Krzysztof Iniewski
*Editor*

**ACSP**
Analog Circuits And Signal Processing

# CMOS Processors and Memories

Springer

# CMOS Processors and Memories

**ANALOG CIRCUITS AND SIGNAL PROCESSING**

*Consulting Editor*: **Mohammed Ismail**, *Ohio State University*

For other titles published in this series, go to
www.springer.com/series/7381

Krzysztof (Kris) Iniewski
Editor

# CMOS Processors and Memories

Springer

*Editor*
Krzysztof (Kris) Iniewski
CMOS Emerging Technologies, Inc.
Stanley Place 2865
V3B 7L7 Coquitlam British Columbia
Canada
iniewski@yahoo.ca

ISBN 978-90-481-9215-1 e-ISBN 978-90-481-9216-8
DOI 10.1007/978-90-481-9216-8
Springer Dordrecht Heidelberg London New York

Library of Congress Control Number: 2010933360

# Contents

# Part I
# Processors

# Chapter 1
# Design of High Performance Low Power Microprocessors

**Umesh Gajanan Nawathe**

**Abstract** The field of Microprocessor design came into existence in the early 1970s with the first microprocessor from Intel (the 4004). Since then, the technology and complexity of Microprocessors has increased exponentially following Moore's Law, which implies that the complexity of integrated circuits doubles every 2 years. The Intel 4004 had a little more than 2,000 transistors. Some of today's microprocessors have more than two billion. Early on, microprocessor design philosophy focused on increasing performance without worrying about how it would affect power consumption. Today, power consumption has become a major design constraint. As a result, power-efficient design became a priority and 'power management' came into existence. Today, it is no longer enough to only reduce maximum power - it is equally important to reduce idle power and also have the ability to manage power and be able to operate at various performance-power points depending upon the customer's needs/choosing. Modern Semiconductor technologies have become very complex. Transistor Performance and Power is increasingly dependent upon its layout and also the layout that surrounds it. Variation of key Transistor parameters has increased and hence statistical analysis has become very important.

**Keywords** Microprocessor • Processor • CMT • Concurrent Multi-Threading • SPARC • Niagara • Power • Static Power • Leakage Power • Gate Leakage • Subthreshold leakage • Drain Leakage • Diode Leakage • Dynamic Power • Power-efficient design • Power Management • Dynamic Voltage and Frequency Scaling • Dynamic Frequency Scaling • DVFS • Leakage, Back-Bias • Clock Power • Clock Design • Clock Skew • Clock Uncertainty • Systematic Skew • Layout Dependent effects • SRAM design • Memory Design • SRAM redundancy • 6-T Memory Cell • Statistical Analysis

U.G. Nawathe (✉)
Oracle Corporation, 4110 Network Circle, Santa Clara, CA 95054, USA
e-mail: unawathe@yahoo.com

K. Iniewski (ed.), *CMOS Processors and Memories*,
Analog Circuits and Signal Processing, DOI 10.1007/978-90-481-9216-8_1,

## 1.1 Introduction

The field of Microprocessor design came into existence in the early 1970s with the first microprocessor from Intel (the 4004). Since then, the technology and complexity of Microprocessors has increased exponentially following Moore's Law, which implies that the complexity of integrated circuits doubles every 2 years. Apart from Intel, early processor designs came from Zilog(Z80), Motoroal (M6800/68000), and Texas Instruments (TMS1000). Over time, Intel's x86 based architecture emerged as the dominant architecture, especially for the Personal Computer (PC) industry. Advanced Micro Devices (AMD) was the other significant player designing processors on the same architecture, referred to as the x86 architecture. For more powerful higher end systems, there were computer architectures from other companies which gave Intel a run for their money. Chief among them were SUN Microsystems' SPARC architecture, MIPS Computer system's MIPS, and IBM's PowerPC and Power architectures. Early implementations of a lot of these architecture were based on RISC (Reduced Instruction Set Computing) as opposed to Intel's CISC (Complex Instruction Set computing). There were also a debate between Super-scalar vs. Super-pipelined approaches to designing processor. All these approaches were focused on improving performance at any cost without paying much attention to power dissipation. This mindset continued until a few years ago. The birth of the internet and its subsequent explosive growth caused the number of computer systems in a datacenter and the number of datacenters themselves to increase dramatically. Datacenters started running out of capacity because datacenter cooling limits were reached. In effect, datacenters started hitting the proverbial power wall. Thus, power-efficient design – not only at the processor/chip level, but also at the system level – became very important. The computer industry started looking into new ways of doing power efficient designs while at the same time maintaining/increasing performance and also ways to manage power consumption in real world scenarios. As a result, the 'Concurrent Multi-Threading' (CMT) approach of doing processor/system design came into existence [2, 3].

## 1.2 Concurrent Multi-threading (CMT)

The basic concept of CMT is to have the processor support the concurrent execution of multiple threads. A lot of high performance power-efficient processor designs today use the CMT architecture approach. CMT also helped reduce the impact of another bottleneck – memory latency. Figure 1.1 shows how memory speed (i.e. latency) has changed over time compared to processor frequency. Figure 1.2 illustrates how the CMT architecture attempts to reduce the impact of this issue. For a single thread, memory access is the single biggest bottleneck to improving performance. For programs that exhibit poor memory locality, only a modest throughput speedup is possible by reducing compute time. As a result, processors which are optimized for Instruction-Level-Parallelism have low utilization and wasted power. Having many threads makes it easier to find something

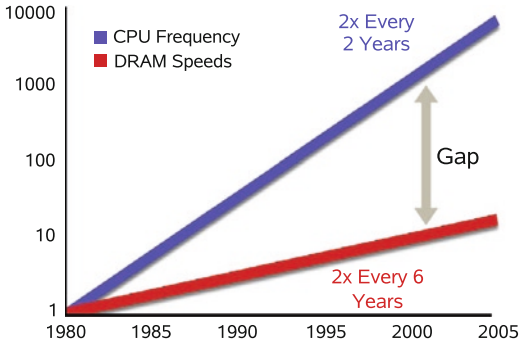


**Fig. 1.1** Relative CPU frequency and memory speeds over time

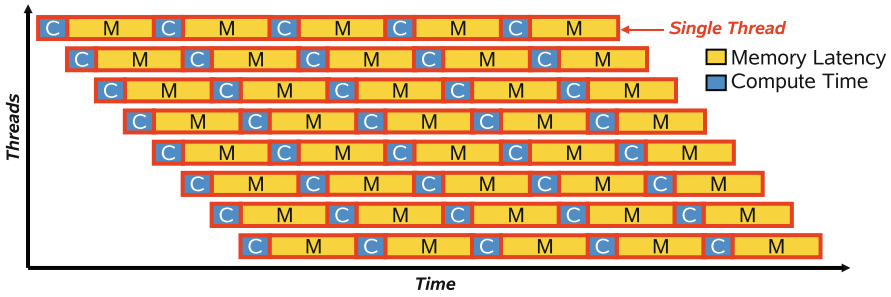


**Fig. 1.2** Throughput computing using the CMT architecture (copyright © IEEE 2007)

useful to execute every cycle. As a result, processor utilization is higher, significant throughput speedups are achievable, and wasted power is reduced by reducing speculative execution (the result of which could get discarded).

## 1.3 Power and Power Management

As we discussed earlier, Processor/Chip and System Power consumption have become big design care-abouts. Hence it is important that we discuss Processor Power Consumption and Power Management in sufficient detail.

Total Chip power can be thought of as having two components: Dynamic Power and Static (Leakage) Power. Let's first focus on Dynamic Power.

### *1.3.1 Dynamic Power*

Dynamic Power can be summarized using the following equation:

$$\text{Dynamic\_power} = \text{aCV}^2 f + \text{P\_crowbar},$$

where

a = Activity factor
C = Switching capacitance
V = Voltage (Vdd)
f = Frequency of operation.

#### 1.3.1.1 Activity Factor and Switching Capacitance

Clock power is a big portion of the dynamic power. For one, the clock signal can switch twice every cycle (once low to high and once high to low). Also, the clock signal is integral to designing synchronous systems and is all-pervasive – every flip-flop or latch on the chip needs it. Reducing the activity factor will result in a linear reduction in power. Most, if not all, processors today employ clock-gating to achieve this. Conceptually a set of flip-flops that are meant to implement a specific architectural feature are grouped together and clock to these flip-flops is turned off during cycles when the outputs of these flip-flops don't hold any useful data and/or the flip-flops don't serve any useful purpose. The way this is implemented is that the clock to these flip-flops is driven by the same clock buffer or sets of clock buffers and its output is turned off using a 'Clock Enable' signal that is generated from logic signals. Even if one flip-flop in this group of flip-flops is performing useful work, the clock to all the flip-flops will have to toggle. Thus, the way flip-flops are grouped is important because it is important to find as may clock cycles as possible during which none of the flip-flops in the group is performing useful computation. Note that you are consuming extra power to generate the 'Clock Enable' signal. Obviously the grouping has to be done in such a way that the power saved by gating off clocks is more than the extra power needed to generate the 'Clock Enable' signals. Having an enable per flip-flop ends up costing too much power and having an enable per, say 1,000 flip-flops, generally ends up being not useful enough since the number of cycles when all the 1,000 flip-flops are not processing valid data is likely very small. Some studies have shown that having in the range of, say, 16 flip-flops per group is a reasonable number. In general, the above numbers are higher for data-paths than for control blocks containing random logic, since datapaths could have wide words (e.g. 128 bits wide) and all 128 flops in a row can generally be controlled by the same 'Clock Enable' signal. Note that the 'Enable' needs to satisfy timing constraints, i.e. it has to be evaluated every clock cycle and has to satisfy setup time constraints to the clock gate in the clock buffer.

Both Max Power and Idle Power are key distinct metrics that drive design decisions in modern processors. Idle power is especially important for processors used in laptops and other mobile devices to conserve battery charge. When the system is in the idle state, there are a lot more flip-flops and a lot more clock cycles when the flip-flops do not hold any useful data or don't do any useful work and hence lot more power savings can be achieved by clock gating when the system is idle.

There are several levels of clock gating implemented in modern processors. There is the 'flip-flop-group' level clock gating that we already discussed. There is also the 'functional-unit' level clock gating. For example, the clock to a 'floating point unit' can be turned off during cycles when the 'floating point unit' is not processing valid instructions. There is also 'chip-unit' level clock gating. For example, in today's multi-core processors, clocks to certain processor cores can be turned off when that processor core is not in use.

Other contributors to power are switching capacitances associated with routing wires, gates, and diffusions of transistors. Generally it is better to optimize wire classes (widths and spacings of wires) used for routing for minimizing 'power-delay product' as opposed to only 'delay' because if a designer does the latter, after a certain point he/she spends a lot on increased power consumption for a very small gain in performance – in effect the law of diminishing returns. Semiconductor technology companies are also making advances and making the transition to using 'lower-k' dielectrics which help reduce routing capacitance (in addition to reducing delay) and hence interconnect power.

Gate and diffusion capacitance of transistors also contribute significantly to power. It is always prudent to minimize sizes of gates in non-critical paths to minimize this power. As we will discuss later, this reduces leakage power as well, which is very important since, typically leakage power is a big (20–30%) portion of the total power in high performance processors.

Technology has a very important role to play in minimizing capacitance. Typically a linear shrink factor of 0.7–0.8 is achieved when a design in one technology node is ported to the next technology node (e.g. going from the 90 nm to the 65 nm technology node, or going from the 65 nm to the 45 nm technology node).

#### 1.3.1.2 Voltage (VDD) and Frequency of Operation

Chip VDD has a big impact on power. As the equation at the beginning of Section 1.3.1 shows, dynamic power is directly proportional to the square of the voltage. As we will discuss later, Chip VDD also has a stronger impact on leakage power. In contrast, performance increases approximately linearly as VDD is increased. So, minimizing VDD is very important in order to reduce power consumption. Most processors today employ multiple voltages on chip. SRAM cell functionality is a big obstruction to reducing voltage because for voltages below what is called 'Vmin' of SRAM cells, the SRAM cells do not function reliably. As a result, quite often SRAM cells are put on a separate voltage supply (different from the rest of the chip) and the VDD to the rest of the chip is determined based on the chips performance/power requirements and tradeoffs. Very often, different parts of the logic on the chip is put on different supply voltages. For example, the high frequency processor cores are put on a higher supply voltage and the relatively lower frequency System on Chip (SOC) blocks are kept on a lower supply voltage. As we will discuss later, a lot of modern processors employ a technique called Dynamic Voltage and Frequency Scaling (DVFS). This means that when

the processor is idle, its Frequency of operation and hence its VDD can be reduced to obtain a close to cubic power savings.

#### 1.3.1.3 Crowbar Power

Crowbar Power is also a non-negligible component of dynamic power. This is really wasted power. When a logic gate switches, as the input transitions from 0V (VSS) to VDD or vice versa, there is a small amount of time when the N-transistors and P-transistors in a CMOS gate are conducting at the same time. During this time, some current flows directly from VDD to VSS. This current/power is wasted. The amount of this power strongly depends on the input and output slew rates. The larger the input slew rate, the longer the time when both N- and P-transistors are on and larger the crowbar current. The slew rate at the output of a gate also has an impact on the crowbar current because it determines the 'VDS' of the N- and P-transistors during the time when both of them are turned on. Processor physical designers always have limits on the maximum slew rates that signals on chips can have to limit crowbar power. Incidentally, other reasons why slew rate limits are in place are to limit age-related transistor degradation due to the Channel Hot Carrier (CHC) effect, improve signal noise immunity, and make sure the signals have a full transition between the power supply rails at the highest frequency of operation.

### *1.3.2 Static (Leakage) Power*

As was previously stated, leakage power in processors today is typically 20–30% of the total power. Leakage power consists of sub-threshold leakage, gate leakage, and diode leakage.

#### 1.3.2.1 Sub-Threshold Leakage

Typically, sub-threshold leakage is the largest component. As the name suggests this is the leakage power due to current flowing from the transistor's drain to source when the transistor is in the off state (VGS < Vth). For a CMOS inverter, when its input is at VSS, its output is at VDD and the leakage current is determined by the N-transistor leakage current. Similarly, when the inverter's input is at VDD, its output is at VSS and the leakage current is determined by the P-transistor leakage current. Leakage current of multi-stack gates, like nand2 and nor2, is lower than that of the inverter because there are two or more transistors in series.

Sub-threshold leakage is highly dependent upon the transistor channel length and threshold voltage of the transistor. Transistors with minimum channel length and lowest threshold voltage have the largest sub-threshold leakage. One of the techniques used to control sub-threshold leakage is to use longer channel length transistors. This is also very useful from a statistical point of view. The threshold

voltage variation of longer channel length transistors is significantly smaller than minimum-channel-length transistors. Hence, statistically for a large group of transistors, leakage is significantly lower if the channel length is increased. It is especially important since leakage increases exponentially as threshold voltage reduces.

Typically, every process technology offers two to four kinds of core (i.e. Non-IO) transistors having different threshold voltages. The HVT (highest threshold voltage) transistors from this menu of transistors typically have 10–20% of the leakage of standard Vt transistors (at the cost of having 30–50% higher delay). These transistors can be used in logic paths where the number of levels of logic is small. Additionally, typically design teams have available to them a library of gates which use longer channel length transistors (sometimes they are referred to as GBIAS transistors since their gate lengths are 'biased' higher). These gates are typically designed to be footprint compatible with the corresponding cells from the library which use minimum channel length. Generally, the gate 'bias' is chosen so that the delay penalty is not as high as HVT devices. Consequently, the leakage reduction is not as much. Typically, 50% leakage reduction at a 15% delay penalty is targeted, though these numbers could vary based on design choices and semiconductor technology. Gates which are not in critical paths are substituted with these GBIAS gates to reduce leakage.

#### 1.3.2.2 Gate Leakage

Several years back, gate leakage was fairly small compared to sub-threshold leakage. As semiconductor processing technology shrank from one generation to next, the gate oxide became thinner and thinner. Gate oxide thicknesses started approaching 10s of Angstroms, and gate leakage started increasing exponentially. Fortunately, the newest generations of process technology (45 nm and beyond) started employing metal gate technology, which effectively got rid of the gate leakage problem by reducing it by orders of magnitude. This can be seen from the four graphs in Figs. 1.3 and 1.4. These figures show sample results for a non-metal-gate and metal-gate process respectively. In these graphs, the magnitude of drain and gate leakage is shown relative to gate leakage at the lowest voltage/temperature point of the respective graph. As the graphs indicate, the ratio of gate to drain leakage current is lower in the metal-gate process (vs. a non-metal-gate process) by almost two orders of magnitude.

#### 1.3.2.3 Diode Leakage

Diode leakage is the reverse biased leakage current flowing through the reverse biased diode formed by the source or drain of the transistor and the bulk or the well that the transistor resides in. Under normal bias circumstances, this current is very small. As we will see shortly, one of the techniques used for controlling sub-threshold leakage is to reverse bias the bulk (also called back bias) and this leads to an increase in diode leakage current.

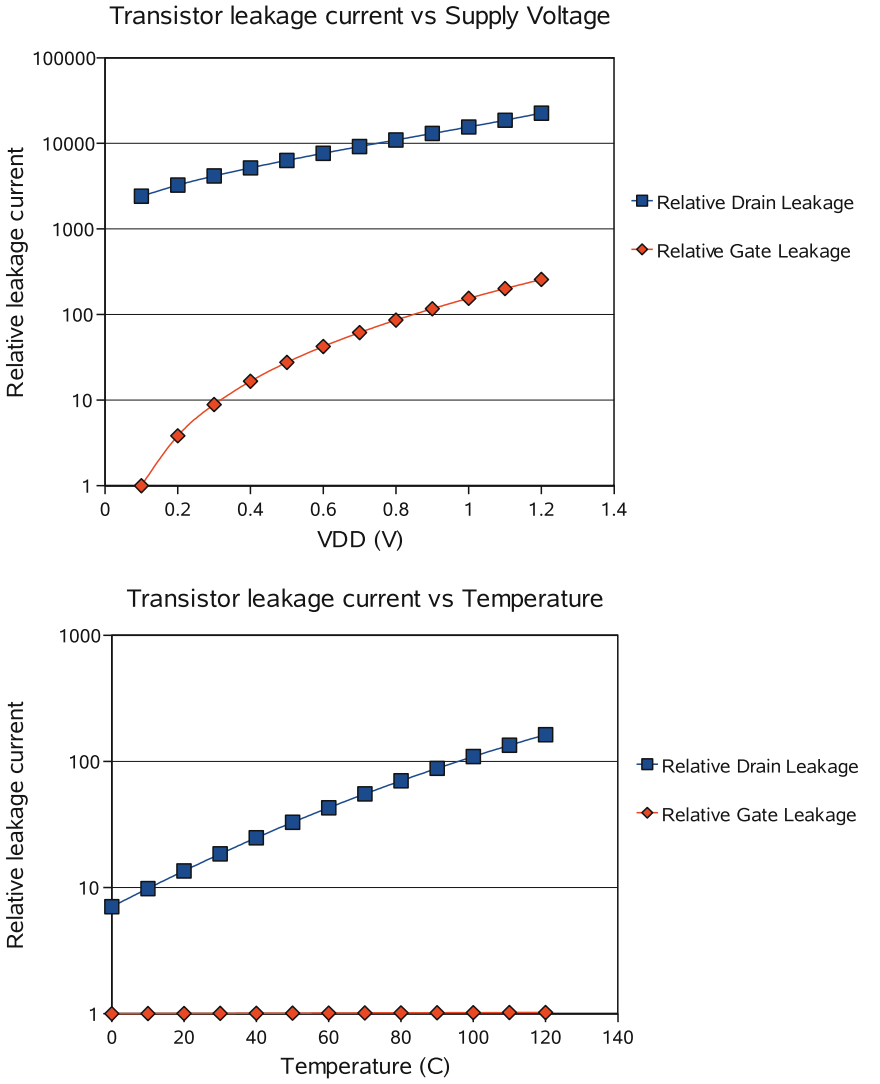


**Fig. 1.3** Dependence of leakage on supply voltage and temperature for a non-metal-gate semiconductor process

### 1.3.2.4 Effect of VDD and Temperature on Leakage

One of the big overlaying factors that affects leakage is VDD. Leakage power has a larger than cubic relationship to VDD. Hence, in modern power-conscious processors, an attempt is made to put significant portions of the chip on lower supply voltages. That is also the reason why an attempt is made to turn off the power supply to cores or significant sections of the chip when they are not in use.

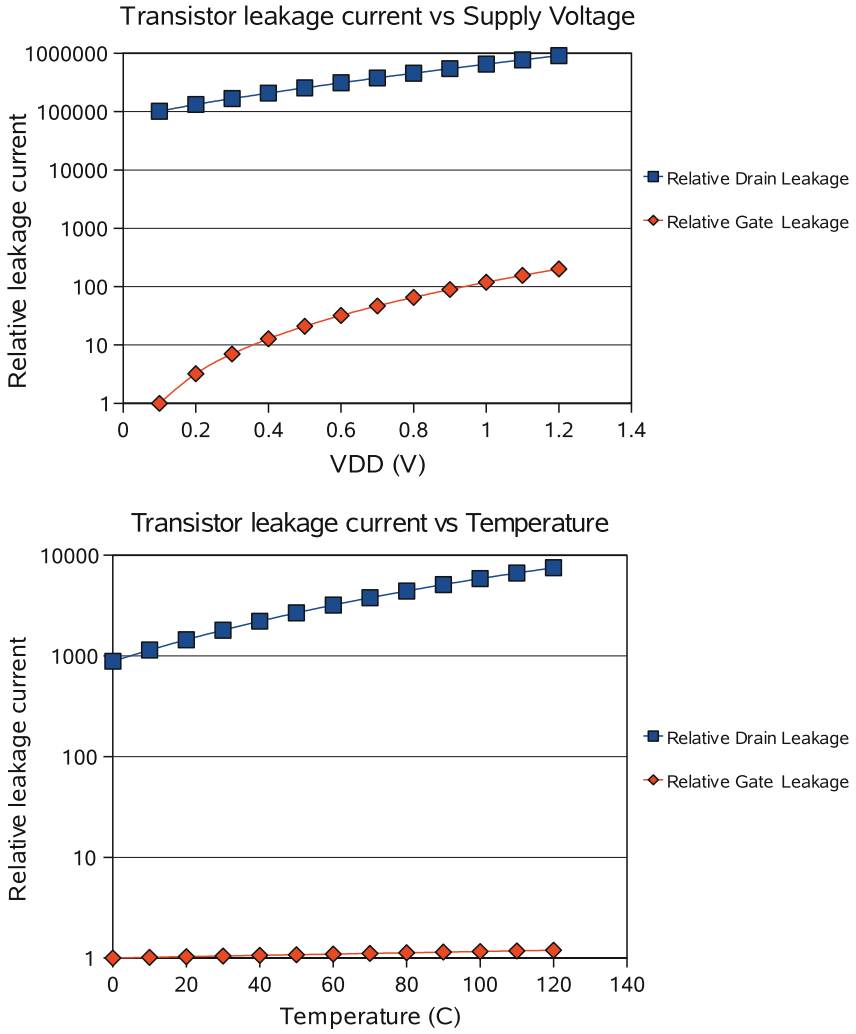


**Fig. 1.4** Dependence of leakage on supply voltage and temperature for a metal-gate semiconductor process

Temperature also has a large effect on leakage. Gate leakage is relatively independent of temperature, but sub-threshold leakage and diode leakage increase as temperature increases. Hence, design of system cooling solutions is very important. Most systems are air cooled using a set of fans that blow air over chips in a system. Typical systems have heat sinks sitting on top of processors. These heat sinks are designed such that they have a large amount of surface area in contact with air and this helps heat escape faster and ultimately helps lower the maximum junction temperature. Under certain circumstances, it might be worth investing in a better cooling solution, e.g. liquid cooling. However the cost of the cooling solution has

to be taken into account in the context of the type of system that is being designed. Typically, only the high end systems are able to afford advanced higher-cost cooling solutions.

Figures 1.3 and 1.4 show how much drain leakage (which is a big portion of the total leakage) changes with Temperature and Supply Voltage for a typical non-metal-gate and metal-gate process.

#### 1.3.2.5 Back Bias

Back bias is a commonly used technique to control leakage power. The principle behind back bias is the effect that body bias has on the threshold voltage of the transistor. The equation of threshold voltage can be written as

$$\mathrm{Vth} = \mathrm{Vth0} +/- \gamma * \left(\left| \mathrm{VSB} \right|\right)^{1/2}$$

Where, the +ve sign is used for N-transistors and −ve sign is used for P-transistors. Vth0 is the threshold voltage of the transistor when the Source and the Bulk are at the same potential. As you can see, as the magnitude of VSB (voltage between the transistor Sources and the Bulk) increases, the diode formed by the source and the bulk becomes more reverse biased and the magnitude of Vth increases. This in turn causes sub-threshold leakage current to reduce. This is equally applicable to N- and P-transistors. In the case of P-transistors, as VSB (voltage between the transistor Source and the Nwell) becomes more negative, the Source-Nwell diode becomes more reverse biased and the magnitude of Vth increases (Vth becomes more negative). Consequently sub-threshold leakage current decreases. Note that an increase in 'back bias' reduces leakage current, but at the cost of increase in delay. After the 'back bias' is increased beyond a certain point, the reverse bias diode leakage current increases and becomes more dominant At that point, the effectiveness of back bias to reduce leakage current vanishes. Figure 1.5 illustrates the various leakage current components for a modern metal-gate process. In this particular example, you can see that beyond a back bias voltage of about 0.5 V, total leakage current is dominated by diode leakage current and back-bias as a technique to reduce leakage current is no longer useful. Note that, using the same principle, as you apply a 'forward bias' to the Source-Bulk or Source-Nwell junction, Vth of transistors reduces, the sub-threshold leakage current increases, and the transistor delay reduces helping improve performance.

### *1.3.3 Using VDD and Back-Bias to Optimize for Performance and Power*

So, how is this used in the real world? Processors that have been shipping for a few years and that ship today have to meet frequency and power constraints. Due to process variation, a portion of the distribution of parts that comes from the fast

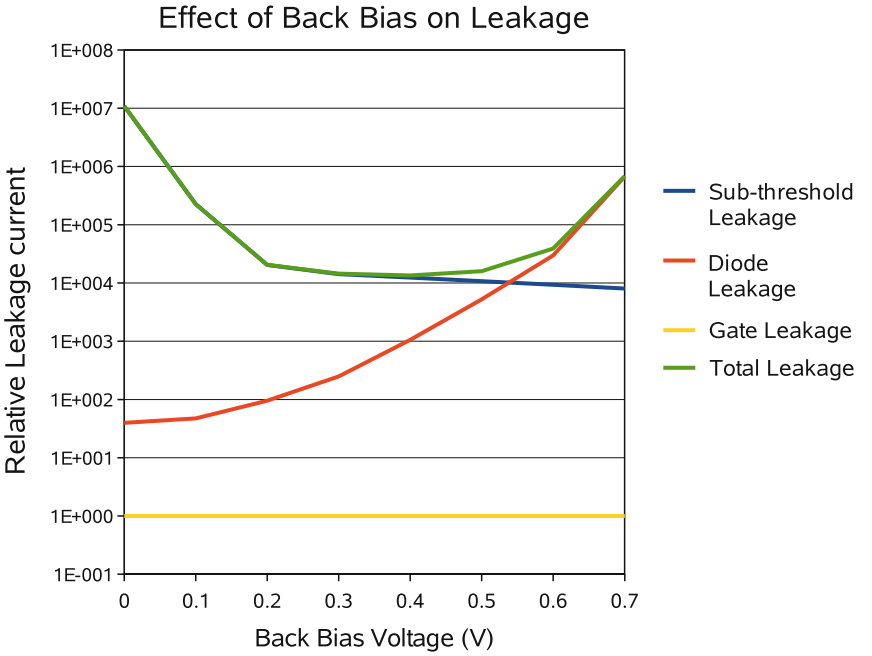


**Fig. 1.5** Effect of back bias on leakage

corner of the transistor process could have more than enough margin in terms of meeting frequency requirements. However, some of these parts may be over the power constraint. For these parts, back bias is an effective technique to trade of frequency margin for a reduction in power. In most processors that are shipped today, the amount of N- and P-back bias is burned into the on-chip fuse lookup table at the time the chip is tested. This information is read by the system firmware and the value of the dc–dc converters on the board that drive the back-bias voltage is set accordingly.

Note that the back bias technique can also be used for the reverse reason for chips that come from that portion of the process distribution which has slower than typical transistors. These chips could be well below the power limit but are not fast enough to meet the frequency target. For these chips, a 'Forward bias' can be applied to trade off power for frequency.

Just as the back-bias voltages can be independently set per chip, the VDD can be independently set per chip to improve the frequency-power yield of a chip. For chips that are from the fast part of the distribution, the VDD is decreased to get their power below the power budget, as long as they still meet the target frequency. Similarly for chips that are from the slow part of the distribution, the VDD is increased to get their frequency above the minimum limit while keeping their power below the limit. As with back bias, the on-chip fuse array is used to implement this technique.

Note that there is a limitation on what the N and P back bias voltages can concurrently be set to. For example, if the N Back bias voltage is increased far more than the P back bias voltage, the P-transistor becomes stronger relative to the N-transistor and the effective 'Beta ratio' of a typical logic gate decreases by a large amount. Similarly, if the P Back bias voltage is increased far more than the N back bias voltage, the N-transistor becomes stronger relative to the P-transistor and the effective 'Beta ratio' increases by a large amount. Obviously, if the design of various circuits on the chip is not robust enough to take this into account, the circuits could fail. For this reason, the circuit design methodology and the design window has to be decided after taking into account the productization strategy of the chip.

### *1.3.4 Power Management: What and How?*

Power Management in the context of a computer system/processor can be defined as the ability to manage the power consumption of the system/processor within a set power constraint/budget and/or opportunistically be able to reduce system/processor power consumption when the right set of circumstances present themselves. Having defined it, let's look at some commonly used techniques that are used to enable/implement Power Management.

#### 1.3.4.1 Dynamic Voltage and Frequency Scaling

One very powerful Power Management technique is called Dynamic Voltage and Frequency Scaling (DVFS). The basic concept of DVFS is the ability to dynamically change the voltage and frequency at which the chip operates depending upon performance requirements and power constraints at any given point in time. For example, when the workload on a processing engine is low or when the processing engine is idle, the frequency and the voltage at which it operates can be lowered to reduce power consumption. Here, the word 'dynamically' means on the fly – i.e. without having to reset the system/processor. The processor and or processing engine will transition to a new frequency and/or voltage as it continues to run whatever code/workload it was running.

A common way of implementing DVFS is to have two different phase locked loops (PLLs) – say PLL1 and PLL2. Let us assume that the system is operating at a particular voltage and frequency using PLL1. Once the decision is made to change the voltage and frequency at which the processor is operating, the PLL2 is locked to the new operating frequency and the processor clock tree is switched to PLL2 using a multiplexer. Note that if the PLL2 frequency is lower than PLL1 frequency, the processor is switched to PLL2 first and then the voltage is lowered to the point at which it can support the new lower frequency. If PLL2 frequency is higher than PLL1 frequency, the voltage is increased to the point that is needed to support the higher frequency first before the clock tree is switched to PLL2.

Typically, DVFS is implemented through OS/software control. The software interacts with control registers on chip to implement the frequency change. In some implementations, it also interfaces with the DC–DC converter on chip to change the voltage at which the processor operates.

There are some processors in the market today, especially from Intel, where there is a dedicated Power Management unit inside the chip. In these chips, typically each processing core is powered by its own separate power domain and in some cases its own PLL, so that their frequency and voltage can be independently controlled. This unit has a lot of intelligence and can decide with little interaction with the OS what frequency and voltage each processing core should be running at.

Computer systems are typically designed for a specified maximum temperature that the transistor junction can be at. Having one or more temperature sensors on chip is vitally important to ensure that this specification is met. The thermal sensor continuously monitors the maximum chip junction temperature and provides this information to the on chip Power Management Unit or the OS/software. If the junction temperature exceeds the set limit, DVFS is used to reduce the voltage and frequency, which reduces power consumption and which results in the junction temperature reducing below the limit. Apart from DVFS, there are other ways of implementing this as well. For example, some systems throttle the chip (i.e. reduce instruction issue rates) to reduce power consumption and achieve the same result. Figure 1.6 illustrates this for Sun Microsystems' Niagara2 Processor [1–3].

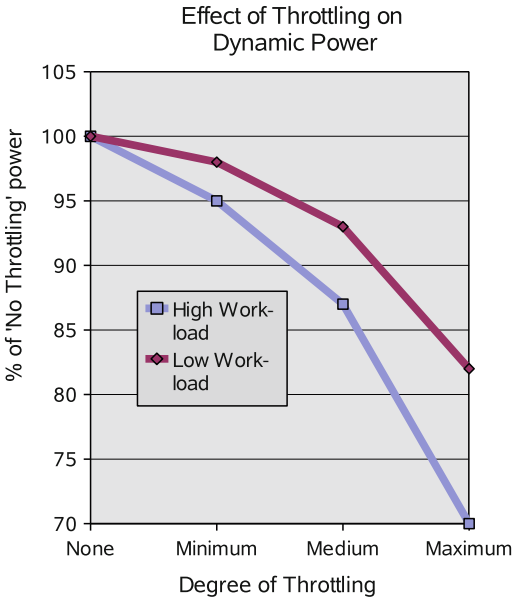


**Fig. 1.6** Power reduction by throttling (copyright © IEEE 2007)

The temperature limit for which the system is designed is called the Thermal Design Point (TDP). Generally, the worst case average power consumed by real applications is about 10–20% lower than that consumed by a 'power virus'. Setting the TDP based on power consumed by a 'power virus' could lead to over-design and increased system cost. Generally, the design of the chip packaging solution, system cooling solution, etc is done in such a way that most (if not all) real applications operating at their worst case power points do not cause the junction temperature to exceed the TDP. In the pathological case when somebody is running a 'power virus', the system will get throttled back to make sure the maximum junction temperature (Tjmax) doesn't exceed the TDP. This way, over-design (and hence) system cost is reduced. Note that in general:

Worst case average power for power virus > TDP > Average CPU Power consumption.

For a multi-core processor, DVFS can also be used to get higher single thread performance than what you would have gotten under normal circumstances. Let's say for example that a processor has four processing cores and the power budget that each core is expected to operate under normal circumstances is 'P'. If there is a piece of code that requires higher single thread performance, three of the four cores can either be turned off or the frequencies that they operate at could be reduced and the power savings achieved by doing this could be transferred as extra power budget to one core which can be run at a higher than nominal frequency and voltage to get increased single thread performance.

#### 1.3.4.2 Other Power Management Techniques

The on-chip circuit performance is dictated by the voltage that the transistors see, which is the Pin VDD minus the Ldi/dt noise, on-chip IR drops, and any other power supply tolerances. Some processors have also incorporated circuits to reduce the board Ldi/dt noise and thus effectively enable reduction of the Pin-VDD (and hence power) for the same performance or increase the processor performance at the same pin-Vdd. Most implementations of this circuit attempt to slow down the chip frequency when it senses an Ldi/dt noise event.

In a computer system, memory power is a big portion of the total power. So, lot of processors provide hooks to help control and manage memory power. For example, on Sun Microsystems' Niagara2 processor, the memory controller is design to enable Memory DIMMs to be powered down. It also has the ability to control Memory access rates to control Memory power consumption. Most processors today have SerDes-based IO interfaces. The number of SerDes lanes in operation can be reduced to reduce SerDes power consumption. Processors today also have several coherency links to support building multi-way glueless systems. For low end systems (e.g. 2-way system vs. an 8-way system), the number of coherency links required is less than the maximum. For these systems, the extra coherency links and the associated logic can be turned off to reduce processor and system power.

## 1.4 Clock Design

It is very important to design a low skew/uncertainty, low power clock distribution network for a high performance low power microprocessor. By its very nature, the quality of the clock distribution network has a truly chip-wide impact. For every peco-second (ps) of clock uncertainty that the clock distribution has (as compared to a reference design), the performance of the rest of the design has to be better by that amount to achieve the same chip frequency. Every addition ps of clock uncertainty will also mean there are some additional min-time paths (also called hold-time paths) that will need to get fixed to avoid functional failure. Also, the clock signal has, by far, the highest activity factor on chip and hence accounts for a large percentage of the total power (could be as much as 25–33% of the total power). Hence it is very important that the clock distribution architecture supports clock gating and other power saving techniques and the clock physical design is done to minimize power. The clock architecture must also be defined to support Design for Test (DFT) and help make chip debug easier.

The kind of clock distribution required is closely tied to the kind of synchronization element used on chip. For example, for a flip-flop based design a single-phase clock is required and for a latch-based design where a logic path can borrow time across the next cycle, a 2-phase clock is needed.

Typically the chip Global Clock Distribution starts at the on-chip PLL, goes to the center of the chip and is then distributed to different parts of the chip using a balanced distribution, like an H-tree distribution. This is illustrated in Fig. 1.7. The layout of the clock drivers is done with the best possible layout techniques to reduce variation due to processing differences. For example, all transistors in clock buffers are shielded with dummy transistors. The clock wires are shielded to prevent noise from getting injected into the distribution, which would increase clock jitter. The clock buffers are designed to support clock gating at various levels, e.g. processor core level, functional unit level, and flip-flop group level (group of flip-flops that are driven by the same clock driver/s).

Some processors also use clock grids at various levels to reduce skew. Typically, clock grids consume more power than clock trees but result in smaller clock skew. One of the requirements for designing a clock grid is that the difference between arrival time of the clock signal at the input of the grid drivers shouldn't be more than a certain limit. Otherwise, you will have multiple grid drivers driving the clock grid in different directions for a substantial amount of time. The result is a bad/unacceptable clock waveform and lot of wasted VDD to VSS crowbar current/power. However, note that one of the functions of the grid is indeed to reduce the amount of skew of the clock signal at the input of the grid-drivers that gets transferred to the output. The ratio by which the skew at the input of the grid gets reduced when it appears at the output of the grid is referred to as the skew reduction factor. Another advantage of using a clock grid is to help connect together different regions having different amounts of clock load and reduce the overall clock skew between these regions. There are some processors

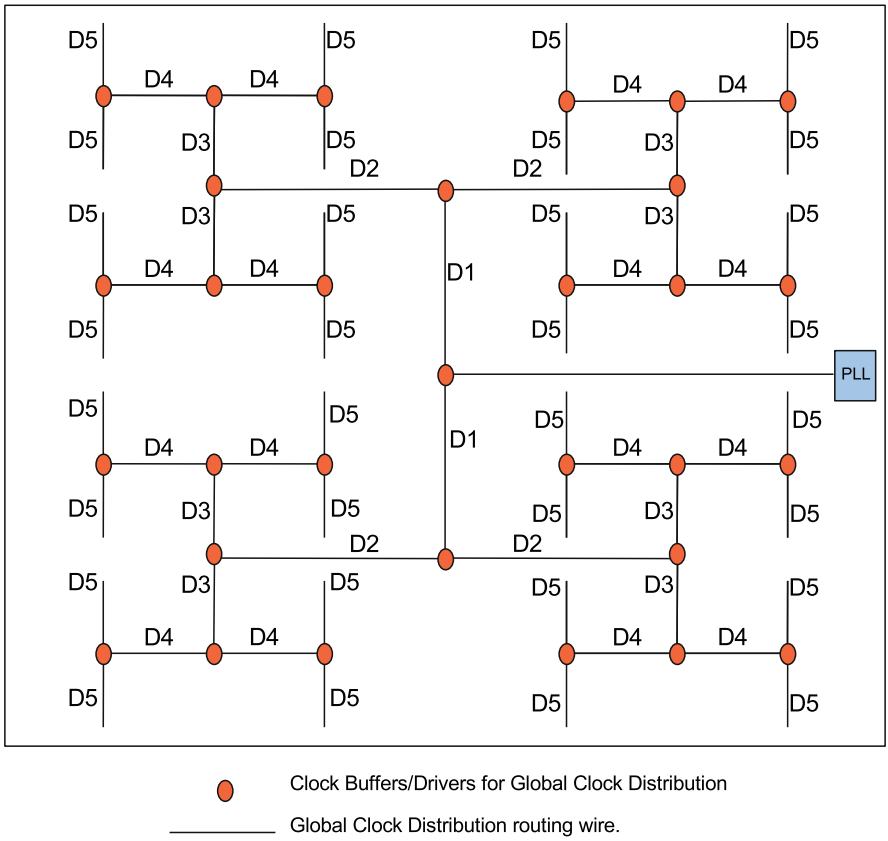


**Fig. 1.7** Typical chip global clock distribution

which employ a single clock grid over the entire chip. However this is not very common since it consumes too much power. Other techniques are used to reduce clock skew as well. For example, the clocking scheme in Intel's Merom Processor [7] implements an active deskewing scheme using digital delay lines to minimize the skew across the die.

Modern processors have several asynchronous and/or mesochronous interfaces on chip. For example, let's take a look at AMD's Quad Core Optron Processor [5]. It is based on a flexible clocking architecture, designed to scale across designs containing different number of cores. Each core has its own PLL, clock distribution circuit, and power grid. The clock (and power) to each core is controlled independent of the others. This allows each of them to be operated at their own performance/power points. Communication between the cores and SOC (Northbridge) is asynchronous. A synchronous mode is provided for tester determinism.

Reference [6] talks about a low power SOC from PA Semiconductor (which was since acquired by Apple Computer). The clocking for this processor is built

using a balanced H-tree distribution. The changing core VDD's poses a challenge to the clocking scheme. Due to the adjustable core VDD, the clock tree delay of the core is variable while the delay for the SoC is fixed. Keeping the core coherent with the coherent crossbar is vital for reducing Level 2 Cache (L2) and memory access latency and for simplicity of architecture and logic design. A phase detector is used to solve this problem. The phase detector detects the phase difference between the core and SOC clocks and the result is used to choose a specific source and destination clock pair to transfer data to maximize setup time and minimize hold time.

Modern processors have several hooks designed into the clocking schemes to support debug and also optimize frequency after testing silicon. Clock stop, clock stretch, and clock shrink have been used for a long time. Some processors employ clock drivers which support movement of clock edges in sections of the chip (w.r.t. the clock edges on the rest of the chip) as a debug aid. This is described in Ref. [7] and is illustrated in Fig. 1.8. Here, two sets of delay-tunable clock buffers 'A' and 'B' drive clocks to Clock domains 'A' and 'B' respectively. For all sets of frequency critical paths where the source flip flop and destination

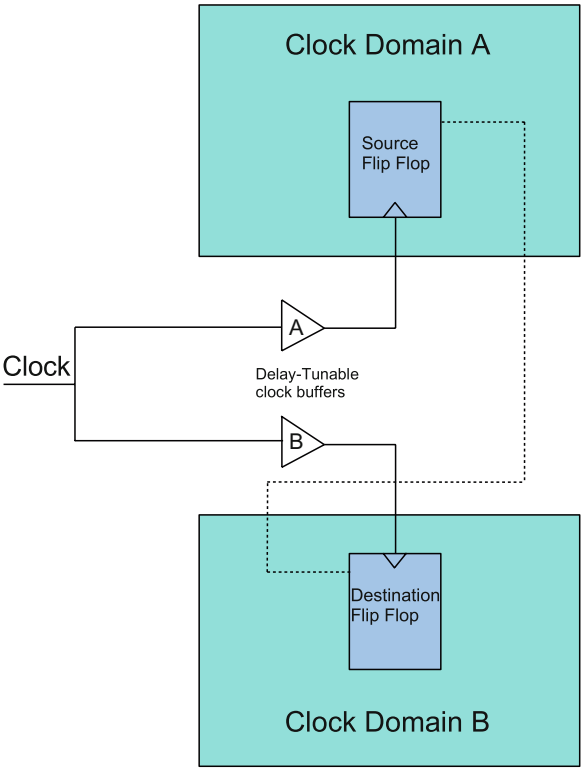


**Fig. 1.8** Locating critical paths in silicon by moving clock edges

flip flop are in different clock domains, the frequency at which the critical paths operate can be changed by varying the delays of clock buffers ‘A’ and ‘B’. This can help locate the exact critical path. This mechanism is referred to as the ‘Locate Critical Path’ (LCP) Mode in Ref. [7]. In LCP Mode, the delay of the local clock drivers is controlled by configuration registers programmed by the Test Access Port (TAP). Programming these enables after testing the chip can help us to get better performance for the same power point. IBM’s Power6 processor [4, 8]) essentially uses the same concept to alter flip-flop launch and capture times using programmable delays in the local clock buffers to improve performance of chip frequency critical paths.

### *1.4.1 Clock Skew/Clock Uncertainty*

Clock Skew or Clock Uncertainty is calculated differently when you are considering a frequency critical path vs. if you are considering a hold time path. Let’s take an example of a flip-flop based design which is synchronized using a single phase clock. For a single clock cycle critical path, under ideal circumstances, the rising edge of clock cycle ‘n + 1’ at the destination flip-flop should occur later than the rising of clock cycle ‘n’ at the source flip-flop by exactly the clock cycle period. Similarly for a half cycle critical path, under ideal circumstances, the falling edge of clock cycle ‘n’ at the destination flip-flop should occur later than the rising edge of clock cycle ‘n’ at the source flip-flop by exactly half the clock cycle period. When you are considering a hold time path, under ideal circumstances, the rising edge of clock cycle ‘n’ at the destination flip-flop should occur exactly at the same time as the rising edge of clock cycle ‘n’ at the source flip-flop. The amount of time by which the clock edge at the destination flip-flop deviates from its ideal position is called clock uncertainty or clock skew.

#### 1.4.1.1 Sources of Clock Skew/Clock Uncertainty

There are several sources of Clock Skew: Cycle-to-cycle PLL jitter, systematic skew, skew resulting from transistor and interconnect process variation, power-supply voltage noise induced skew, skew due to temperature gradient across chip. Assuming we are discussing Clock Skew in the context of a single clock cycle path, Cycle-to-cycle PLL jitter is defined as the amount by which the rising edge of clock n + 1 can deviate from its ideal position in time w.r.t. the rising edge of clock N at the output of the PLL. In other words, this is the amount by which the period of the cycle (at the source PLL) can differ from its ideal value. Hence it is also called PLL period jitter. Typically it is specified in terms of ‘ps’ per ‘sigma’ (i.e. a standard deviation). More often than not, the on-chip PLL is powered by a voltage regulated power supply to help minimize this.

Systematic skew is the skew induced due to the differences in the structure of the design. For example, ideally the clock designer attempts to match the length and

metal type of every branch of the H-tree distribution. But due to some reason, e.g. chip floorplan restrictions, there could be a finite mismatch in the wire length between two branches. The skew due to this mismatch in length is classified as systematic skew. Referring to Fig. 1.7, all wire sections labeled 'D1' (or 'D2' … 'D6') ideally have same metal routing topology (metal layer, length, width, spacing, etc). For example, if all 'D1' segments do indeed have identical topologies but one of the D1 segments is longer than the remaining D1 segments, this length difference will introduce a skew which will be categorized as systematic skew. Also, if a clock buffer is driving a group of flip-flops, the difference in clock arrival times at the inputs of these flip-flops, say due to RC delay, is also classified as systematic skew. Obviously, systematic skew is under the control of the designer. Every attempt should be made to minimize it.

Process variation is another component of clock skew. Even though a set of clock buffers at different locations of the chip have identical designs (including identical layouts), the buffers will have some variations between them once they go through the various processing steps and actually get built on chip. The variations could result from, from example, differences in gate length, gate width, differences in transistor threshold voltage, etc. Wires with identical layouts will also vary w.r.t. one another once they get built on the chip – due to variation in wire thickness, wire width, dielectric thickness, etc. Hence the actual delay through buffers and wires with identical layouts will be different and will result in clock skew. This skew can be minimized by using highly controlled layouts. Basic statistics tells us that a buffer with large number of gate fingers will have a smaller variation than a buffer with smaller number of gate fingers. Now-a-days (increasingly so with 45 nm technologies and beyond), the context in which a gate resides (the transistors/channel area that surrounds the gate) also has an impact on its delay. Hence, the context should be as controlled as possible.

Power-supply noise induces changes in buffer delays and hence results in clock skew. For this reason (as well as for several other reasons), a good quiet power supply distribution is very important in order to design a high performance processor. Having a lot of on-chip de-coupling capacitors is very important to reduce power-supply noise. It is especially important to have a lot of de-coupling capacitors near drivers that switch at high frequencies and high activity factors – like clock drivers. Finally, depending upon the kind of workload that is being run on the processor, a temperature gradient can exist across the chip and since temperature affects transistor drive current, it will induces differences in delay, which translate into clock skew.

## 1.5 Memory Design

In this context, the word 'Memory' is a term loosely used to refer to all storage elements on the chip that are not flip-flops. In a typical microprocessor, multiple types of memory cells are used depending upon the design requirements. Typically, the semiconductor foundary provides one or more 6-transistor (6T) memory cells.

The smallest cell is generally used for the largest cache memory. This is also the memory that is farthest from the processor in terms of memory hierarchy and has the largest on-chip access time. Since this cell is small, it has the smallest read current. Generally, the foundary will provide a larger cell – which also has a larger read current and is used for memories that need higher performance, like the primary data/instruction caches. Cells with more transistors are generally used as well. The additional transistors in these cells are used to increase the number of read and write ports. These cells are generally used in structures like register files, queues, etc. However, the basic storage element is generally similar to the 6T cell. Hence it is instructive to study the 6T cell in more detail.

### *1.5.1 The 6-T Memory Cell*

Figure 1.9 shows a basic representation of a 6T memory cell. The back-to-back inverters store the data. The cell is accesses using the Word Line (WL) and the Bit Lines (BL and BLb – collectively referred to as BLs) through the two N-pass transistors. Of the transistors in the back to back inverters, the P-transistors act as 'load' transistors and the N-transistors act as 'driver' transistors. The ratio of the drive strength of the 'driver' transistor to the pass-gate transistor is called the 'β' ratio. If we normalize all drive strength w.r.t. the pass-gate transistor, the drive strength of the 'driver' transistor will be 'β'. This ratio is an extremely important parameter in the design of a 6T cell. It affects the DC and AC performance of the cell and also affects the cell area. Ultimately, the 'β' ratio ends up being a tradeoff between write margin, memory cell stability, read current, and cell area. The cell is accessed by driving the WL high. To write the cell, the write drivers have to drive the BLs (which have several memory cells connected to them) and over-power the 'driver' transistor of the cell being written. To read data from the cell, the BLs are pre-charged high, WL of the accessed cell is asserted, and the 'driver' transistor of

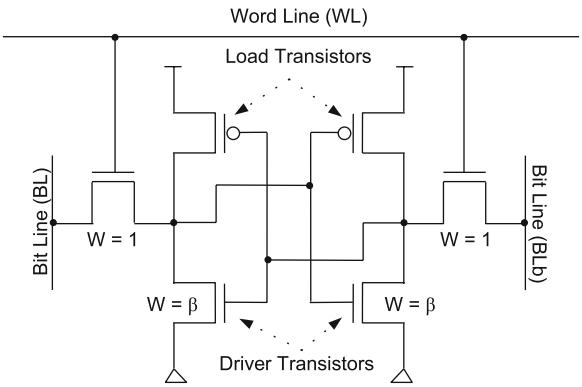


**Fig. 1.9** 6-T SRAM cell

the side that stores a low slowly starts pulling the BL low. Care has to be taken to make sure the accessed cell doesn't flip the data it has stored (read-disturb). If 'β' is too large, the cell area becomes large and it becomes difficult to write new data into the cell. Also, if we assume the area of the cell to be fixed, a large 'β' ratio will limit the drive strength of the 'pass-transistor' and thus limit read current and hence speed. If 'β' is too small, the cell becomes vulnerable to read-disturb due to power supply noise, channel length/width and threshold voltage imbalances between the two sets of transistors in the storage element, etc. The read current also suffers.

#### 1.5.1.1 Important Metrics/Tests for Evaluating a 6-T Memory Cell

There are several kinds of tests that are performed on the memory cell itself to make sure it is robust enough. One of them is the Static Noise Margin (SNM) test. This tests the ability of the cell to store logic 0 or logic 1(VDD) in spite of external DC noise sources, e.g. variation in channel width/length and threshold voltages between like transistors on either half of the (symmetric) cell, unequal transistor degradation over the lifetime of the cell due to '1' being stored for a longer time than '0' or vice versa, etc. This is most often performed by plotting the widely used 'butterfly-plot' of the cell (Fig. 1.10). The diagonal of the largest square (squares in Fig. 1.10) that can fit inside the area enclosed by the transfer curves represents the SNM of the

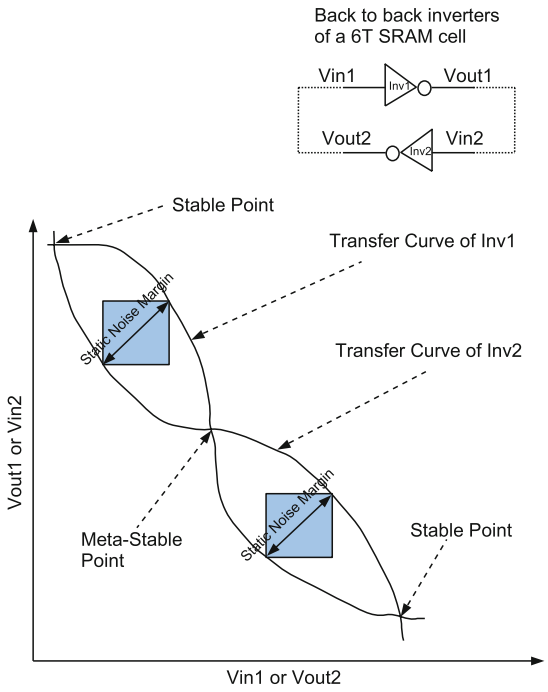


**Fig. 1.10** Butterfly plot of a 6T SRAM cell

cell. A second test tests the writeability of the cell. This test find the difference in the voltages of the BLs required to flip the cell. A third test tests the stability of the cell at different values of VDD. This is performed by applying a very slow (to make it appear as a DC test) VDD ramp and checking the lowest VDD at which the cell is able to maintain its stored data (or the VDD at which the cell flips to the opposite state). This helps define the 'Vmin' of the cell (the minimum voltage at which the cell is functional) and also helps understand the robustness of the cell to Power Supply offsets. A fourth test tests the read current of the cell. This test is performed to help us measure the amount of current the cell can sink to pull one of the pre-charged bitlines below VDD. The larger the current, the faster the cell can pull the BL low and faster the cell data can be read.

### *1.5.2 Memory Redundancy*

Redundancy strategy is a big part of memory design in microprocessors today. Most designs have built in redundant rows or columns of memory cells. The redundancy strategy has to be conceived after detailed discussions with the semiconductor foundary on yield, failure modes (rows/columns/random bit fails), etc and the processor architects on the desired memory architecture. The presence of redundant rows and/or columns of memory cells enables test engineers to recover chips that otherwise would have to be thrown away by replacing bad rows, columns, or bits of memory cell by spare functional rows or columns. This is implemented by simple multiplexers that are controlled by fuse bits that are burned into the fuse array at the time of testing the die. Needless to say, the redundancy strategy can have a big impact on the cost of the product as well.

### *1.5.3 The Importance of Statistical Analysis*

Statistical analysis is becoming more and more important in the design of processors today. This is especially the case for memory design. There are several reasons as to why that is the case. A big reason is that the caches in modern processors generally are several MBytes in size. So, there are millions of 6T cells on the same chip, most (all if there is no redundancy built into the design) of which need to function for the die to yield. Also, transistor variability, especially for the small transistors used in the 6T cells, is increasing.

Variations can be broadly classified into:

1. Chip Mean (CM) variation
2. Across Chip variation, also called ACLV (Across Chip Line Variation)
3. Local Variation, also referred to as Mismatch (MM) variation

CM variation encompasses the entire spec range of the semiconductor manufacturing process. This includes the variation between different dies on the same wafer, wafer

to wafer variation, and lot to lot variation. ACLV variation is the variation seen across the same die. In effect, this is the variation across the scanner field. MM variation is the variation between two identical transistors or wires (or any other identical component) that are placed next to each other (or within a local area) on a single die. The actual parameters that vary include (but are not limited to) transistor channel length, transistor channel width, transistor oxide thickness, transistor threshold voltage, interconnect thickness, interconnect width, inter/intra-layer dielectric thickness, etc.

Each metric of interest for a 6T cell has to be evaluated with statistical models that account for all these kinds of variation and the designer has to ensure that the appropriate operating margin exists after accounting for the statistics associated with the number of instances of each memory cell that the designer plans to have on the chip. Obviously, the desired yield and the amount of redundancy that is built into the design will have an impact on these calculations as well. The details of how these calculations are done are outside the scope of this chapter.

## 1.6 Process Technology and Impact of Layout on Performance and Power

As semiconductor manufacturing technology has continued to advance from one generation to next as per Moore's Law, transistor geometries have continued to shrink and the interaction between transistor layout and key transistor parameters has continued to increase. As a result, the way a transistor is laid out and the context in which the transistor exists on chip has come to have a significant impact on the performance of the transistor. So, layout and context-induced performance changes have to be taken into account when designing and characterizing circuits. The use of stress/strain to improve transistor performance has also contributed to this. For example, tensile stress along the channel direction of an NMOS transistor (and compressive stress along the channel direction of a PMOS transistor) improve the transistor's drive current. So, semiconductor manufacturers deposit stress liners on transistors to get the desired kind of stress to boost performance. If there are several transistors of the same width and length in a stack, each transistor will see different amounts of stress and will thus have different drive current. Shallow Trench Isolation (STI) technique is the technique of choice to isolate one transistor from another. The Oxide in the STI also generates stress in the adjoining diffusion thus affecting the transistor performance. The amount of stress will depend upon the distance of the STI from the diffusion of the transistor. Similarly, ions during well implants can scatter at the edge of the photoresist and can get embedded in transistor channel regions. These act as additional dopants and alter the threshold voltage of the transistor. The closer the transistor channel is to the edge of the well, the higher is the concentration of the scattered dopants. Photolithographic effects are also a big factor, especially since the geometries being drawn are smaller and smaller fractions of the wavelength of light being used. 90° angles on layout actually

become rounded edges on silicon and this has to be taken into account when doing layouts and predicting performance of the transistors. In most modern semiconductor technologies, all transistors need to be shielded by dummy transistors or dummy field polys to ensure that the channel length variation is kept below a certain limit. For modern technologies, especially 45 nm and beyond, transistor spice models now have to take the layout type and the layout context into account, and hence the transistor models have become very complex.

## 1.7 Conclusion

Microprocessor design has dramatically evolved since the birth of the first Microprocessor (Intel's 4004) in the early 1970s, to today. The Intel 4004 had a little more than 2,000 transistors. Today's microprocessors can have more than two billion. As can be expected, design and manufacturing complexity has increased by several orders of magnitude. Computing power has found more and more applications. The hunger for computing power has increased and this has propelled the field of microprocessor design forward.

In the last 10 years, the growth of the internet caused a dramatic increase in the need for more computing power. Tens of thousands of data centers were built. Data centers started hitting the coolable power limits. Power-efficient design and Power-management rapidly gained in importance. CMT became popular.

What would the next 20 years bring? What would a microprocessor look like 20 years from now? How would the computing landscape change during the next 20 years? Only time will tell. It would be foolish to venture a guess but let me do so just for the sake of it. Microprocessors would likely have 100s of billions, of transistors. Computing threads will be available in plenty. These computing threads might be spread across multiple locations all around the globe separated by thousands of miles, but will likely be accessible enough through the internet. There would be great advances in software to make use of this computing power more efficiently. Lowering power consumption will continue to be extremely important. Network computing will truly come into its own. Whatever happens, one thing looks certain - the hunger for computing power will continue to increase and the field of microprocessor design will continue its march forward.

## References

1. Umesh Nawathe et al., An 8-Core, 64-Thread, 64-Bit, Power Efficient SPARC System on a Chip, ISSCC, February 2007.
2. Greg Grohoski et al., Niagara2: A Highly Threaded Server-on-a-chip. Hot Chips Symposium, August 2006.
3. Robert Golla et al., Niagara2: A Highly Threaded Server-on-a-chip, Microprocessor Forum, October 2006.

4. Joshua Friedrich et al., Design of the Power6™ Microprocessor, ISSCC, February 2007.
5. J. Dorsey et al., An Integrated Quad-Core Optron™ Processor, ISSCC, February 2007.
6. Zongjian Chen et al., A 25W SoC with Dual 2GHz Power™ Cores and Integrated Memory and I/O Subsystems, ISSCC, February 2007.
7. Nabeel Sakran et al., The Implementation of the 65nm Dual-Core 64b Merom Processor, ISSCC, February 2007.
8. B. Curran et al., Power-constrained high-frequency circuits for the IBM POWER6 microprocessor, IBM J. RES. & DEV. VOL 51 NO 6, November 2007

# Chapter 2
# Towards High-Performance and Energy-Efficient Multi-core Processors

**Zhiyi Yu**

**Abstract** Traditional uni-core processors have met tremendous challenges to improve their performance and energy efficiency, and to adapt to the deep submicron fabrication technology. Meanwhile, traditional ASIC implementations are also widely prohibited due to their inherent inflexibility and high design cost. On the other hand, rapidly advancing fabrication technologies have enabled the integration of many processors into a single chip, called multi-core processors, and promise a platform with high performance, high energy efficiency, and high flexibility.

This chapter will discuss the motivations of shifting from traditional IC systems (including uni-core processors and ASIC implementations) to multi-core processors, investigate the design cases of multi-core processors and their key features, and look forward to the future work.

**Keywords** Multi-core • Processors • ASIC • Energy-efficient • Network-on-Chip (NoC)

## 2.1 Motivating Multi-core Processors

### *2.1.1 Challenges on Uni-core Processors*

Previous IC designers have been mainly concerned with fabrication cost and performance. Minimizing the number of transistors to reduce the area is the main approach to reduce the cost, and increasing the clock frequency is the main approach to increase the performance. Currently, how to achieve energy efficiency and how to adapt to the advanced fabrication technologies also become important challenges.

Z. Yu (✉)
State-Key Laboratory of ASIC & System, Fudan University
No. 825 Zhangheng Rd., Shanghai, 201203, P.R. China
e-mail: zhiyiyu@fudan.edu.cn

K. Iniewski (ed.), *CMOS Processors and Memories*,
Analog Circuits and Signal Processing, DOI 10.1007/978-90-481-9216-8_2,

#### 2.1.1.1 High Performance Innovations are Challenging

Increasing clock frequencies and using wide issue architectures has worked well to improve processor performance, but recently has become significantly more challenging.

Deeper pipelining is one of the key techniques to increase the clock frequency and performance, but the benefit of the deeper pipeline is eventually diminished when the inserted Flip-Flop's delay is comparable to the combinational logic delay. Moreover, deeper pipeline stages increase cycles-per-instruction (CPI) and negatively impacts the system performance. Research has found that the depth per pipeline is approximately 8 Fanout-4 (FO4) inverter delays to obtain the highest performance [1], which corresponds to 20–30 pipeline stages for a typical programmable processor. The pipeline delay of some modern processors is already close to ten FO4 [2] so the deeper pipelining technique for high performance is reaching its limit. Also, increasing pipeline stages necessitates more registers and control logic, thereby further increasing design difficulty as well as power consumption. As reported by A. Hartstein [3], the optimum pipeline depth for maximum energy efficiency is about 22.5 FO4 delay (about 7 stage pipeline), using $BIPS^3/W$ as the metric – BIPS are billions of instructions per second.

The other key technology – shrinking the size of transistors to increase the clock frequency and integration capability – has amazingly followed Moore's Law [4] for about 40 years. Although the pace of this innovation is still going on, the limitations are right ahead in another couple of generations: either because of the physical limit when the size of transistors approaches the size of atoms, or because of the fabrication cost prohibiting further progress.

Wide issue processor architectures such as VLIW and superscalar are other efficient approaches for high performance computation while their benefit is also quickly diminished when the issue width is more than 10; since most applications do not have many independently parallel executable instructions per fetch. For example, an eight-way VLIW TI high performance C6000 DSP [5] and an eight-wide superscalar RISC microprocessor [6] was reported in 2002, and there are no wider-issue examples reported since then.

As the result of all those challenges mentioned above, the nearly 50% performance improvement per year lasting about 20 years has comes to the end, as shown in Fig. 2.1. New computer architectures are urgently demanded to overcome those challenges.

#### 2.1.1.2 Power Dissipation Becomes the Key Constraint

The high performance design of modern chips is also highly constrained by power dissipation as well as the circuit constraints. Power consumption is generally dominated by dynamic power with the trend that leakage power is playing a growing role. The dynamic power of a common static gate is described by Eq. 2.1

$$P = aCV^2f \tag{2.1}$$

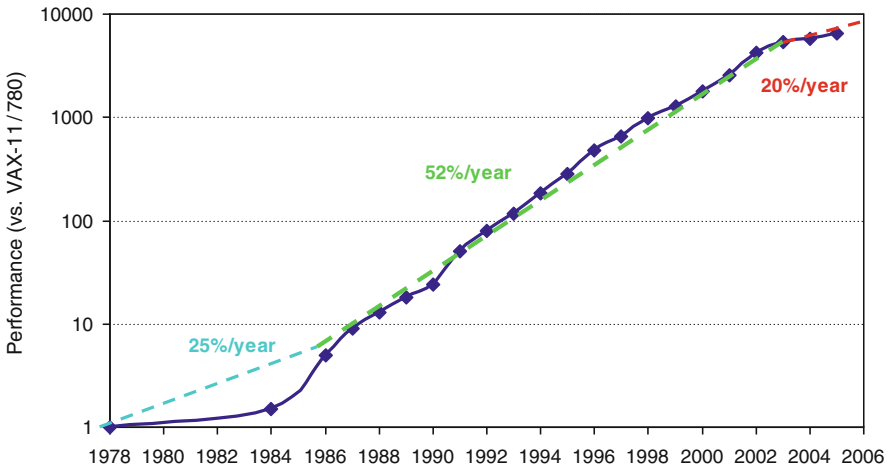


**Fig. 2.1** Growth in processor performance since 1978 [7]. Further improving of the processor performance is challenging and the 16-year renaissance is over

where a is the circuit state transition probability, C is the switched circuit capacitance, V is the supply voltage, and f is the clock frequency. The leakage power mainly results from the reduction of the transistor threshold voltage [8] and is also highly dependent on the supply voltage. Most high performance techniques, such as increasing clock frequencies and increasing processor issue-widths (which means increasing number of circuits and increasing capacitance) result in higher power consumption. All these imply a new era of high-performance design that must now focus on energy-efficient implementations [9].

Portable devices powered by batteries are strongly affected by their power consumption since it determines their operational life time between each battery charging. Traditional non-portable systems such as PCs are also concerned with power consumption, since it highly determines the packaging costs, cooling system costs, and even limits the operation speeds and integration capacities of the systems. Figure 2.2 shows the power consumption of main Intel microprocessors from 1970 to 2006. The data between 1970 to 2000 is from S. Borkar [10] where he found that the changes of the power consumption follow the Moore's law increasing from 0.3 to 100 W, and estimated that the processor power consumption will go up to 1,000 W in a couple of years if this trend continues. Similarly to the power consumption, the power density increased significantly from a couple of W/cm² to about 100 W/cm² and becomes another key issue. This trend has been mostly halted recently thanks to low power techniques such as voltage and frequency control technologies. The power consumption of recently-reported microprocessors is still around 100 W. It also implies that 100 W is the power limit the current packaging technology and cooling technology can tolerate at reasonable cost. Power consumption has become the highest constraint for designers and limits the achievable clock frequency and processor performance [9].

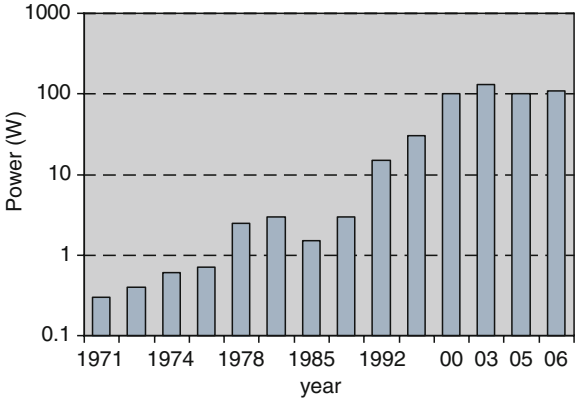


**Fig. 2.2** Power consumption of Intel microprocessors from 1970 to 2006; data from 1970 to 2000 are from [10]; data in 03, 05, and 06 are from [11–13] respectively

#### 2.1.1.3 The Gap Between Dream and Reality on Performance and Energy Efficiency

H. D. Man shows a future ambient intelligent computing example which illustrates the gap between the future requirement and the current reality in both performance and power [14]. In his opinion, there will be three major devices in the future intelligent computing system. One is the main powerful computation components, like today's PC; its target performance is 1 TOPS, with power consumption less than 5 W, corresponding to the energy efficiency 100–200 GOPS/W. This requirement is about 1,000 times higher than today's PC which has about 10 GOPS while consuming 100 W, corresponding to 0.1 GOPS/W. The second device is the handable devices powered by battery, like current mobile phone, targets to 10–100 GOPS with less than 1 W, corresponding to 10–100 GOPS/W. This requirement is about ten times higher than current solutions which use RISC processors and/or DSP processors. The third component is the sensor network to receive and transfer information, powered by energy harvesting methods such as mechanical vibration, with power consumption less than 100 μW; developing such low power components is another challenging topic.

#### 2.1.1.4 Future Fabrication Technologies Imposing New Challenges

Future fabrication technologies are also imposing new challenges such as wiring and parameter variations.

In the early days of CMOS technology, wires could be treated as ideal. They transmit signals with infinite speed, without power consumption, and without coupling effect. This assumption is no longer true. For global wires, their length is nearly constant along with the technology scaling if the chip size stays the same; which makes their delay nearly constant. Compared to gate delay which scales

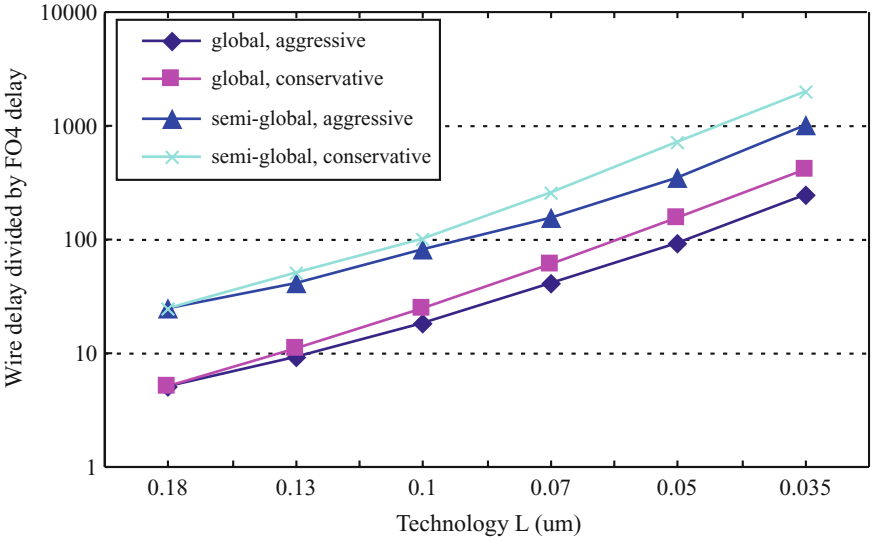


**Fig. 2.3** Wire delays (in FO4s) for fixed-length (1 cm) wires [15]; global wires have larger width, height, and spacing which result in smaller resistance than semi-global wires; aggressive and conservative are two extreme scaling projections

down with the technology, the delay of the *global* and *long* wires scales up with the technology. As shown in Fig. 2.3 [15], a 1 cm long global wire delay in modern 65 nm technology is around 100 FO4 delay; which corresponds to more than one clock cycle period in modern processors and the global wires have to be pipelined or be eliminated through architecture level innovations. Similar with the delay, the power consumption of the long wires also scales up along with the technology compared to the gates. Besides the effect on delay and power consumption, the inductive and capacitive coupling between wires adds signal noise and impacts system reliability.

Furthermore, future fabrication technologies are expected to have tremendous variability compared to current technologies in both gates and wires. Fundamental issues of statistical fluctuations for submicron MOSFETs are not completely understood, but the variation increases leakage of transistor and causes a variation of the speed of individual transistors, which in turn leads to IC timing issues [16]. Borkar et al. reported chips fabricated in advanced nanometer technologies can easily have 30% variation in chip frequencies [17]. This variance will be present at time of fabrication and also have a time-varying component.

### *2.1.2 Challenges on ASIC Implementations*

ASIC implementations have the key advantages of much higher performance and energy efficiency compared with other implementation techniques such as programmable processors and FPGAs, which makes ASICs widely used in applications

**Table 2.1** The multiple HDTV transmission standards, traditionally implemented using ASICs

| Standards | DVB-S2 | DVB-C | DVB-T | ATSC | DTMB |
|---|---|---|---|---|---|
| Tran. media | Satellite | Cable | Wireless | Wireless | Wireless |
| Countries | Europe | Europe | Europe | USA | China |

where high performance and high energy efficiency are the key design constraints. But unfortunately, ASIC implementations have inherent drawback of lacking flexibility, which make it less attractive along with the keep increasing design and fabrication cost, especially in application domains demanding multi-standard solutions.

Firstly, the design cost and fabrication cost of IC systems keep growing because of the increasing complexity of modern chips and the more advanced fabrication technologies. The total design and fabrication costs of a modern chip can easily run into the tens of millions of dollars. It is desirable to make the IC systems flexible enough so that the cost can be shared among a variety of applications.

In addition, nowadays many applications (such as communication, and multimedia) are implemented based on specific standards. Different application domains have different standards, and even in one application domain there can have multiple standards. As shown in Table 2.1, High-definition television (HDTV) transmission systems have at least five standards for different countries (areas) and different transmission media. Consumers certainly wish to have a product which can be used in different places and the product suppliers also wish to have ONE solution for all areas to simplify their management. In other words, people want to see flexible solutions which can support multiple standards of one application domain, or even different application domains.

Overall, it is highly desirable to have high flexible IC systems to lower the cost and to support multiple standards, but unfortunately ASICs lack flexibility inherently.

### *2.1.3 Solution: Multi-core Processors*

In order to address the challenges the uni-core processors and ASIC implementations faced, innovations on computer architecture and design are required. Deep submicron fabrication technologies enable very high levels of integration such as a recent dual-core chip with 1.7 billion transistors [12], thus reaching a key milestone in the level of circuit complexity possible on a single chip. A highly promising approach to efficiently use these circuit resources and to address the current IC design challenges is the multi-core processors which integrate multiple processors onto a single chip. The multi-core processors will soon enter a new era called multi-core processors as the number of cores in a single chip keeps increasing.