jian-Jia Chen, Lothar Thiele, Tei-Wei Kuo: Energy-efficient real-time task scheduling with temperature-dependent leakage. DATE 2010: 9-14
Dynamic Thermal Management (DTM)

- Avoid possible over heating
  - DVFS
  - DPM
Thermal Networks (only for your reference here)

- Thermal models of applications depend on neighbouring cores.
  - A resistance-capacitance (RC) thermal network is widely used
    - A set of first order differential equations
  - Steady states (the equilibrium temperatures if the power does not change)
    - Simple linear algebra
  - Transient states (temperature profile in time)
    - Approximate the solution by using fourth-order Runge-Kutta numerical method [HotSpot, Huang et al. 2009]
    - Exact solution by using matrix exponential (many approximations are available) methods [MatEx, Pagani et al. to be published in DATE 2015]

[Pagani, Santiago, Muhammad Shafique, Jian-Jia Chen and Jörg HenkelMatEx: Efficient Transient and Peak Temperature Computation for Compact Thermal Modelsin 18th Design, Automation & Test in Europe (DATE) 2015]
Power Budget / Power Constraint

- Abstraction: Not deal directly with temperature.
- Generally, a power budget (for thermal safety) is a single value:
  - For each core (per-core).
  - For the entire chip (per-chip).
Per-Chip / Per-Core Power Budgets

16 cores with area 5.3 mm²
Threshold temperature: 80° C
Power budget: 90 W

8 active cores
Highest Temperature: 80.0° C

6 active cores
Highest Temperature: 86.3° C

4 active cores
Highest Temperature: 98.8° C
Per-Chip / Per-Core Power Budgets

16 cores with area 5.3 mm²
Threshold temperature: 80° C
Power budget: 90 W

Highest Temperature: 80.0° C
8 active cores

Highest Temperature: 73.2° C
12 active cores

Highest Temperature: 69.5° C
16 active cores

Pagani et al.,
CODES+ISSS 2014
Problem with Per-Chip / Per-Core Power Budgets

16 cores with area 5.3 mm²
Threshold temperature for DTM: 80° C
Power budget: 90 W
Thermal Safe Power (TSP): Power Budget depending on # of activated cores

Power budget depends on the number of active cores
Safe for any ‘m’ active cores => Abstract mapping decisions

TSP table:

<table>
<thead>
<tr>
<th>Active Cores</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
<th>16</th>
</tr>
</thead>
</table>

Per-core budget

Per-chip budget (Estimated)
Summary

Hardware in a loop

- Sensors
- Discretization
- Information processing
  - Importance of energy/power efficiency
  - Special purpose HW very expensive
  - Energy/power efficiency of processors
  - Code size efficiency
  - Run-time efficiency
  - MPSoCs
- D/A converters
- Actuators