Abstract

This tutorial covers the circuit fundamentals of CMOS circuits which contribute to the consumption of energy in portable products, as well as guidelines for the design of systems in order to reduce energy consumption and prolong battery life. Circuit fundamentals will include a definition of terms, basic circuit elements, laws of operation, and basic circuit theory applying energy consumption. We will then present three major principles of energy reduction: reducing number of transitions, reducing the amount of switched capacitance, and reducing the operating voltage. Several guidelines that can be applied during the system design process which utilize the three major principles.

1. Introduction

Embedded system containing portable devices

Typically, consists of several portable devices each providing its user with local computational capability. The devices are capable of communicating with each other through RF links to an intermediate communication network or directly through point-to-point RF or IR links.

Design objectives

1. Reduce energy consumption of portable devices

2. Reduce avg. power consumption of stationary devices

1 and 2 may conflict since, given two systems where the first consumes less energy but in a shorter period of time the first may have a higher average power consumption

This tutorial will focus on design objective 1

2. MOS digital circuit fundamentals

Definition of terms

\[ R = \text{resistance} \quad C = \text{capacitance} \]
\[ I = \text{current} \quad V = \text{voltage} \]
\[ P = \text{power} = I \times V \quad E = \text{energy} = \int P \, dt \]

NMOS transistor operation

The NMOS transistor acts as a switch. When the gate-source voltage is less than a voltage, \( V_{th} \), the transistor has a high resistance between its drain and source, as in Fig. 1(a). When the gate-source voltage is greater than \( V_{th} \) the transistor has a low resistance between its drain and source, as in Fig. 1(b).

![Figure 1](image-url)
Sources of energy consumption

The CMOS inverter, as shown at the top of Fig. 2, will be used to illustrate the primary sources of energy consumption in a typical CMOS logic gate.

1. Capacitive load current

   ![Figure 2.](image)

   Occurs when the output voltage of the inverter changes from a low voltage to a high voltage.

   When the output is low, resistance from the output to \( V_{dd} \) is high and resistance to ground is low, causing the load capacitance to be discharged.

   As the resistance from the output to \( V_{dd} \) goes low and resistance to ground goes high, due to a change in the input voltage, the output voltage goes high and the output capacitance is charged.

   \[ E = 0.5 \cdot C \cdot V_{dd}^2 \] per transition.

   Primary source of energy consumption in CMOS.

2. Short circuit current

   ![Figure 3.](image)

   Occurs when output voltage is between high and low values.

   Resistance to \( V_{dd} \) and ground is moderate setting up a conductive path between them.

   \[ E = ((V_{dd} - (V_{in} \cdot V_{tp}) \cdot I_{max} \cdot (T_{rise} + T_{fall})) / 2) \] per transition.

   Typically 20% of total energy consumption.

3. Static current

   ![Figure 4.](image)

   Occurs when output voltage is high or low.

   Finite resistance between \( V_{dd} \) and ground results in a small, but non-zero current between them.

   \[ E = \int_0^t \frac{V_{dd}^2}{R_{l} \cdot R_{d}} \cdot dt \]

   Usually significant only in devices inactive for large periods of time.
Principles of energy consumption reduction

To a first approximation $E = 0.5 \cdot C \cdot V_{dd}^2 \cdot N$, where $N = \# $ transitions

Three principles:

1. *Reduce operating voltage*
2. *Reduce capacitance being switched*
3. *Reduce number of transitions*

Important: It's the number of transitions that counts and not the frequency with which they occur

To achieve significant reductions in each of these three factors

a) The goal of low energy must be part of the design process

b) Low energy design must be approached at all levels of design, including the system, algorithm and device architecture levels

3. Design process

**Guideline 1: Make energy a design constraint**

1. Maintain an energy budget - set energy targets for parts of the design from the start

2. Assign responsibility for meeting energy goals

3. Track progress through use of power estimation tools

**Spreadsheets - advantages and disadvantages**

+ Easy to generate early in design cycle with minimal design information

- Can be difficult to update as design changes

- Prone to error as spreadsheet model not linked directly to design tools and database

- Prone to error as assumptions made concerning component activity not linked to profiling information

**Architectural Power Analysis Tools [1] - advantages and disadvantages**

+ Solves many of the drawbacks of spreadsheets

- Not quite soup yet, no commercially available tools

4. Review progress through regularly scheduled design reviews

4. System level

Network communication often a significant source of portable device energy consumption

**Guideline 2: Use compression**

Reduces number of “transitions” on the network

Many compression techniques exist - designer must select one(s) which match network communication characteristics and application

Important network and application characteristics:

- Real time vs. non-real time - in real time systems compression/decompression typically must be simpler so that they can be performed in less time

- Ability to tolerate loss of data - systems that can tolerate loss of data can use lossy compression schemes which achieve higher compression rates

- Packetized vs. non-packetized - in packetized networks, if compression is to be done on a per packet basis, packet size will influence the relative effectiveness of compression schemes

**Guideline 3: Use unbalanced communication**

Shift energy use to parts of system where it is not a concern

Example: image compression [2]

Consider a system where the image originates and is compressed in a portion of the system where
energy consumption is not a concern, such as the network, and then transmitted to and decompressed on a portable device.

Computational requirements of two methods for image compression are shown below.

<table>
<thead>
<tr>
<th>Method</th>
<th>Computation Reqs/pixel</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>compression</td>
<td>decompression</td>
</tr>
<tr>
<td>Transform (DCT)</td>
<td>4 multiply</td>
<td>4 multiply</td>
</tr>
<tr>
<td>(row-col. fast algorithm)</td>
<td>8 additions</td>
<td>8 additions</td>
</tr>
<tr>
<td></td>
<td>6 mem. access</td>
<td>6 mem. access</td>
</tr>
<tr>
<td>Vector Quant. (Tree search - differential CB)</td>
<td>8 multiply</td>
<td>1 mem. access</td>
</tr>
<tr>
<td></td>
<td>8 additions</td>
<td></td>
</tr>
<tr>
<td></td>
<td>8 mem. access</td>
<td></td>
</tr>
</tbody>
</table>

Vector quantization requires only one memory operation per pixel to decompress vs. four multiplies, eight additions and six memory operations per pixel for DCT. However, compression is more energy consumptive for vector quantization than for DCT. Thus, vector quantization saves energy over DCT in the portable device at a cost in energy in the network.

5. Algorithm level

Surprisingly, many guidelines at algorithm and device architecture level are similar to ones for achieving high performance.

Guideline 4: *Exploit parallelism*

Use parallelism to maintain throughput while reducing clock rate and operating voltage [3]

Example: execution path parallelism

Consider the execution path of a typical processor as shown in Fig. 5 (sans GP register bank)

\[ E_{\text{parallel}} = 2C_{\text{eff}}(V/2)^2 (f/2)t = 0.25E_{\text{original}} \]

Figure 5.

The input latches are clocked with frequency \( f \) so that \( f \) operations/sec are performed.

Now consider an execution path with two ALU’s, each of which is clocked at half the rate of the original, as in Fig. 6.

\[ E_{\text{parallel}} = 2C_{\text{eff}}(V/2)^2 (f/2)t = 0.25E_{\text{original}} \]

Figure 6.

Reduction in clock speed, due to parallelism, allows operating voltage to be reduced by an approximately proportional amount.

Reduction in operating voltage reduces energy consumption proportional to \( V^2 \), and in this example by a factor of 4.

Energy consumption reduction depends on ability of the software to exploit the parallelism in the hardware. Greater parallelism in algorithm allows for greater utilization of parallelism in hardware.

Significant gains possible - research shows a reduction of up to 7 times demonstrated for simple circuits.

Guideline 5: *Choose an energy efficient algorithm*
Energy to perform a given function depends upon the algorithm chosen to implement the function.

Example: cascade vs. parallel realization of a discrete time system [4]

Consider the discrete time system, whose transfer function, \( H(z) \), is:

\[
H(z) = \frac{3z^2 + 3.6z + 0.6}{z^2 + 0.1z - 0.2}
\]

The flow diagram of the “cascade” realization of this system is given in Fig. 7. It requires 4 multiplies, 4 additions and 5 memory accesses to implement.

![Figure 7](image)

The flow diagram of the “parallel” realization of this system is given in Fig. 8. It requires 5 multiplies, 4 additions and 5 memory accesses to implement.

![Figure 8](image)

The “cascade” form in this case uses 20% fewer multiplies.

Potentially large difference for some algorithms

Efficiency vs. parallelism - more parallel algorithm may require more operations but use less energy if it allows system to run at a lower operating voltage

**Guideline 6: Apply optimizing transformations**

Improve energy efficiency of an algorithm by reordering operations, substituting for energy consumptive operations, changing where variables are stored, etc.

Example: product of a complex variable by a complex constant

![Figure 9](image)

In Fig. 9(a), the real and imaginary terms are generated independently.

In Fig. 9(b), the real and imaginary terms of the constant are combined and used to eliminate one multiplication at the expense of one addition.

Since additions tend to consume less energy than multiplies, this transformation saves energy.

Similar to compiler optimizations for performance.

Efficiency vs. parallelism - same trade-off as for Guideline 5, since (a) is more parallel than (b) but uses less energy consumptive operations.

### 6. Device architecture

**Guideline 6: Match hardware to algorithm**
Processor - Choose processor with instruction set, data path width, functional units suitable to algorithm.

16-bit RISC reduces memory traffic by ~25% over 32-bit RISC, with comparable performance, for limited memory systems [5].

Least energy consumptive DSP processor varies according to the application [6], as shown in the table below:

**Energy per Function** (in W)

<table>
<thead>
<tr>
<th>Function</th>
<th>ADI ADSP 2103</th>
<th>Motorola DSP56L002</th>
<th>TI TMS320LC 51</th>
</tr>
</thead>
<tbody>
<tr>
<td>Real FIR</td>
<td>6.32</td>
<td>7.86</td>
<td><strong>4.86</strong></td>
</tr>
<tr>
<td>IIR</td>
<td><strong>.35</strong></td>
<td>3.6</td>
<td>.51</td>
</tr>
<tr>
<td>Vect. Max</td>
<td>.88</td>
<td>.74</td>
<td>.95</td>
</tr>
<tr>
<td>256 Pt FFT</td>
<td>89.08</td>
<td><strong>70.22</strong></td>
<td>113.83</td>
</tr>
</tbody>
</table>

Support logic - Implement often used functions in hardware.

Often functions are less energy consumptive when performed on special-purpose hardware than when performed in software on general-purpose hardware.

Thus, the addition of hardware timers, special-purpose coprocessors/functional units tend to reduce energy consumption.

**Guideline 7: Localize communication**

Map functions in algorithm to hardware such that interchip communication is minimized.

May result in a trade-off vs. maximal exploitation of parallelism inherent in algorithm.

Use memory hierarchy [7].

Use distributed vs. centralized resources [2].

Example: motion compensation.

Uniprocessor

Single high speed ALU

4ns memory

Supply voltage 5V

\[ f_{\text{clock}} = 250Mhz \]

\[ E = 88.2e-6J = 0.5 \cdot C_{\text{uni}} \cdot 5^2 \cdot 2.5e + 8 \cdot 1.26e - 5 \]

Multiprocessor

48 concurrent processors

Supply voltage = 1.1V

\[ f_{\text{clock}} = 1Mhz \]

\[ E = 60e-9J = 0.5 \cdot C_{\text{multi}} \cdot 1.1^2 \cdot 1e + 6 \cdot 1.2e - 5 \]

\[ C_{\text{multi}} = .27 \cdot C_{\text{uni}} \]

Some reduction in the effective capacitance due to operations being performed on smaller processors, but remaining reduction due to lack of communication over highly capacitive global busses.

**Guideline 8: Reduce resource sharing**

Resource sharing tends to decrease signal correlation and hence increase switching activity [2].

Example: dual counters.

Two counters in Fig. 10(a) are multiplexed onto a bus in alternate cycles. Successive values for each counter are highly correlated, i.e. they differ in only two bit positions on average. However, successive values appearing on shared bus differ significantly depending upon skew between counts.

Non-shared configuration of Fig. 10(b) keeps the output data streams of the two counters from interfering with each other.
Other examples include: multiplexed data busses within an execution unit and multiplexed external address and data busses occurring on some processors. Does not always increase switching activity - depends on autocorrelation of each data stream vs. correlation between data streams when sharing resources.

**Guideline 9: Partition design**

Partition design into subcircuits whose power level can be independently controlled.

Power down subcircuits when not in use.

This allows the system to keep its energy consumption closer to the minimum level necessary to perform the function at hand.

Higher resolution partitions tend to yield greater energy savings but savings are eventually offset by energy needed to operate circuitry controlling power level of sub-circuits.

7. Conclusion

Energy consumption is proportional to $C \cdot V^2 \cdot N$

Energy consumption reduction must be a part of the design process and addressed at all levels of design.

Careful choice of algorithm, hardware, and mapping of algorithm to hardware can yield orders of magnitude reduction.

J. Rabaey, R. Brodersen, et al. at UCB have designed and built a portable multimedia terminal with total power consumption of 50mW, excluding power of display - order of magnitude less than comparable designs.

Once system design is complete, need to continue to address energy consumption at the logic and physical design level.

8. References


