LOW-POWER DESIGN: OUTLINE

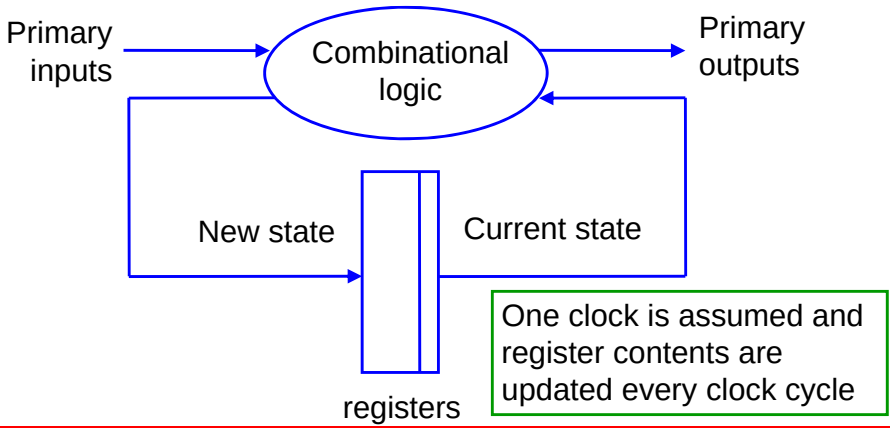
- Wrap-up register-transfer level design
- Implementation techniques: CMOS, standard cells, clock trees
- Sources of power dissipation
- Signal transition probabilities
- Power simulation and estimation techniques
- RT-level low-power design: with variable and constant voltage
- Architectural considerations
- Software-level control
- Asynchronous design
- Further reading

RELATED COURSE

- 2nd quarter course:

Energy-Efficient Embedded Systems (192130122)

REGISTER-TRANSFER (RT) VIEW OF HARDWARE



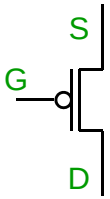
REGISTER-TRANSFER MODEL IN PRACTICE

- An IC with some degree of complexity will have more than one clock.
- Register-transfer model is valid for each clock separately.
- One says that registers connected to the same clock belong to the same *clock domain.*
- One needs to take special care of signals crossing clock domains.

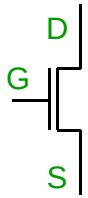
CMOS TECHNOLOGY

- CMOS = *complementary metal-oxide-silicon*
- Building blocks: *n-type* and *p-type transistors*
- A transistor can be considered to act as a switch.

- A *p-type* transistor conducts current when its gate (G) is *low.*
- So, S=D when G=0; S is isolated from D when G=1.

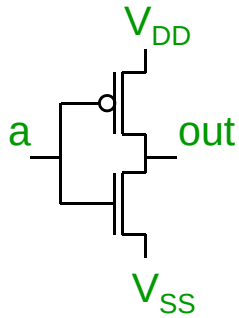


- An *n-type* transistor conducts current when its gate is *high.*
- So, S=D when G=1; S is isolated from D when G=0.



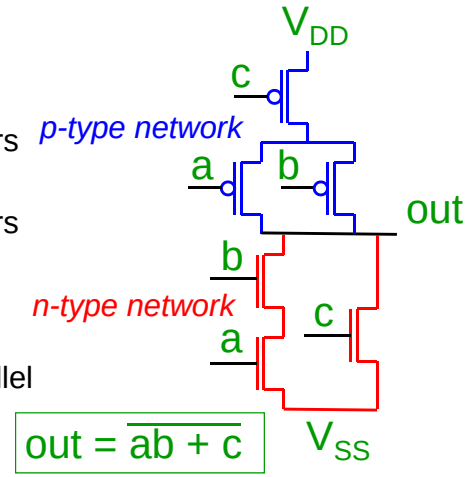
CMOS INVERTER

- The inverter: out = not(a)
  - When a = 1, out is connected to V_{SS}, so out = 0.
  - When a = 0, out is connected to V_{DD}, so out = 1.
  - Ideally, no *static* current! ("static" means: stable input value)
  - Some *dynamic* current, see later.



CMOS COMPLEX GATE

- A CMOS gate consists of two networks:
  - Network of p-type transistors for the connection to V_{DD}
  - Network of n-type transistors for the connection to V_{SS}
  - The two networks are complementary: a series connection in one is a parallel connection in the other.
  - Ideally, no static current!



$\text{out} = \overline{ab + c}$

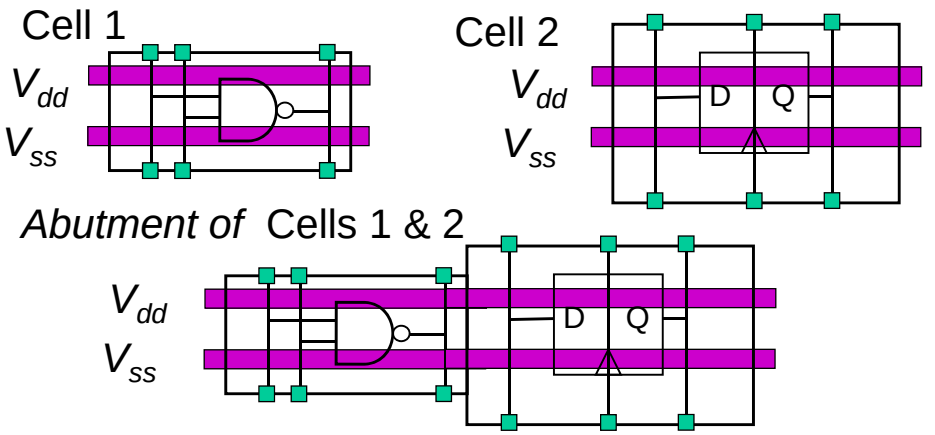
STANDARD-CELL BASED DESIGN (1)

- The designer is given a *library* of elementary circuits called *cells.*
- The design of library cells (from transistors) is a specialized task and is done by a dedicated group or company.
- Library information consists of:
  - cell function and simulation models (delays!)
  - cell layouts.

STANDARD-CELL BASED DESIGN (2)

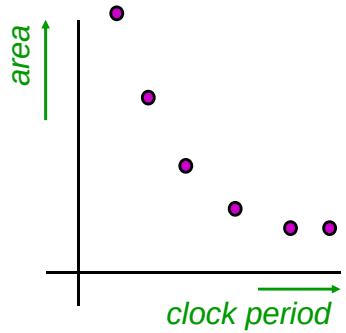
- Designing amounts to delivering a *netlist* of cells for layout.
- The netlist is generally obtained by means of *logic synthesis.*
- The netlist is mapped on a layout by means of *placement* and *routing.*

STANDARD-CELL LAYOUT



TIME-AREA TRADE-OFF IN LOGIC SYNTHESIS

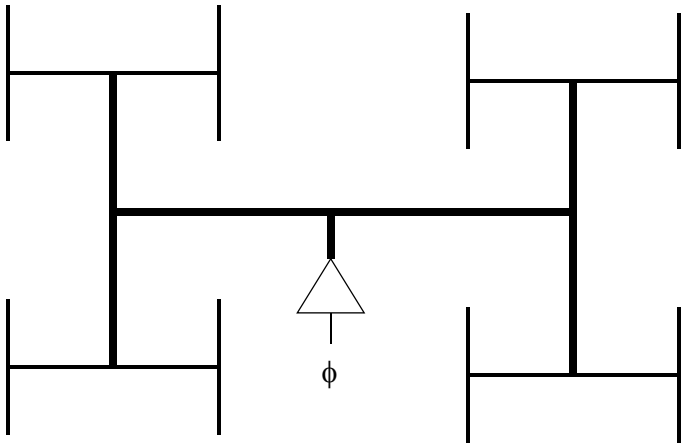
- In general, one can make a circuit faster by using more area.
- Example: a *carry-ripple adder* is small but relatively slow; a *carry-lookahead adder* is faster, but larger.
- In logic synthesis one constrains the clock period and the synthesis tool will choose the implementation that fits the constraint.



CLOCK DISTRIBUTION

- Goals:
  - deliver clock to all memory elements with acceptable *skew* (“skew”: difference between earliest and latest arrival of clock edge to a register);
  - deliver clock edges with acceptable sharpness.
- Clocking network design is one of the greatest challenges in the design of a large chip.

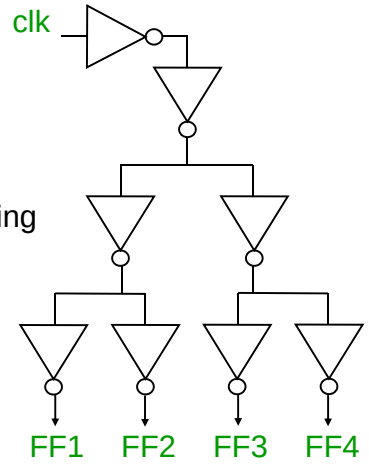
H-TREE



Wire length from source to any endpoint is equal!

CLOCK TREE

- In order to balance the delay from the clock source to the flip-flops, *clock trees* are used.
- In current-day practice clock trees are generated during layout as wiring delay is significant.

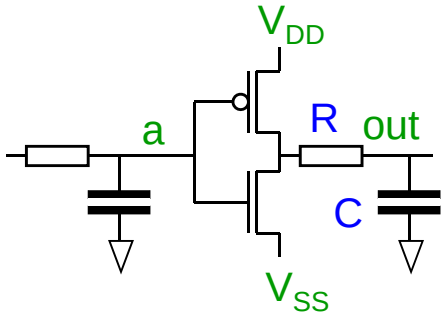


SOURCES OF POWER DISSIPATION IN CMOS

- Charging and discharging nodes due to switching: *capacitive dissipation.* Energy loss:

$$C \times V_{DD}^2$$

- A direct short-circuit path through both p-type and n-type transistors during switching: *short-circuit dissipation*
- Leakage



POWER DISSIPATION EQUATION

- The power related to charging and discharging nodes is:

$$P = f \times C_{\text{eff}} \times V_{\text{dd}}^2$$

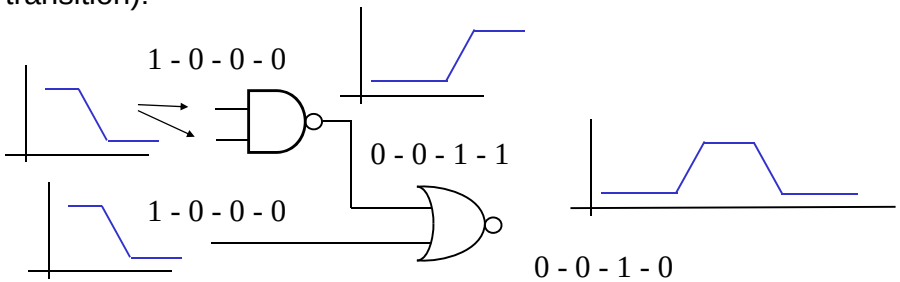
- Where:
  - f : clock frequency
  - C_{eff} : *effective* capacitance, average capacitance switched in one clock cycle
  - V_{dd} : power-supply voltage

WAYS TO REDUCE POWER

- Reduce clock frequency
- Reduce effective capacitance
- Reduce supply voltage:
  - Has a quadratic effect on power savings
  - Has a negative effect on switching delay
- Avoid unnecessary switching (glitching)

GLITCHING: SWITCHING NOT CONTRIBUTING TO FUNCTION

- Example, using *unit-delay* simulation (1 time unit for each signal transition):



- Glitches *propagate* through the logic resulting in unnecessary power consumption

SWITCHING ACTIVITY: TRANSITION PROBABILITY

- Probability of some signal *s* being logic '1': P_s
- Then the signal transition probability for a signal that is *temporally uncorrelated,* becomes:

$$P_{\text{tr}} = 2P_s(1 - P_s)$$

- Examples:

$$P_s = \frac{1}{2} \rightarrow P_{\text{tr}} = \frac{1}{2} \qquad P_s = \frac{1}{4} \rightarrow P_{\text{tr}} = \frac{3}{8}$$

SWITCHING ACTIVITY FOR LOGIC GATES

- Probability of gate inputs being '1' can be translated to probability of outputs being one:

$$\text{Inverter: } P_{\text{out}} = 1 - P_{\text{in}}$$

$$\text{AND gate: } P_{\text{out}} = \prod_k P_{\text{in}_k}$$

$$\text{OR gate: } P_{\text{out}} = 1 - \prod_k (1 - P_{\text{in}_k})$$

- Assumption: inputs switch independently (often not true)

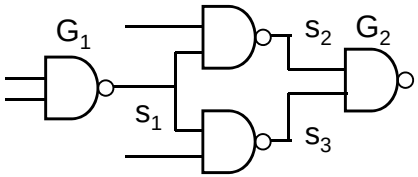
TEMPORALLY CORRELATED SIGNALS

- One should be careful with the assumption that signals are temporally independent.
- A clock signal, for example, will always have 2 transitions per clock period, even when its probability of being '1' is 0.5.
- Consider the MSB of a sine wave sampled 20 times per period and encoded in 2's complement. Its probability of being '1' is 0.5. However, its transition probability is 0.1 (two transitions per period).

SPATIALLY CORRELATED SIGNALS

- Signal having a common origin will be *spatially correlated.*
- In case of *reconvergent fanouts* independence is lost.

Output s_1 of G_1 *reconverges* in G_2



s_2 and s_3 are spatially correlated

- MSBs in a bus, representing mostly small numbers, using 2's complement encoding, will also be spatially correlated.

POWER-ESTIMATION TECHNIQUES (1)

- Circuit-level simulation
  - Use e.g. SPICE
  - All currents and voltages are known
  - Power can be computed accurately
  - Time consuming
- Gate-level simulation
  - Simulate standard-cell netlist
  - Keep track of switching activities
  - Combine with known power numbers for each cell
  - Glitches are taken into account

POWER-ESTIMATION TECHNIQUES (2)

- Analysis-based estimation:
  - Propagation of probabilities
  - Correction for correlated signals
- Architecture-level estimation:
  - Based on characterization of building blocks, such as ALUs, memories, etc.
  - Combined with switching activity at the blocks' inputs.

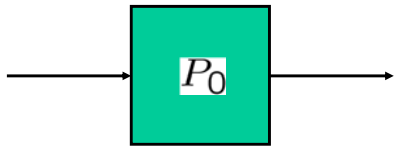
VOLTAGE REDUCTION

- Most effective
- Difficult to apply on individual designs:
  - Standard-cell library has been characterized for a specific voltage range.
- Even more difficult to use multiple voltages:
  - For the digital circuitry on the same IC
  - Within the same clock domain.
- Still, quite some research has been performed on designing with multiple voltages in the same clock domain
- Not supported by common logic synthesis tools

AREA-POWER TRADE-OFF (1)

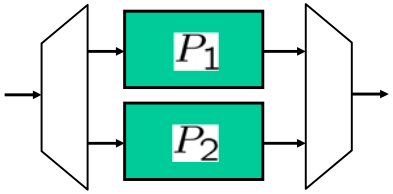
- Suppose a processor P_0 with effective capacitance C_{eff} performs some task with given deadline operating at some frequency-voltage combination (f, V_{dd}).
- So, its power consumption is:

$$P = f \times C_{\text{eff}} \times V_{\text{dd}}^2$$



AREA-POWER TRADE-OFF (2)

- Suppose now that two processors can be deployed to perform the task:
  - The effective capacitance will more than double (some overhead to distribute tasks).
  - The task's deadline can now be met by reducing the operating frequency to about half the original value.
  - No positive effect on power if voltage is kept constant.



AREA-POWER TRADE-OFF (3)

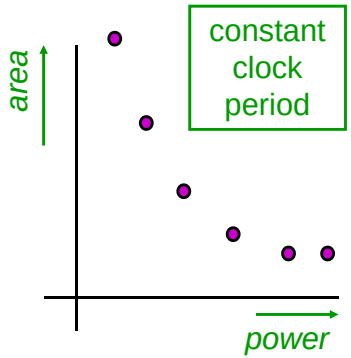
- As the frequency was halved, the voltage can also be more or less halved without causing circuit malfunction (of course, the voltage cannot be reduced indefinitely).
- In the new situation:

$$\left.\begin{aligned} C'_{\text{eff}} &= 2.1 C_{\text{eff}} \\ f' &= 0.5 f \\ V'_{\text{dd}} &= 0.5 V_{\text{dd}} \end{aligned}\right\} \quad \begin{aligned} P' &= 0.26 f \times C_{\text{eff}} \times V_{\text{dd}}^2 \\ &= 0.26 P \end{aligned}$$

Power reduction at the expense of area!

COMPUTATIONAL MARGIN

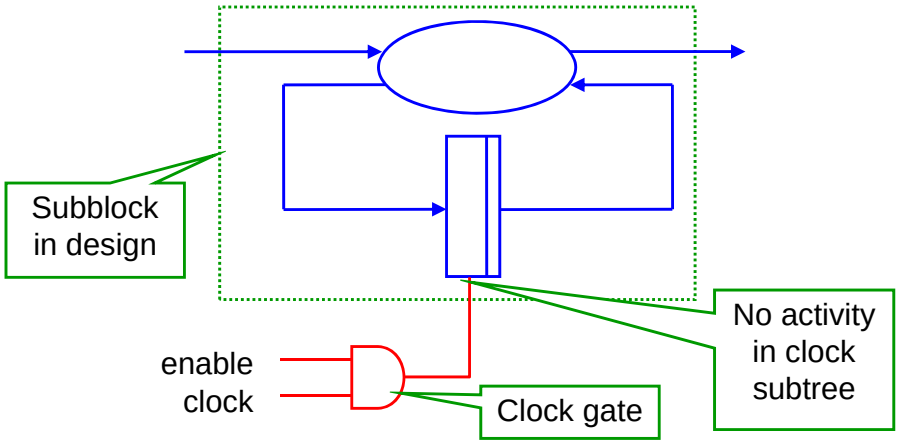
- Critical path determines maximal clock period.
- Critical path may be reduced by increasing parallelism.
- Use the created *slack* or *computational margin* in the clock period, to reduce voltage.
- So, there may be power-area trade-off (if the effective capacitance has less than quadratic growth).



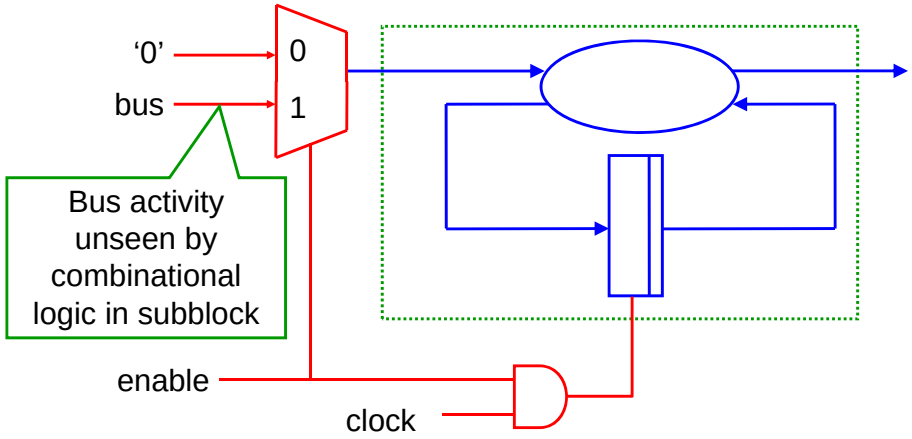
CONSTANT-VOLTAGE RTL POWER REDUCTION

- Targeted to removing unnecessary switching including glitching.
- The easiest thing is to turn off entire block of a design when not in use:
  - Example: some blocks on a radio IC are only used for *transmitting* others only for *receiving;* the IC will probably do neither most of the time.
  - Implemented by *clock gating*

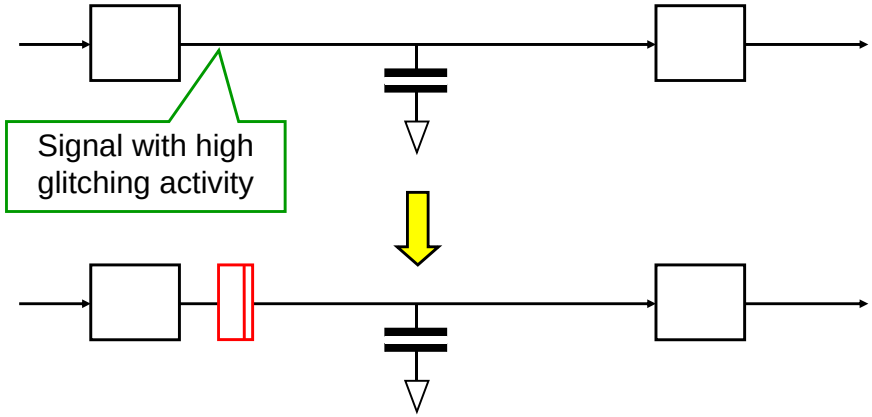
BLOCK-LEVEL CLOCK GATING



PREVENT COMBINATIONAL ACTIVITY

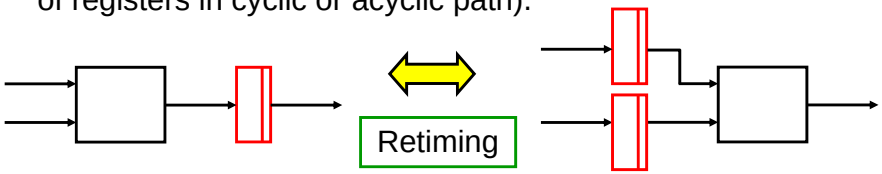


PREVENT GLITCH PROPAGATION WITH REGISTERS (1)



PREVENT GLITCH PROPAGATION WITH REGISTERS (2)

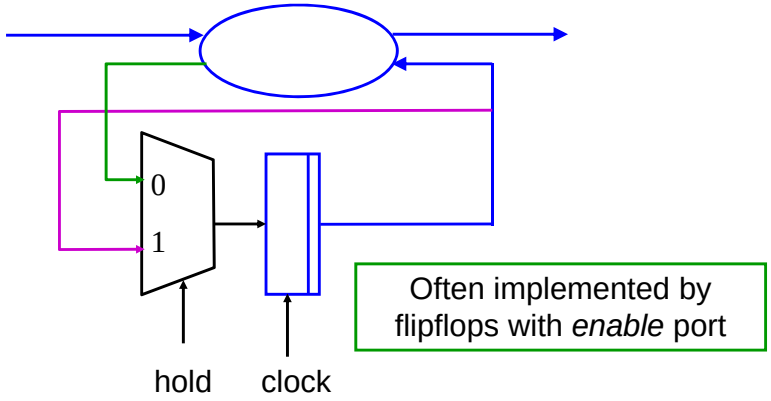
- One normally can't just add registers without modifying circuit functionality:
  - *Pipelining* is allowed in acyclic data flow; it increases the *latency* (number of clock cycles for input to propagate to output).
  - *Retiming* is always allowed (it does not change the number of registers in cyclic or acyclic path).



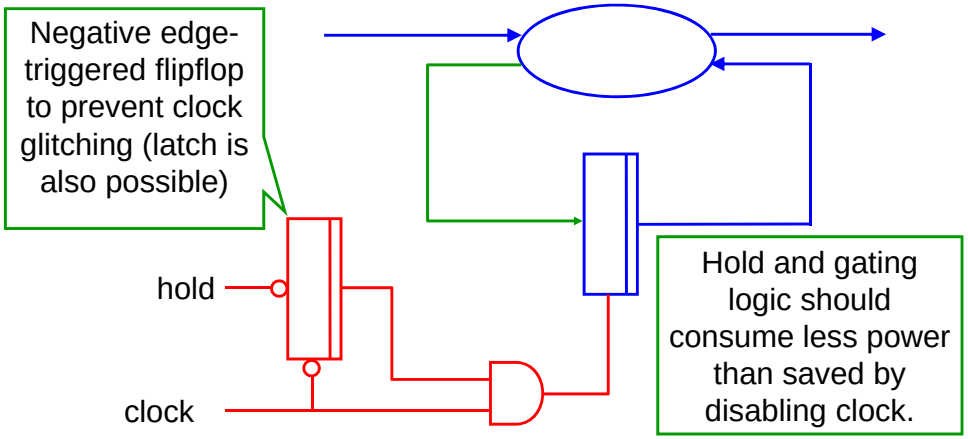
INTRA-BLOCK CLOCK GATING (1)

- Registers do not always change value at each clock cycle; they sometimes keep their previous value (e.g. no "else" branch for new state computation in VHDL).
- In the latter case, the clock can be disabled.
- This saves power consumption in the clock tree and inside the register.
- This idea is encountered in many places, for example:
  - Theeuwen, F. and E. Seelen, *Power Reduction Through Clock Gating by Symbolic Manipulation,* VLSI '97, Gramado, Brazil, (1997).
- Also implemented in Synopsys tools.

INTRA-BLOCK CLOCK GATING (2)



INTRA-BLOCK CLOCK GATING (3)



POWER GATING

- Clock gating preserves the circuit's state.
- For situations when batteries should last extremely long (several years), as for a node in a sensor network, clock gating is not enough. Leakage cannot be neglected.
- The solution is to turn off the power: *power gating*.
- Only a small subcircuit remains powered:
  - It periodically turns on and off the processor;
  - It stores the state information;
  - It may operate at a reduced clock frequency.

ARCHITECTURAL CONSIDERATIONS

- Traditional processor has the *Von Neumann* architecture:
  - Program code and data in same memory.
  - This *bottleneck* not only affects performance but power as well (larger memories require more energy per access).
  - Time and energy is lost in memory transfers.
  - Multiplexing data and code streams destroys temporal correlation.
- Low-power architectures have *locality of reference*:
  - a high degree of parallelism
  - with local memories to avoid data transport across large distances.

SOFTWARE-LEVEL LOW POWER DESIGN

- Compact code not only reduces the clock cycles needed for some computation, but also the switching activity and therefore the power consumption.
- Software (e.g. operating system that knows about the tasks to be performed) can control:
  - Voltage
  - Clock frequency
- Having dedicated co-processors for computation-intensive tasks will, in general, reduce power.
- Careful choice of sleep policy contributes to power reduction (wake-up time is relevant).

ASYNCHRONOUS DESIGN

- The ultimate low-power implementation (given a technology).
- No clocks, no losses in clock trees, etc.
- Data stability is encoded in the data itself rather than guaranteed by clock edge.
- *Handshaking* is used to ensure that data transports are successful.
- There is only activity that is functionally meaningful.
- Computation speed adapts to requirements of the moment.
- Commercial designs exists, such as ARM processors.
- As well as design tool vendors: e.g. *Handshake Solutions* ☹.

FURTHER READING (1)

- Review papers:
  - Pedram, M., *Power Minimization in IC Design: Principles and Applications,* ACM Transactions on Design Automation of Electronic Systems, Vol.1(1), (January 1996).
  - Pedram, M. and A. Abdollahi, *Low-Power RT-Level Synthesis Techniques: A Tutorial,* IEE Proceedings on Computers and Digital Techniques, Vol.152(3), pp. 333-343, (May 2005).
  - Benini, L., G. De Micheli and E. Macii, *Designing Low-Power Circuits: Practical Recipes,* IEEE Circuits and Systems Magazine, Vol.1(1), pp. 6-25, (2001).

FURTHER READING (2)

- Some books:
  - Chandrakasan, A.P. and R.W. Brodersen, *Low Power Digital CMOS Design,* Kluwer Academic Publishers, Boston, (1995).
  - Rabaey, J.M. and M. Pedram (Eds.), *Low Power Design Methodologies,* Kluwer Academic Publishers, Boston, (1995).
  - Benini, L. and G. De Micheli, *Dynamic Power Management, Design Techniques and CAD Tools*, Kluwer Academic Publishers, Boston, (1998).
  - Piguet, C. (Ed.), *Low-Power Electronics Design,* CRC Press, Boca Raton, (2005).

FURTHER READING (3)

- Some more books:
  - Piguet, C. (Ed.), *Low-Power Processors and Systems on Chips*, Taylor and Francis, Boca Raton, (2006).
  - Otis, B. and J. Rabaey, *Ultra-Low Power Wireless Technologies for Sensor Networks*, Springer, (2007).
  - Mohanty, S.P., N. Ranganathan, E. Kougianos and P. Patra, *Low-Power High-Level Synthesis for Nanoscale CMOS Circuits*, Springer, (2008).

FURTHER READING (4)

- A recent book by an expert in the field (on-line copy available through the UT Library):
  - Rabaey, J., *Low Power Design Essentials*, Springer, (2009).
  - http://www.springerlink.com/content/978-0-387-71712-8

- You are welcome to search in my database:
  - http://www.bibix.nl/literature-db/search.html