TKT-2431 Soc Design

Lec 11 – Energy consumption

Erno Salminen, Tero Arpinen

Department of Computer Systems
Tampere University of Technology

Fall 2010

Remember the Guest lecture+Conclusions Wed 1.12.2010 at 10:15
Contents

- Power consumption breakdown
- Low-power design at system level
- Dynamic power management
  - Clock gating, power supply shutdown
  - Dynamic voltage/frequency scaling
  - Low-power operating modes
  - Prediction methods
  - ACPI
Copyright notice

- Part of the slides
  - S. Dey, VLSI Advanced Topics, course material, UCSD

- Part of figures from
At first

Make sure that simple things work before even trying more complex ones

You should believe this by now...
“The power problem is the No. 1 issue in the long-term for computing. It's time for us to stop making 6-mile-per-gallon gas guzzlers. [...] Now you're going to see the great un-marketing of megahertz because it doesn't matter anymore."

Motivation (1)

- **Portable Systems**
  - Notebooks, palm-tops, PDA, cellular phones, pagers, etc.
  - Battery-driven - long battery life crucial
  - System cost, weight limited by batteries
    - 40W, 10 hrs @ 20-35 W-hr/pound = 7-20 pounds
    - Slow growth in battery technology
- **Must reduce energy drain from batteries**

- **Environmental Concerns**
  - EPA estimate: 80% of office equipment electricity is used in computers
  - "Energy Star" program to recognize power efficient PCs
  - Power management standard for desktops and laptops
- **Drive towards "Green PC"**

- **Thermal Considerations**
  - 10 °C increase in operating temperature => component failure rate doubles
  - Packaging: ceramic vs. plastic
  - Cooling requirements
- **Increasing levels of integration / clock frequencies make the problem worse**
  - 10cm², 500 MHz => 315Watts

- **Reliability Issues**
  - Electron-migration
  - IR drops on supply lines
  - Inductive effects
- **Tied to peak/average power consumption**

Motivation (2)

- A large and increasing number of devices are battery driven

- Desktop PCs (100-300 W)
- Laptop PCs (5 - 15 W)
- Multimedia Terminals (2 - 5 W)
- PDAs, Pocket PCs (0.5 - 3 W)
- Cellphones (0.5 - 1.5 W)
- Pagers (< 1W)
- Wireless Sensor Nodes (~mW)

Motivation (3)

- Batteries evolve at lower rate than other parts

- Fig: [John Hockenberry, Building a better battery, Wired, iss. 14.11, Nov 2006]
Motivation (4) : Cooling

Figure 1. The cost of powering and cooling servers has increased during the last decade and will increase steadily during the next few years, according to market research firm IDC. Reflecting this is the growing percentage of server purchase expenditures that energy costs represent.

Power consumption breakdown
Power breakdown: laptop

Energy breakup for a laptop with a wireless LAN NIC

[S. Dey, Design of Low-Power, Battery-Efficient Systems, ECE206C course material, UCSD, 2004.]
Power breakdown: Imagine stream processing chip

Smart memory hierarchy:

- memories ~21%
  - MBANKs 3%
  - UC SRAMs 3%
  - SRF SRAMs + SBs 15%
  - Clock Tree 11%
  - Other 5%
  - Cluster LRFs, Switch, and Control 21%

- Cluster ALUs 42%

- Imagine (0.18 μm – 48 FP ALUs)
  - 3.1 W, 132 MHz, 1.5 V (meas.)

- Power dissipation is dominated (>90%) by very predictable sources
  - RFs
  - ALUs
  - switches between ALUs
  - clocks

Mattan Erez, Stream Architectures –Programmability and Efficiency,

Tampere SoC, Nov. 17 2004
Power consumption in CMOS (1)

- Two measures
  - Peak power consumption
  - Average power consumption
    - Usually more interesting than peak power
    - However, large peaks degrade battery life-time and cause electronmigration

- $P_{avg} = P_{dynamic} + P_{short} + P_{leakage}$

---

Power consumption in CMOS (2)

- $P_{\text{dynamic}}$ has been dominant in CMOS (~50-90%)
- Leakage power likely has increased with smaller geometries!
  - E.g. $P_{\text{short}} + P_{\text{leakage}} = \sim 10\%@130\text{nm}$ but $40\%@65\text{nm}$
- $P_{\text{dynamic}} = K \times C_{\text{out}} \times V_{dd}^2 \times f$
  - $K =$ avg transitions on node per clock cycle
  - $C_{\text{out}} =$ driven output capacitance of node
  - $V_{dd} =$ supply voltage
  - $f =$ operating frequency

- Faulty circuits, may have also $P_{\text{static}}$
  - E.g. there is DC from $Vdd$ to GND, if gate of PMOS is stuck-at-zero

Muy importante!
Sources of power consumption

- Dynamic power dominates in logic
- Leakage power dominates in memory
  - Small activity
  - Dynamic: access one 32b word at a time in 32MB memory
  - Leakage: in all other 32MB-4B mem cells
  - Also in devices that are mostly in stand-by, e.g. cell phones
- Different methods must be applied

Example ASIC – 130nm worst case voltage (1.1 V)
worst case temperature (125°C)

Reducing dynamic power

- Minimize $P_{\text{dynamic}} = K \cdot C_{\text{out}} \cdot V_{\text{dd}}^2 \cdot f$
- Hence, minimize
  1. activity $K$
  2. the amount of logic (capacitance) $C_{\text{out}}$
  3. supply voltage $V_{\text{dd}}$ – quadratical impact!
  4. frequency $f$ – aim for "just fast enough"
  5. combination of the above
- Parameters are coupled
  - E.g. high $f$, requires large $V_{\text{dd}}$
  - Parallel processing may increase $C$ but lowers $f$ and $V_{\text{dd}}$
Capacitance and switching minimization

- Minimize K, i.e. useless switching
  - K depends on input sequence
  - Disable new values from entering the logic when results are not needed

- $C_{out} = C_{fo} + C_{w} + C_{p}$
  - $C_{fo}$ = input capacitances of fan-out gates (~50%)
  - $C_{w}$ = wiring capacitance (~40%, increases with new technologies), hard to estimate before placement
  - $C_{p}$ = parasitic capacitance of driving gate itself (~10%)

- No need to minimize $C$ if it is rarely switched
  - $C_{eff} = K \times C_{out}$, effective capacitance

- $C_{out}$ might increase when $C_{eff}$ minimized
  - Still beneficial
Power supply minimization

- Supply voltage has big effect
- Designer can rarely change the voltage freely
- Decreasing f and $V_{dd}$ together, saves energy
- Decreasing $V_{dd}$, increases delay of transistors ($=t_p$)

Inverter delay $t_p$

Voltage vs. Frequency vs. Power

Power vs. energy

- Batteries store **energy** not power
  - Power measures rate of energy consumption

**Energy** \( (E = P \times t) \) saving is often the real goal!

- Decreasing \( f \), increases \( t \)
- Frequency scaling alone does not decrease energy
- Execution time \( t \) is usually constrained

---

PDP and EDP

- **Power-Delay Product (PDP)** = \( P \times t \)
  - avg. **energy** consumed per switching event
  - Watt*sec = Joule

- **Energy-Delay Product (EDP)** = \( \text{PDP} \times t \)
  - avg. energy multiplied by execution time
  - Takes into account the trade-off between increased delay and lower energy/operation
Low-power design at system level
Importance of design level

- Applies to all design decision not just power

50-90%
40-70%
30-50%
20-30%
10-20%
5-10%

System level
Algorithm level
Register-transfer level
Gate level
Transistor level
Physical level

Alexander Worm, Algorithm Manipulation for Low-Power Communication Circuit Implementation, Tampere SoC, Nov. 20 2001. Who copied the figure from:

Power reduction methods

Systemic Areas For Power Reduction

- Clock gating
- Multi-voltage support
- Multi-threshold support
- Lower power clock tree synthesis
- Dynamic voltage drop
- Power grid synthesis and analysis
- Power gating for leakage opt. (state retention support)
- Power-aware placement
- Gate-biasing for leakage optimization

Importance in future:
- -
- -/+
- +
- -
- -
- +
- +
- -/+
- +

Barry Pangrle, Panels discussion "It's About Power - Performance and area alone don't quite cut it anymore!", DATE 2/14/2005.
## Methods

<table>
<thead>
<tr>
<th>Design Level</th>
<th>Dynamic Power</th>
<th>Leakage Power</th>
</tr>
</thead>
<tbody>
<tr>
<td>Physical Synthesis</td>
<td>Push min area for min speed spec</td>
<td>Push min area for min speed spec</td>
</tr>
<tr>
<td>Choice of Cell Library and Process Technology</td>
<td>Scaled VDD</td>
<td>Gate bias</td>
</tr>
<tr>
<td></td>
<td>Choice of process</td>
<td>Scaled VDD</td>
</tr>
<tr>
<td></td>
<td>Back bias</td>
<td>Multi-Vt</td>
</tr>
<tr>
<td>Configuration of Logic, Micro-architecture and Architecture</td>
<td>Clock gating</td>
<td>Functional clock gating+</td>
</tr>
<tr>
<td></td>
<td>Memory</td>
<td>gate bias</td>
</tr>
<tr>
<td></td>
<td>configuration/size, pipeline depth</td>
<td>Memory size, pipeline depth</td>
</tr>
<tr>
<td></td>
<td>Instruction set</td>
<td>Instruction set</td>
</tr>
<tr>
<td>System Design</td>
<td>Processor configuration</td>
<td>Processor configuration</td>
</tr>
<tr>
<td></td>
<td>Sleep process</td>
<td>Sleep process</td>
</tr>
<tr>
<td></td>
<td>Energy management, Dynamic frequency and voltage scaling</td>
<td>Gate bias, substrate bias</td>
</tr>
</tbody>
</table>

*Figure 2. Power reduction methods*

Choose the right implementation

- Point solutions are of course most efficient w.r.t to power
  - Reduced flexibility
- Large differences: $9x-1075x$!
- Pay attention to ratio $\text{active}_P/\text{idle}_P$
  - Poor ratio (small) in general-purpose devices
System level: Choose the right digital architecture (1)

- 0.5-5 MIPS/mW
- 10-100 MOPS/mW
- 100-1000 MOPS/mW

- Direct Mapped hardware
- Embedded FPGA
- Embedded Processor (IpArm)
- Reconfigurable Processors (Maia)
- DSP (TI C6xxx)

Factor of 100-1000

Power Dissipation

System level: Choose the right digital architecture (2)

Intercom TDMA MAC Implementation alternatives

<table>
<thead>
<tr>
<th></th>
<th>ASIC</th>
<th>FPGA</th>
<th>ARM8</th>
</tr>
</thead>
<tbody>
<tr>
<td>Power</td>
<td>0.26mW</td>
<td>2.1mW</td>
<td>114mW</td>
</tr>
</tbody>
</table>

- ASIC: 1V, 0.25 μm CMOS process
- FPGA: 1.5 V 0.25 μm CMOS low-energy FPGA
- ARM8: 1 V 25 MHz processor; n = 13,000
- Ratio: 1 - 8 - >> 400

[Jan Rabaey, System-on-a-chip: A case for heterogeneous architecture, Tampere Soc, 1999].
# Reminder: opt for locality

<table>
<thead>
<tr>
<th>Operation</th>
<th>Energy (0.13um)</th>
<th>Energy (0.05um)</th>
</tr>
</thead>
<tbody>
<tr>
<td>32b ALU Operation</td>
<td>5pJ</td>
<td>0.3pJ</td>
</tr>
<tr>
<td>32b Register Read</td>
<td>10pJ</td>
<td>0.6pJ</td>
</tr>
<tr>
<td>Read 32b from 8KB RAM</td>
<td>50pJ</td>
<td>3pJ</td>
</tr>
<tr>
<td>Transfer 32b across chip (10mm)</td>
<td>100pJ</td>
<td>17pJ</td>
</tr>
<tr>
<td>Execute a uP instruction (SB-1)</td>
<td>1.1nJ</td>
<td>130pJ</td>
</tr>
<tr>
<td>Transfer 32b off chip (2.5G CML)</td>
<td>1.3nJ</td>
<td>400pJ</td>
</tr>
<tr>
<td>Transfer 32b off chip (200M HSTL)</td>
<td>1.9nJ</td>
<td>1.9nJ</td>
</tr>
</tbody>
</table>

1:20:260 local to global to off-chip ratio today
1:56:1300 in 2010

Locality is key to achieving energy efficiency

Mattan Erez, Stream Architectures –Programmability and Efficiency,

Tampere SoC, Nov. 17 2004
Will new technologies minimize heat?

Figure 1. Power density comparison of CMOS multi chip modules between bipolar and CMOS chip technology.

Methods for dynamic power management
Full speed is not required all the time

- Running at full speed wastes energy
- The workload is NOT constant
  - But hard to forecast at design-time

⇒ Adapt dynamically


Figure 1. Average CPU utilization of more than 5,000 servers during a six-month period. Servers are rarely completely idle and seldom operate near their maximum utilization, instead operating most of the time at between 10 and 50 percent of their maximum utilization levels.
Full speed is not required all the time (2)

- Servers do use dynamic management
  - Far from ideal
- Energy seems to be way too cheap

Figure 2. Server power usage and energy efficiency at varying utilization levels, from idle to peak performance. Even an energy-efficient server still consumes about half its full power when doing virtually no work.
Dynamic power managements (DPM)

- Configures electronic systems at run-time to provide required performance with minimal activity
- Applicable if components experience non-uniform workload
  - Predictable periods of activity and idleness
  - e.g. simple timeout policy in laptops shuts down components if they have been idle for certain period
- Power manageable component (PMC) has two or more modes of operation
  a) High performance and power consumption
  b) Low performance and power
- Usually the number of modes very limited
### Power control unit

- Determines when FU is shut down
  - Rule of thumb: pow_ctrl area max 1/10 FU area
- **Power consumption of power control must be smaller than resulting savings**
- Either shut down the Clk and/or V_{dd}
  - State is lost, if V_{dd} shut down
  - Internal idleness requires keeping state
- Shutdown and recovery have non-negligible delays
  - Shutting down is not beneficial if sleep state is short
  - Hard to determine to optimal timing
  - Performance loss due to recovery time

---

**Diagram:**

- a) Power supply shut-down
- b) Clock gating principle
Isolating the shutdown units

- Isolate sleeping unit from its neighbors
- Drive control and status signals into inactive state
  - E.g. sleeping FIFO appears full and other won’t try to write data to it
  - At the same time, sleeping FIFO appears empty and other cannot read from it
- Sometimes all output signals must be frozen during sleep (even if power is cut)
Power supply shutdown

- Cut of power supply with switch
  - Removes also $P_{\text{leakage}}$
  - Switch has non-ideal delay $\Delta T$ and resistance $R$
- Physical placement and layout important due to transient noise
- Volatile memories (RAM, flip-flops) lose their state
  - Must be saved elsewhere and restored
  - Longer shutdown/wakeup delays
- Utilized in coarse-grain not with small units inside the chip
Power supply shutdown example
Clock gating

- Clock of FU is shut down
  - Needs latch and AND gate to avoid glitches
  - Adds clock skew
- Input values do not propagate through input registers
  - No switching inside FU
- Relatively simple and low overhead control
  - Automatic clock gating supported in CAD tools
  - Small grained control, even at the level of a couple of DFF’s
Clock gating (2)

- Well suited for self-managed components
- Clock distribution network itself consumes energy
  - Highly active (K=1)
  - Large net = large capacitance
  - Stop master clock PLL or oscillator
  - However, most energy consumed by local clocks
  - GALS approach may help since large global clock network may be split into several small clock networks
Clock power in Pentium

- 30% of the total power is attributed to clock
- Most of the clock power is used in the final clock buffers and flip-flops

Anderson, ISSCC-2002
Reducing clock network power

Reduce clock frequency
- Multiple frequency domains
- Dual edge triggered flip-flops

- Reduce voltage swing
  - Low swing clocks

Clock Power = f * C * V^2

- Reduce clock loading
  - Clock gating
  - Clock-on-demand flip-flop
  - Optimized routing

In GALS, all local clocks can be optimized separately

Dynamic Frequency/Voltage Scaling (DVS / DFS)

- DFS: frequency is changed at runtime
- DVS: both frequency and voltage are changed at runtime

\[ E(\text{orig}) = t \cdot P \]
\[ E(\text{DFS}) = 2t \cdot \frac{P}{2} = E(\text{orig}) \]
\[ E(\text{DVS}) = 2t \cdot \frac{P}{4} = \frac{E(\text{orig})}{2} \]
DVS/DFS: Idle power is not zero!

- More realistic example:
  - Tasks are initiated with 16 cycle interval
    - measured as original cycles
  - Total energy = active + idle energy

<table>
<thead>
<tr>
<th>Orig</th>
<th>DFS</th>
<th>DVS</th>
</tr>
</thead>
<tbody>
<tr>
<td>P (act)</td>
<td>1.0</td>
<td>0.5</td>
</tr>
<tr>
<td>P (idle)</td>
<td>0.2</td>
<td>0.2</td>
</tr>
<tr>
<td>Cycle time</td>
<td>1.0</td>
<td>2.0</td>
</tr>
<tr>
<td># Cycles (act)</td>
<td>8.0</td>
<td>8.0</td>
</tr>
<tr>
<td># Cycles (idle)</td>
<td>8.0</td>
<td>0.0</td>
</tr>
<tr>
<td>Energy (act)</td>
<td>8.0</td>
<td>8.0</td>
</tr>
<tr>
<td>Energy (idle)</td>
<td>1.6</td>
<td>0.0</td>
</tr>
<tr>
<td>Energy (tot)</td>
<td>9.6</td>
<td>8.0</td>
</tr>
<tr>
<td>Saving %</td>
<td>-</td>
<td>16.7</td>
</tr>
</tbody>
</table>
Controlling DPM
Power states

- Low power states have
  - lower performance
  - longer transition latency

Fig. 1. Power state machine for the StrongARM SA-1100 processor.
Power-managed systems

- Observer collects workload information
- Controller forces transitions between power states
- In large networked systems, observations and controls cannot be centralized
  - Problematic

Fig. 2. Abstract structure of a system-level power manager.
Power state machine

- Trivial greedy policy can be used, if transitions are instantaneous and consume no power
  - Not realistic assumption
- Returning from power-down mode requires
  1. turning on and stabilizing power supply
  2. reinitializing system
  3. restoring context
  - non-negligible delay and energy
- Tolerated performance degradation must be explicitly stated
  - Max power saving when device is not designed at all. However, performance loss is 100%...
Power state machine (2)

- Strong ARM SA-1100 (cf. fig 1) can be modeled with FSM below
- Transition RUN ⇔ IDLE is so fast that
  - Greedy policy applicable
  - they can be combined into single state ON (self-managed with greedy policy)
- $P_{ON}$ is weighted sum of $P_{RUN}$ adn $P_{IDLE}$
- OFF coresponds to state Sleep

Fig. 4. PSM of a two-state power-manageable component.
Break-even time (1)

- No computation possible during state transition → performance loss
- Break-even time $T_{BE}$ for inactive state is the minimum inactivity time required to compensate the cost of state transition(s)
  - Cost depends on transition times and power consumption(s)
- If inactive time $T_n < T_{BE}$, it is not beneficial to enter inactive state because cost is not compensated
Break-even time (2)

- In simple case, $T_{BE}$ is sum of *time for entering state* and *exiting state*
  - Assuming that state transition does not increase power consumption (like it does with hard-drives)

- Multiple power states result in multiple break-even time values
Applicability of DPM

- One can calculate the max power saving
  \[ P_{\text{saved,max}} = P_{\text{on}} - P_{\text{ideal}} \]
  where \( P_{\text{ideal}} \) refers to \( P \) with ideal DPM

- States with small \( T_{\text{BE}} \) are more likely applicable, e.g. Strong-ARM
  \[ T_{\text{BE, idle}} = 0.02 \text{ ms} \]
  \[ T_{\text{BE, sleep}} = 169.09 \text{ ms} \]
  \( \Rightarrow \) Idle state can be entered much more often

Fig. Example of ideal DPM policy (workload known a priori, and hence wakeups always on time and no performance loss)
Applicability of DPM (2)

- In most cases, workload is *not known* a priori and DPM *reacts* to the changes.
- Sometimes, workload is known, e.g. sampling sensor values once per second and processing them.

Fig. Difference between ideal and realistic DPM policies
Prediction

- In real world, little information (or not at all) is available about future inputs
- Must predict

- Overprediction/Underprediction
  - Predicted idle period longer/shorter than actually
- Overprediction causes performance loss
  - Not enough time for wakeup
- Underprediction consumes unnecessary power
  - Low power mode not entered always
Prediction methods

a) Fixed timeout:
- When elapsed idle time longer than threshold, enter low-power mode
- Big threshold increases performance and power
- Waste power when waiting for timeout
- Performance loss upon wakeup

b) Predictive shutdown
- Predict idle time from duration past idle and active periods
  - No automatic way to decide regression equation
  - Offline data collection required
- Predict idle time from last active period
  - Short active periods are usually followed by long idle periods
  - Offline data required
c) Predictive wakeup

- To reduce performance loss
- When elapsed time in low-power mode longer than threshold, start wakeup procedure
  - Increases power in idle period longer than predicted

d) Adaptive methods change threshold at runtime

- E.g. use several timeout values and measure how well they perform
Selecting $T_{BE}$ in fixed timeout

Interactive programs (e.g. games) have shorter idle periods.

With high break-even time, low-power mode seldom used.

---

Plot of $P(T)$ for the Sleep state of the StrongARM SA-1100 processor. The three curves refer to three different workload statistics, computed from real-world CPU traces provided by the IPM monitoring package [5].
Safety vs efficiency

Safety means probability of avoiding performance loss → there's always some loss

Efficiency means proportion of achieved power saving from ideal saving

Quality of a timeout-based predictor evaluated as a function of timer duration. Safety and efficiency of the timeout used to predict idle periods longer than T=160 ms.
Predictive shutdown

Example threshold (i.e. active periods shorter than this are likely followed by long idle time)

L-shape is necessary condition for prediction

Fig. 7. (a) Scatter plot of $T$ versus $T$ for the workload of the CPU of a personal computer running Linux.
ACPI

- **Advanced Configuration and Power Interface**
  - by Intel, Microsoft and Toshiba
- Defines interfaces between OS and HW
- Targets personal computers (PCs)

Fig. 11. ACPI interface and PC platform.
ACPI (2)

- System has 4 global power states
  - G0 = ON, G3=OFF
  - Additional state *legacy* if devices don’t support ACPI
- State G1 *(sleeping)* divided into 4 sub-states
- State G0 (ON) divided into 4 device states and 4 processor states

Fig. 12. State definitions for ACPI.
Case: ACPI with hard disk

- Power management SW takes <1% of time
- Wakeup power (52.5J / 7s) is larger than active power
  - Due to inertia when disks start rotating
- Break-even time 17.6 sec
- Power reduction 23- 55%

Fig. 14. PSM for IBM DTTA HDD.
**Case: Results**

**TABLE I**

**Disk Parameters: Subscripts**

*sd* and *wu* denote Shut Down and Wake Up, respectively.

<table>
<thead>
<tr>
<th>Model</th>
<th>$P_{Off}$</th>
<th>$P_{On}$</th>
<th>$T_{sd}$</th>
<th>$E_{sd}$</th>
<th>$T_{wu}$</th>
<th>$E_{wu}$</th>
<th>$T_{break-even}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>IBM</td>
<td>0.75</td>
<td>3.48</td>
<td>0.51</td>
<td>1.08</td>
<td>6.97</td>
<td>52.5</td>
<td>17.6</td>
</tr>
<tr>
<td>Fujitsu</td>
<td>0.13</td>
<td>0.95</td>
<td>0.67</td>
<td>0.36</td>
<td>1.61</td>
<td>4.39</td>
<td>5.4</td>
</tr>
</tbody>
</table>
Conclusion

- Power saving does not necessarily save energy

- Basic methods
  - Low-power technology, signal reordering
    - Minor effect, not interesting
  - Voltage scaling, power shutdown
    - Difficult, voltage levels CANNOT be freely chosen
  - Frequency scaling
    - Must not sacrifice performance too much
    - Does not affect energy/task, if used alone
  - Clock gating, enabled flip-flops
    - Reasonable way for energy saving
    - Supported by CAD tools
Conclusion (2)

- Several power/performance modes needed
  - Modes have different break-even times
- Policy defines the current operating mode
  - Static
  - Adaptive
- Policy decisions based on
  - off-line data
  - monitoring
Glitch Minimization

- Low-level technique
- Glitches may add 20% to power [Raghunathan, DAC96]
- Raghunathan et al. suggest RTL modifications to decrease glitches
  - Stop glitch propagation (e.g. with registers)
  - Glitch generation (due to uneven gate delays) not considered
- Important to avoid glitches in control signals
Shutting down units

- Unit is idle when its output values are not needed
  - Hence, it can be shut-down

  a) **External idleness** – changes in units output are not visible in system outputs
     - Output of ADD is *don’t care*

  b) **Internal idleness** – units output do not change even if units inputs change
     - State-holding required

- Not practical to detect all idle conditions
  - Too large overhead
  - Detect most common

- ADD

- SUB

'a) ADD is externally idle'

'b) FU is internally idle'
How Free Is Solar Energy?

Renewable-energy technologies promise to liberate us from fossil fuels. But this implies that their energy payback periods—the time it takes for a system to recover the energy used to produce it—is just as important as financial payback. If you install solar cells on your roof, you want the system to pay for itself eventually, but you also want to help your country get a grip on global warming and stop depending on foreign oil supplies.

"Most people who take the initiative to put photovoltaics on their homes and businesses are looking at the economics, certainly, but they’re looking beyond that too," in the words of Gary Schmitt, a spokesman for the National Renewable Energy Laboratory, in Golden, Colo. A key variable is how solar cells perform once installed. The energy available to them could be close to 7 kilowatt-hours per square meter per day in Phoenix or 2 kilowatt-hours/m² per day in Moscow [map]. For the most common types of cells, which use multicrystalline silicon, the energy payback periods can be between one and four years [chart, bottom right]. Thin-film solar cells, which eventually will be much cheaper than multicrystalline silicon cells, have shorter energy payback periods. Cadmium-telluride cells recover their energy inputs in 10 to 22 months. Most of the energy spent on solar modules goes into purifying the materials and encapsulating the modules. Vasilis Fthenakis, a scientist at Brookhaven National Laboratory, estimated the environmental footprint of solar systems, using assumptions about the transportation distances for materials and the amounts of energy needed to produce the cells, the modules, and the electrical and electronic devices [chart, bottom left].

Of course, the attractiveness of renewable technologies depends not only on their energy paybacks, but too often, energy technologies are discussed solely in terms of their direct cost to end users without taking into account surrounding factors, Fthenakis says. “Taken together, they define the total cost of a renewable energy system.”

Global Warming Potential Of Energy Options

There’s no free lunch, but several alternatives to fossil fuels come close.

When Solar Cells Break Even

A multicrystalline-silicon photovoltaic system needs to operate for more than a year to recover the energy invested in its manufacture. The values for 10 cities [map, above; chart, below] apply to modules tilted at an optimal angle to the sun.

1. Phoenix
2. Johannesburg
3. Madrid
4. Adelaide, Australia
5. New York City
6. São Paulo, Brazil
7. Feldberg, Germany
8. Tokyo
9. Moscow
10. Santiago, Chile
Stochastic methods

- Take into account
  - uncertainty in workload, power consumption, and response times
  - many power states, buffers, queues etc.
- Offer controlled trade-off between performance and power
- Controlled Markov chains
  - Service requester (SR) models workload
  - Service provider (SP) model power modes
  - Power manager implements commands for SP
  - Cost metrics combines power and performance
Stochastic methods (2)

- State transitions have probabilities
- Bursty workload in Fig 9a)
  - High probability (0.85) for several requests in a row
  - Average request stream $1/(1-0.85) = 6.7$ requests

0 = no request/workload
1 = request issued

(a) SR = workload

(b) SP = power modes

Fig. 9. Markov model of a power-managed system and its environment.
Stochastic methods (3)

- Power mode is changed with commands `switch_ON` and `switch_OFF`
- Transition probabilities model the transition delay
  - Even if `switch_OFF` is issued, transition does not occur immediately
- Advantages
  - Possible to search global optimum
  - Exact solution in polynomial time
  - Strength and optimality of randomized policies
- Note
  - Performance and power are expected values, no guarantees given
  - Hard to obtain accurate Markov models