UNIT 1

Saturday, August 15, 2020 9:23 AM

UNIT 1

1. **Explain the sources of power dissipation on Digital integrated circuits.**
5. **Mention the source of power dissipation in the digital VLSI integrated circuits**

Static power consumption:

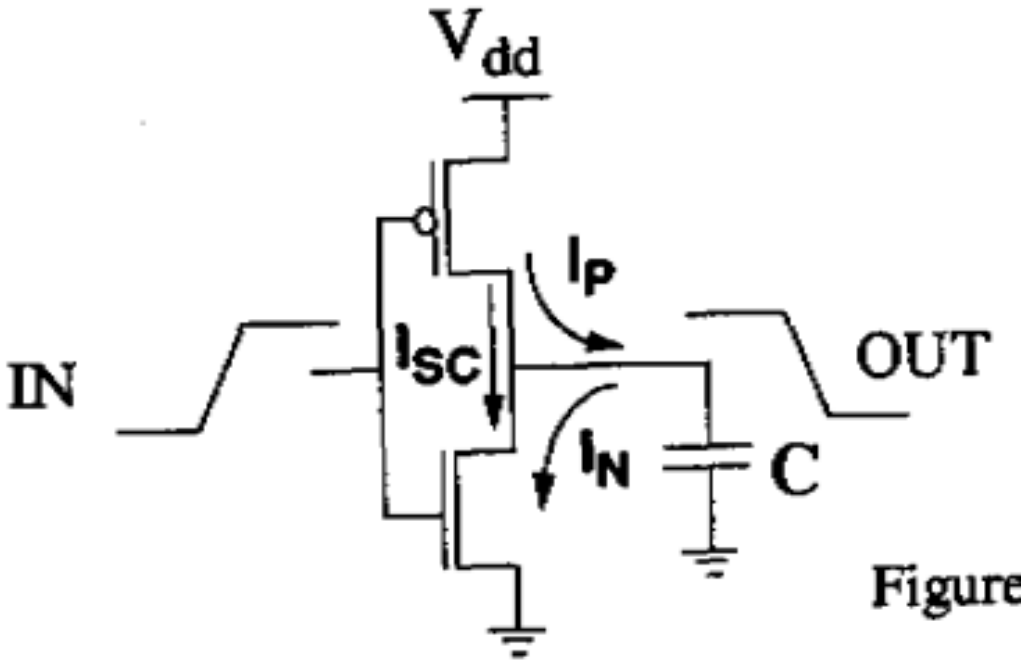
Ideally CMOS circuits dissipates no static DC power since in steady state there is no direct path from V_{dd} to ground. In reality MOS transistor is not a perfect switch. There will always be leakage currents and substrate injection currents, which will give rise to a static component of CMOS power dissipation.

Substrate injection current is on the order of 1-100μA for a V_{dd} of 5V. Since gate voltages are only transient in this range as device switch, the actual per contribution of substrate injection is several orders below other contributors. Reverse bias junction leakage currents associated with parasitic diodes in CMOS device structure are on the order if nano amps and will have leek effect on overall power consumption.

Another major contributor of static power can be found in ratioed logic. Here during switching a direct path from V_{dd} to ground is created and static current flows. If these types of logic families are avoided then static power consumption should be negligible.

Dynamic power consumption:

The dynamic component of power dissipation arises from the transient switching behavior of the CMOS device. At some point during the switching transient, both the NMOS and PMOS devices in Figure 1.4 will be turned on. This occurs for gates voltages between V_{tn} and V_{dd}-$|V_{tp}|$. During this time, a short-circuit exists between V_{dd} and ground and currents are allowed to flow. A detailed



Figure

short circuit dissipation cannot always be completely ignored, it is certainly not the dominant component of power dissipation in well-designed CMOS circuits.

Instead, dynamic dissipation due to capacitance charging consumes most of the power used by CMOS circuits. This component of dynamic power dissipation is the result of charging and discharging parasitic capacitances in the circuit. The

[redacted] Consider the behavior of the circuit over one full cycle of operation with the input voltage going from V_{dd} to ground and back to V_{dd} again. As the input switches from high to low, the NMOS pull-down network is cut off and PMOS pull-up network is activated charging load capacitance C up to V_{dd}. This charging process draws an energy equal to CV_{dd}^2 from the power supply. Half of this is dissipated immediately in the PMOS transistors, while the other half is stored on the load capacitance. Then, when the input returns to V_{dd} the process is reversed and the capacitance is discharged, its energy being dissipated in the NMOS network. In summary, every time a capacitive node switches from ground to V_{dd} (and back to ground), an energy of CV_{dd}^2 is consumed.

$$P_{dyn} = \alpha C V_{dd}^2 f$$

2. Discuss emerging low power approaches requires optimization at all design abstraction levels.

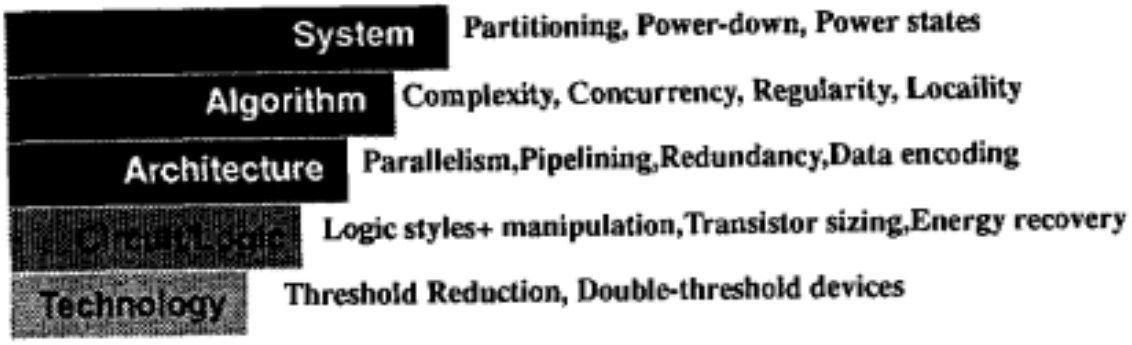


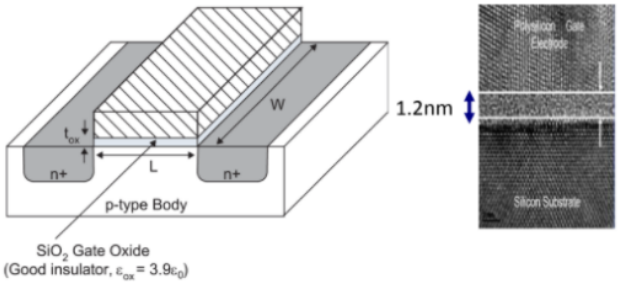
Figure 1.8 An integrated low-power methodology requires optimization at all design abstraction layers.

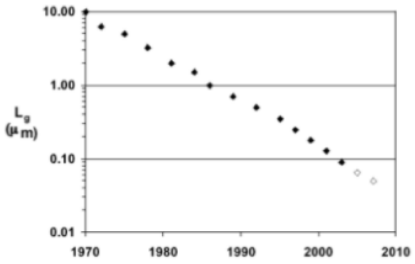
Barring a dramatic introduction of a novel low power manufacturing technology, it is now commonly agreed that low power digital design requires optimization at all levels of the design hierarchy, i.e. technology, devices, circuits, logic, architecture (structure), algorithm (behavior) and system levels, as is illustrated

The presented techniques and approaches ultimately all come down to a fundamental set of concepts: dissipation is reduced by lowering either *the supply voltage, the voltage swing, the physical capacitance, the switching activity* or a combination of the above (assuming that a reduction in performance is not allowable).

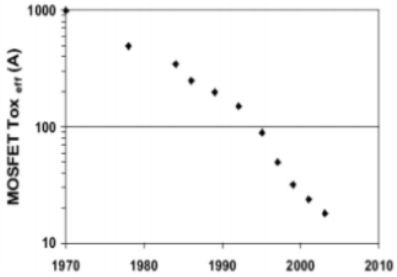
3. Explain the impact of technology scaling in CMOS.

7. How technology scaling is done in CMOS technology. What are its impact on device performance.(Look at the table and explain in your own words)

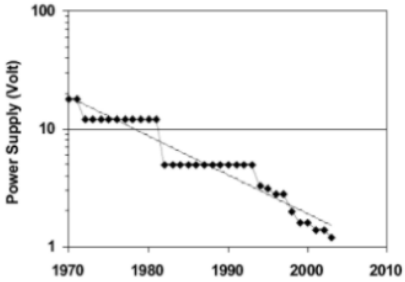




minimum feature size (gate length)



Current oxide thickness ~ 1.0 – 2.0nm thickness → 3 – 4 atomic layers of oxide



Power supply voltage scaling

Delay: $\tau = \frac{CV_{dd}}{4}\left(\frac{1}{I_{dsatn}} + \frac{1}{I_{dsatp}}\right)$, Energy: $E = CV_{dd}^2$

	L	V_{dd}	T_{ox}	W_p/W_n	W_p+W_n
to minimize τ	min	max, $>4V_t$	$C_{tox}=\frac{C}{2}$	1-3	max
to minimize τE	min	$2V_t$	$C_{tox}=\frac{C}{4}$	1-3	$C_d=\frac{C}{2}$

C: total load capacitance, C_{tox}: all load capacitances attributable to gate oxide, C_d: load capacitance attributable to driver devices.

Table 2.1 Optimization for Delay and Delay-Energy Product

Delay: $\tau = \frac{CV_{dd}}{4}\left(\frac{1}{I_{dsatn}} + \frac{1}{I_{dsatp}}\right)$, Energy: $E = CV_{dd}^2$

	L	V_{dd}	T_{ox}	W_p/W_n	W_p+W_n
to minimize τ	min	max, $>4V_t$	$C_{tox}=\frac{C}{2}$	1-3	max
to minimize τE	min	$2V_t$	$C_{tox}=\frac{C}{4}$	1-3	$C_d=\frac{C}{2}$

C: total load capacitance, C_{tox}: all load capacitances attributable to gate oxide,

C_d: load capacitance and available to drive devices.

Table 2.1 Optimization for Delay and Delay-Energy Product

Gate Length (μm)	3	2	1.5	1	0.7	0.5	0.35	0.25	0.18	0.12
V_{dd} (V)	5	5	5	5	5	5/3.3	3.3/2.5	2.5/2.0	1.5	1.5
V_T (V)	0.7	0.7	0.7	0.7	0.7	0.7/0.7	0.7/0.6	0.6/0.5	0.4	0.4
T_{ox} (nm)	70	40	25	25	20	15/10	9/7	6.5/5.5	4.5	4
I_{dsatn} (mA/μm)	0.1	0.14	0.23	0.27	0.36	0.56/0.35	0.49/0.40	0.48/0.41	0.38	0.48
I_{dsatp} (mA/μm)		0.06	0.11	0.14	0.19	0.27/0.16	0.24/0.18	0.23/0.19	0.18	0.24
Inverter Delay (ps)	800	350	250	200	160	90/100	70/65	50/47	40	32

Table 2.2 Impact of V_{dd}, L, and T_{ox} scaling on MOSFET current and inverter speed

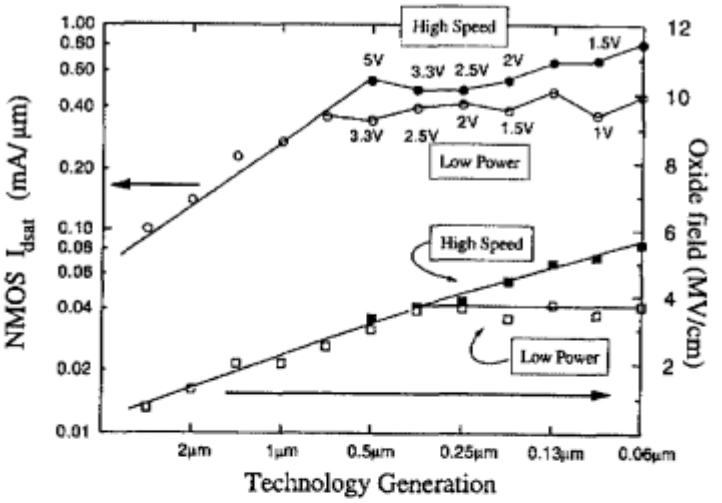


Figure 2.8 MOSFET current hardly increases beyond the 0.5 μm generation of technology due to velocity saturation even in the high-speed scenario, where oxide field rises aggressively.

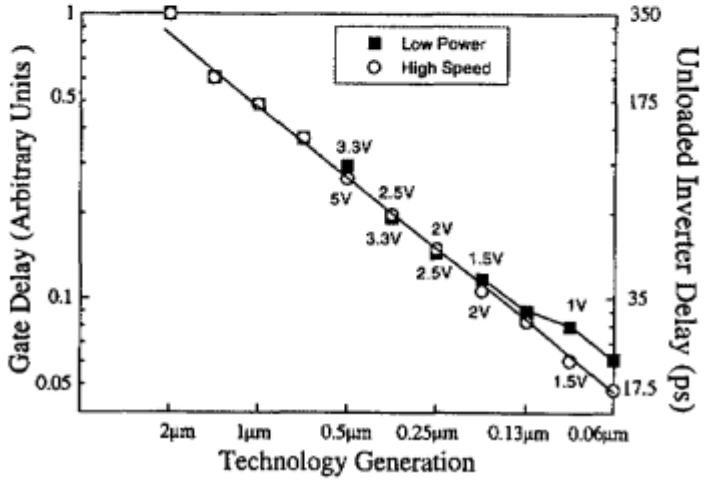


Figure 2.9 Speed in the low-power scenario lags that in the high-speed scenario, where speed doubles every 4 generations, rather than 2 generations as in the past.

4. What is the need for low power design in the VLSI circuits

Power dissipation was neglected due to : Low device density, Low operating frequency.

Now it is important issue due to : High device density, High operating frequency, Proliferation of portable consumer electronics, Concerns on Environments and energy sources.

Competitive Reasons –

- Battery Powered Systems – Extended Battery Life and reduce weight and size.
- High-Performance Systems

Cost -

- Package (chip carrier, heat sink, card slots, plenum, ...)
- Power Systems (supplies, distribution, regulators, ...)
- Fans (noise, power, reliability, area, ...)
- Operating cost to customer – Energy Star issue.

Reliability –

- Failure rate increase by 4X for Tj @ 110C vs 70C
- Mission critical operation at 100C

Size and Weight – Product footprint (office and desk space)

6. How the Dynamic dissipation is dominant in CMOS circuits.(refer question 1)

What are the Techniques to reduce it.

static and dynamic component. Dynamic power is usually the dominant component and is incurred only when the node voltage is switched.

$$P = P_{dynamic} + P_{static} \tag{2.1}$$

$$P_{dynamic} = CV_{dd}^2 \alpha f + P_{sc} \tag{2.2a}$$

$$P_{sc} \approx V_{dd} I_{sc} \frac{\tau_{in}}{4} 2f \approx V_{dd}^2 f \frac{C}{10} \tag{2.2b}$$

The second term in Eq. (2.2a) is known as the short-circuit power. When the inverter input is around $V_{dd}/2$ during the turn-on and turn-off switching transients, both the PFET and the NFET are on and a short circuit current I_{sc} flows from V_{dd} to ground. The width of this short-circuit current pulse is about 1/4 of the input rise and fall time. This term is typically only 10% of the first term as shown in Eq. (2.2b). There, we assumed $\tau_{in} \approx \tau_{out} \approx V_{dd}C/I_{dsat} \approx V_{dd}C/5I_{sc}$. Combining Eq. (2.2a) and Eq. (2.2b).

$$P_{dynamic} = kCV_{dd}^2 \alpha f \tag{2.3}$$

$$\text{Switching Energy, } E = kCV_{dd}^2 \tag{2.4}$$

C = oxide capacitance + junction capacitance + interconnect capacitance

$$= C_{ox} + C_j + C_{int} \quad (2.5a)$$

$$= \frac{b}{T_{ox}} + C_j + C_{int} \quad (2.5b)$$

What can one do to reduce $P_{dynamic}$? k is approximately 1.1 with a lower bound of 1.0. No one seems to have a clever idea for reducing it except for raising the ratio V_t/V_{dd}. We will discuss ways of minimizing C later. f is the clock frequency. αf is the average rate of cycling this node experiences. For example, an idle block of the circuit may not experience switching because the clock signals to the function blocks are gated. In this case α may be much smaller than one.

8) What are the innovative techniques developed in CMOS technology to improve its performance.

10) Explain technology and device innovation techniques developed in CMOS.

High speed low power devices based on quantum tunneling, single electron effect, etc have excellent switching speed but are not capable of driving the capacitance of long interconnects. It is also quite difficult to manufacture.

i) GaAs based chips are expensive with respect to Si but are very high speed.

ii) Silicon-On-Insulator (SOI) can improve delay & power through a reduction by 25% in total capacitance. Optimised SOI MOSFET have lower capacitance & higher I_{Dsat}.

iii) Minimum possible width for metal interconnects can be used to reduce capacitance.

iv) Low permittivity insulators for inter metal dielectrics reduce metal capacitance.

v) Dynamic threshold MOS can be used for very low power V_{dd} (< 0.5 V).

9) Explain transistor sizing and gate oxide thickness impact on device performance.

For a given $W_n + W_p$, there is a certain W_p/W_n ratio that minimizes τ (Eq. 2.12 and τE (Eq. 2.13) as shown in Figure 2.5. This value can be obtained by differentiating $1/(\kappa - W_p) + 2.2/W_p$ with respect to W_p ($\kappa = W_n + W_p$ = constant) and equating the differential to zero. This optimal ratio is independent of the load capacitance.

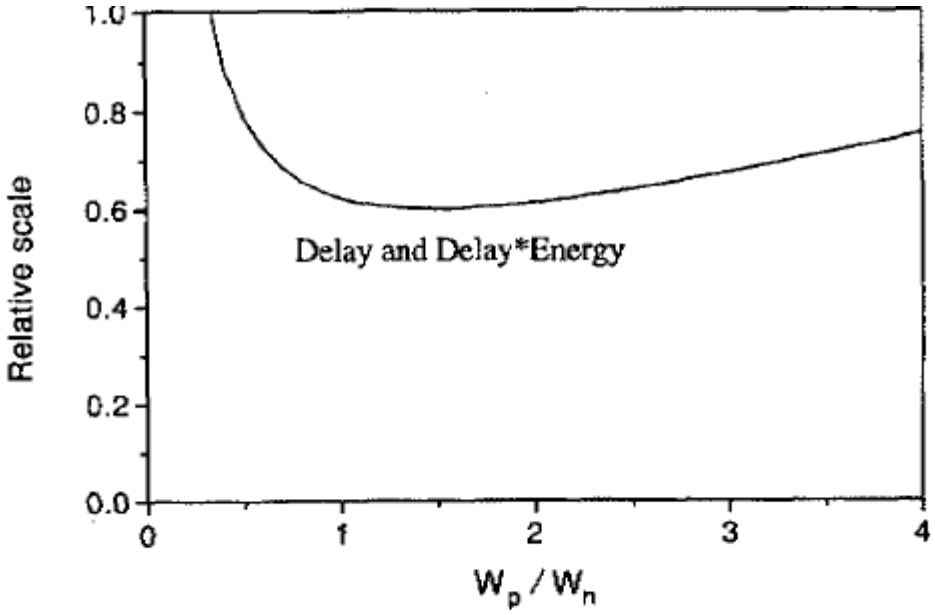


Figure 2.5 Both delay and delay-energy product have a very broad minimum around W_p/W_n=1-3, independent of load capacitance.

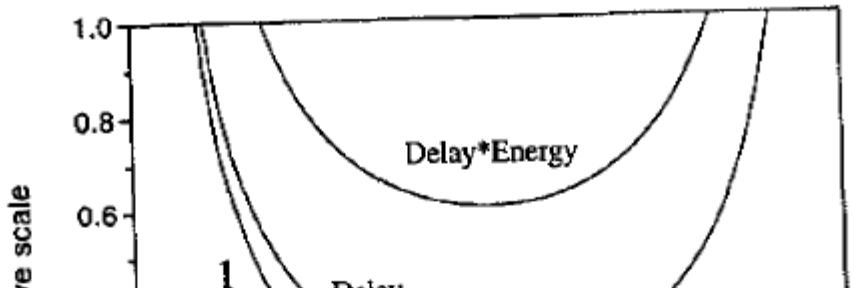
$$\frac{W_p}{W_n} = \sqrt{2.2} = 1.5 \tag{2.15}$$

Figure 2.5 shows a very broad minimum. $W_p/W_n = 2$ is also a good choice. C in Eq. (2.5) may be divided into a factor contributed by the driver devices and the rest of the load.

$$C = a(W_n + W_p) + C_{other} \tag{2.16}$$

What value of $W_n + W_p$ should one choose? Eqs. (2.12 and 2.13) together with Eq. (2.16) indicate that τ decreases monotonically with increasing $W_n + W_p$ as shown in Figure 2.6. It also shows that, regardless of W_p/W_n, E is minimized when the drive devices contribute the same amount of capacitance as the load devices and the interconnect.

Eq. (2.12) and Eq.(2.5b) indicate that τ is minimized when b/T_{ox}, i.e. the total oxide capacitances, contributes half the loading. Eq. (2.13) and Eq. (2.5c) indicate that τE is minimized when the total oxide capacitance is $^1/_4$ of the load. These facts are illustrated in Figure 2.7.



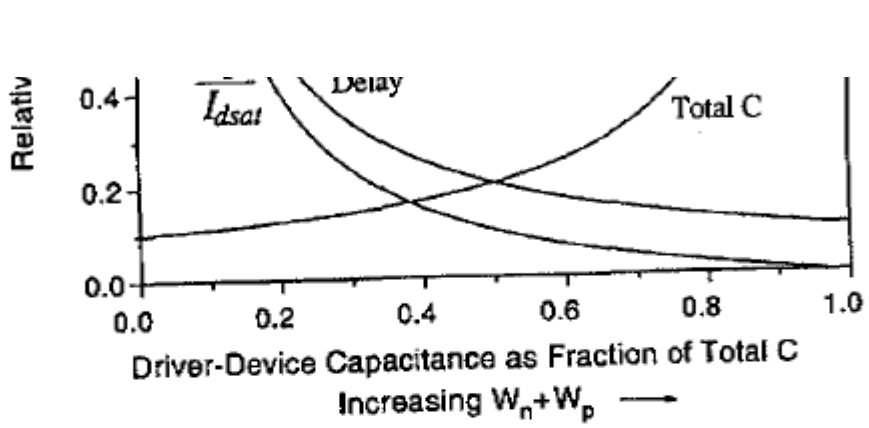


Figure 2.6 Delay decreases monotonically with increasing W_n+W_p but the delay-energy product is minimized when the drive devices contribute half of the total load capacitance.

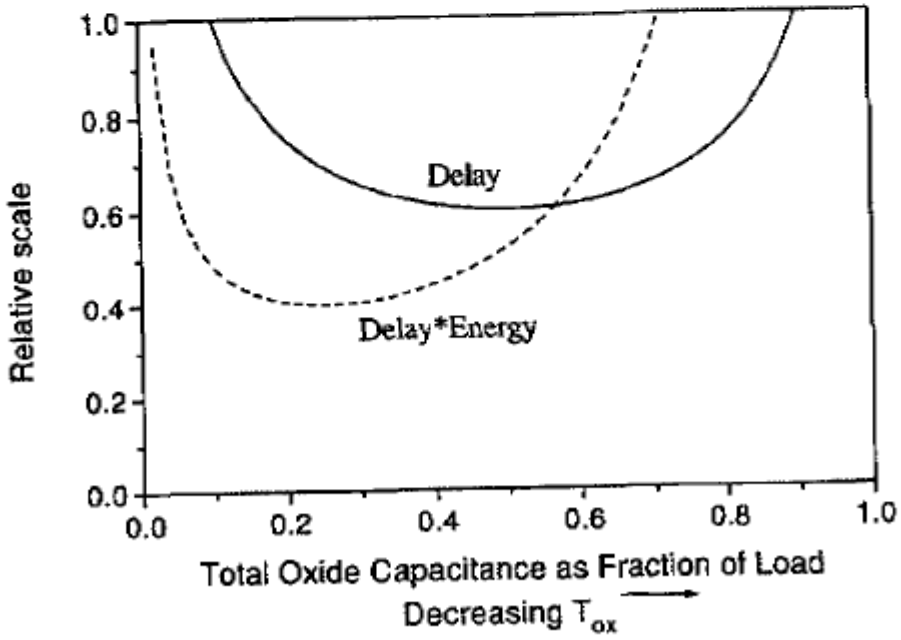


Figure 2.7 Minimum delay is obtained when T_{ox} is chosen such that oxide capacitance accounts for half the total load. The minimum delay-energy product requires thicker T_{ox} such that $^1/_4$ of the load is attributable to oxide capacitance.

UNIT 2

15 August 2020 11:26

UNIT 2

1. Discuss the effect of different switching probabilities at different nodes in a circuit.

In order to discuss the effect of different switching probabilities at different nodes in a circuit, we will introduce the following probability factors:

α_x, probability that the node has the value x.

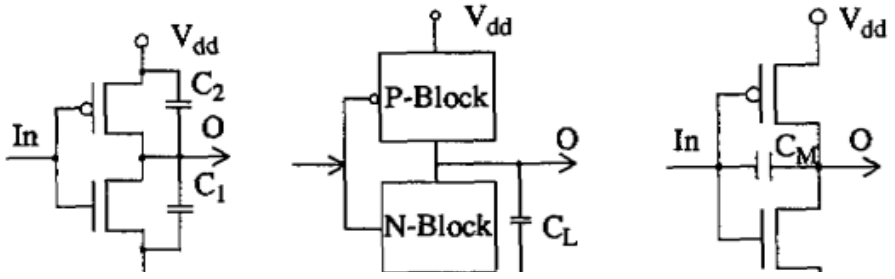
α_{xy}, probability that the node change from x to y during one clock cycle.

As an example, a clocked node has $\alpha_{01} = \alpha_{10} = 1$, telling that the node changes both from 0 to 1 and from 1 to 0 during each clock period. The corresponding power consumption expression will then include $\alpha_{01} f_c = f_c$. As another example, a data node must have equal probabilities to change from 0 to 1 and from 1 to 0. We further assume that the probability for any change in data during one clock period is α_a, called the activity of the node. We then have $\alpha_{01} = \alpha_{10} = \alpha_a/2$. The corresponding power expression will include $\alpha_{01} f_c = (\alpha_a/2) f_c$.

It is obvious that the value of the activity in different situations is crucial for the power consumption. Unfortunately, our knowledge about activity is quite limited. However, it has been shown, that the maximum activity of a random data signal is $\alpha_a = 0.5$ [30]. Simple logic circuits, driven by random signals, tends to have activities of 0.4-0.5 [30]. For more complex circuits, for example finite state machines, the activity tends to be lower.

2. Explain the effects of circuit capacitance for power consumption.

the Miller capacitance, C_M (Figure 3.1c). When the input of the inverter is made high the output becomes low. This means that the voltage over C_M changes from V_{dd} to $-V_{dd}$, its charge changed by the amount of $2C_M V_{dd}$. The charging current is taken from the input (or from the supply of the previous stage) and dumped into ground. When the input is made low instead, the same charging current flows from the supply of the present stage and is dumped via the input into the ground of the previous stage. As a result, C_M appears as an effective capacitance of $2C_M$ at both the input and the output of the stage. In Figure 3.2a C_M appears as C_{GD}.



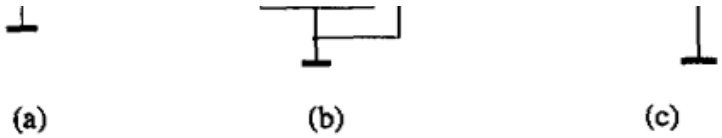


Figure 3.1 (a) Inverter Capacitances, (b) The Load Capacitance and (c) the Miller Capacitance

Other capacitances are: C_g - gate capacitances, (C_{gs}, C_{dg}, C_d) - parasitic capacitances,

C_w – wire capacitance

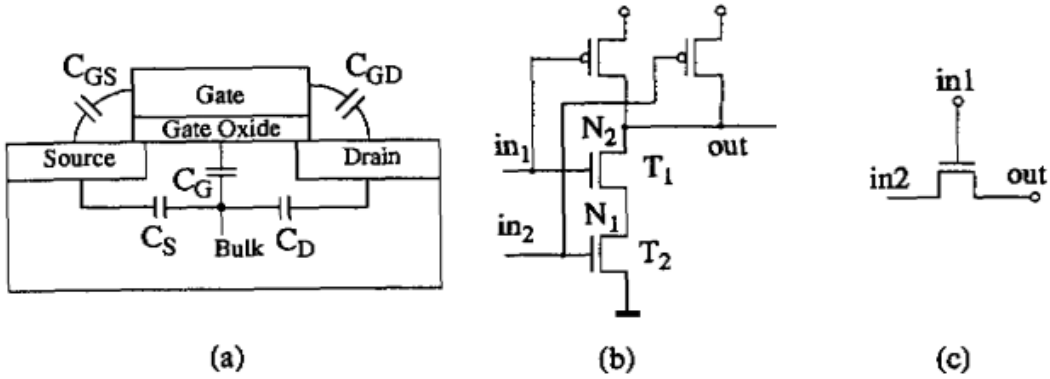


Figure 3.2 (a). Capacitances in a MOSFET (b) A CMOS Gate, (c). A Transmission Gate.

inverter case C_G+C_{GS} appears at the input of the stage and C_D at the output.

We can estimate the capacitance of the gate in figure 3.2(b) as:

$$C_L = C_w + nC_o + mC_i \qquad (3.2)$$

with $C_o = (C_D+2C_{GD})$ and $C_i = (C_G+C_{GS}+2C_{GD})$

Here n is the total number of transistors in the stage and m is the total number of load transistors. If transistors are different in sizes or have different capacitances of other reasons, the formula will be changed into a sum over all transistors involved. C_o and C_i represents effective transistor output and input

3. **Discuss the power consumption in static and dynamic flip-flops taking an example of D-flip-flop.**
4. **Discuss the power consumption in static and dynamic flip-flops taking an example of D-flip-flop using Transmission gates.**

STATIC FLIP-FLOP:

ure 3.6a we show a positive edge-triggered static flip-flop. It is designed from 4 3-input gates and two inverters, and therefore includes 28 transistors. By assuming a data activity of α_a, we may estimate the dynamic power consumption of this

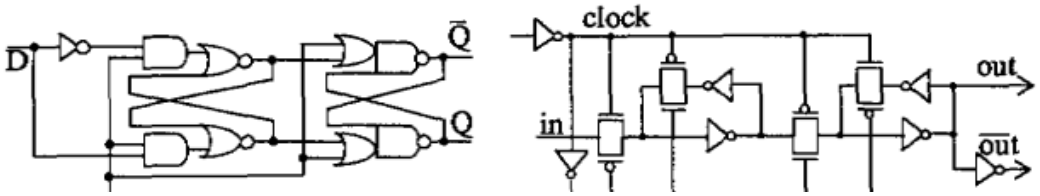


Figure 3.6 (a) A static D flip-flop made from gates,
(b) A static D flip-flop made from transmission gates.

flip-flop according to the following. We have 4 clocked inputs and 1 inverter, with a total of 10 transistor input capacitances and 2 output capacitances. Furthermore, we have two And-Or-Inv, and two Or-And-Inv gates, each gate including 6 transistors, and an inverter. There are 10 inputs driven by data signals, each with 2 transistors and also one inverter output (with 2 connected transistors) and 4 gate outputs, each with 6 transistors connected. Thus totally:

$$P_d = 10f_cC_iV_{dd}^2 + 2f_cC_oV_{dd}^2 + (10+2)(\alpha_a/2)f_cC_iV_{dd}^2 + 2(\alpha_a/2)f_cC_oV_{dd}^2 + 24(\alpha_a/2)f_cC_oV_{dd}^2$$

or

$$P_d = [10C_i + 2C_o + 12(\alpha_a/2)C_i + 26(\alpha_a/2)C_o]\, f_cV_{dd}^2 \qquad (3.14)$$

In Figure 3.6b we have a positive edge-triggered static flip-flop based on transmission gates. This circuit is made from 7 inverters and 4 transmission gates, that is a total of 22 transistors. Following the same procedure as above, we may estimate the power consumption by noting that 12 transistor gates are clocked, 2 inverter outputs are clocked, and 5 inverters and 4 transmission gates have a data signal on their inputs and outputs. This gives a total power consumption of:

$$P_d = [12C_i + 4C_o + 10(\alpha_a/2)C_i + 26(\alpha_a/2)C_o]f_cV_{dd}^2 \qquad (3.15)$$

where we have treated transmission gate inputs and outputs as inverter outputs from a capacitance of view (this may be a slight overestimation as the transmission gate do not have a Miller effect, see section 3.2.3.).

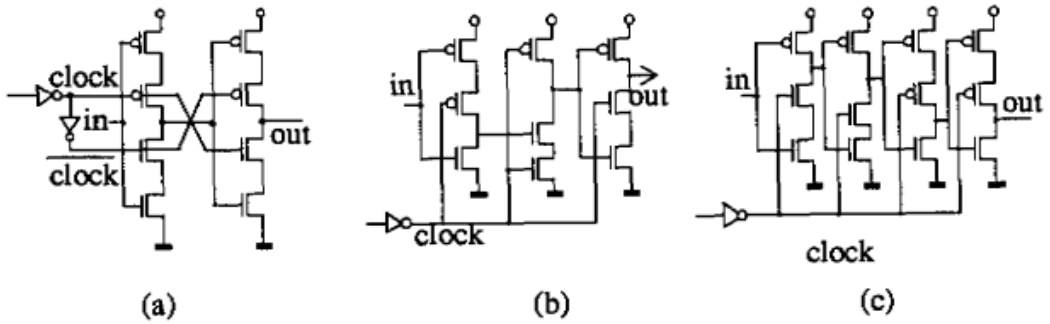


Figure 3.7 Dynamic MS D flip-flops, (a) C^2MOS,
(b) TSPC, (c) Non-precharged TSPC

each flip-flop. We will treat 3 types below. We start with an edge-triggered flip-flop made from two C^2MOS latches, Figure 3.7a. This circuit contains 12 transistors, of which 8 are clocked at their gates and 4 at their outputs. Furthermore there are 4 transistors with data on their inputs and 8 transistors with data on their outputs. From this we may estimate the power consumption as:

$$P_d = (8C_i + 4C_o + 4(\alpha_a/2)C_i + 8(\alpha_a/2)C_o)f_c V_{dd}^2 \quad (3.16)$$

In Figure 3.7b we have a precharged TSPC (true single phase clock) flip-flop containing 11 transistors, of which 6 are clocked on their inputs, 2 are clocked on their outputs, there are 3 transistors with simple data on their inputs, 2 transistor gates connected to a precharged node, 2 data nodes with 3 transistors and 1 precharged node with 3 transistors. Totally we estimate the power consumption to:

$$P_d = (6C_i + 2C_o + 3(\alpha_a/2)C_i + 6(\alpha_a/2)C_o + 2(1/2)C_i + 3(1/2)C_o)f_c V_{dd}^2 \quad (3.17)$$

We may also create an edge-triggered flip-flop from two non-precharged TSPC latches, see Figure 3.7c. In this case we have 14 transistors, of which 6 are clocked on their inputs, 2 are clocked on their outputs and the others have data signals. From this we get:

$$P_d = (6C_i + 2C_o + 8(\alpha_a/2)C_i + 12(\alpha_a/2)C_o)f_c V_{dd}^2 \quad (3.18)$$

5. Explain the possibilities to reduce power consumption for large capacitive loads.

DRIVERS FOR LARGE LOAD:

The standard method to drive large loads is to use a tapered inverter chain, Figure 3.14a [32], [14]. Let us study such a chain loaded by C_L and using N stages and a tapering factor of f. f is given by $f^N = Y = C_L/2C_i$. For such a chain it can be shown that there exist an optimum value of f for which the total delay of the chain is minimum [14]. For $C_o = C_i$ (as before), this optimum will occur at f = 3.5. The delay is however a relatively flat function of f, so also larger values of f may be acceptable.

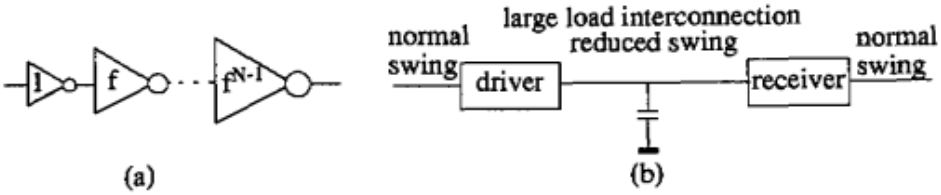


Figure 3.14 Driving the large load (a) A driver chain (b) Off chip driving

Let us study the power consumption of the inverter chain. The total switched capacitance of the chain can be written [14],[6]:

$$C_{driver} = 2(C_i+C_o)(Y-1)/(f-1) \quad (3.26)$$

or, if we instead calculate the total driver capacitance divided by the load capacitance:

$$C_{driver}/C_L = (1+C_o/C_i)(1-1/Y)/(f-1) \quad (3.27)$$

This expresses the power consumption of the driver chain in terms of the power consumption in the load. For large values of C_L, we may neglect 1/Y. Let

us furthermore again let $C_o = C_i$. Using the above optimum value of f=3.5 leads to a power consumption of the driver which is 80% of the power consumption in the

LOW SWING VOLTAGE:

An efficient way to save power is to reduce the voltage swing on high capacitance nodes, see Figure 3.14b. Here we depict an interconnection, at which we first convert the signal to a low swing and then restore it to full swing again. In this scheme we save power in two ways. First, the charge needed for charge/discharge C_L is lowered. Second, as the current to be delivered by the driver to charge/discharge C_L in a certain time is lowered, the driver size can be reduced, so the driver itself will consume less power. The problems are to produce a small swing with small static power consumption and to amplify the signal with a small power consumption.

Another straight-forward approach is to use a simple n-MOS driver with a resistive load (as in Figure 3.5b). Even simpler, we may size the n-transistor to give the correct current, I_0, as in Figure 3.15b. The drawback with this circuit is that it consumes static power (it can however be put in a standby mode just by setting the output to high).

6. Discuss low power digital cell library with respect to cell sizes and spacing.

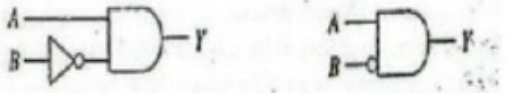
Low Power Digital Cell Library

- Over the years, the major VLSI design focus has shifted from masks, to transistors, to gates and to register transfer level
- Undoubtedly, the quality of gate level circuit synthesize depends on the quality of the cell library
- Cell Sizes and Spacing
  - In the top-down cell based design methodology, the tradeoff among power, area and delay is performed by selecting the appropriate sizes of the cells
  - Therefore, the important attribute that constitute a good low power cell library is the availability of wide ranges of cell sizes for commonly used gates
  - Further, the library cell count can be reduced without too much compromise in quality is to have more size selections for gates that are commonly

used than those are less likely used.

Low Power Digital Cell Library

- Varieties of Boolean Functions
  - The lack of varieties of Boolean functions in a cell library can result in inferior circuits to be generated
  - For example if the Boolean function $Y = A\bar{B}$ were to be implemented and the inverted input cells are not available, the logic synthesis system has to use an INVERTER and an AND gate to implement the function



Inverted input cells for low power cell library

7. Explain gate reorganization with necessary logic diagram

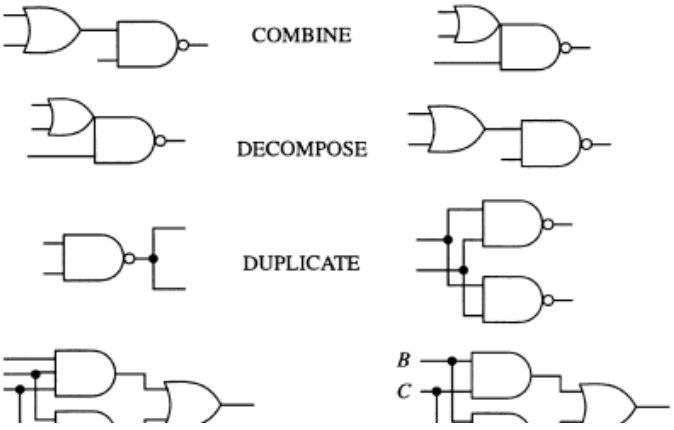
The reorganization idea is not limited to transistor networks only since the same problem exists in gate level networks. Network reorganization is applied to the gate level network to produce logically equivalent networks with different qualities for power, area and delay. Technology mapping:

– Original network is expressed in a generic form such as two input NANDgates only

The reorganized network hopefully has better power efficiency than the original network. The complexity of the gate reorganization problem limits manual solution to small circuits only. Most gate reorganization tasks are performed by automated software in the logic synthesis system.

LOCAL RESTRUCTURING: Gate reorganization is an operation to transform one logic circuit to another that is functionally equivalent. Logic restructuring techniques use local restructuring rules to transform one network to another. Some basic transformation operators are:

1. Combine several gates into a single gate
2. Decompose a single gate into several gates
3. Duplicate a gate and redistribute its output connections
4. Delete a wire
5. Add a wire
6. Eliminate unconnected gates



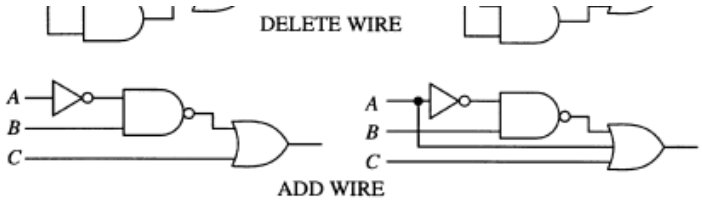


FIGURE 5.1 Local transformation operators for gate reorganization.

LOCAL RESTRUCTURING:

COMBINE operator can be used to hide high frequency nodes inside a cell so that the node capacitance is not being switched.

DECOMPOSE and DUPLICATE operators help to separate the critical path from the non-critical ones so that the latter can be sized down.

DELETE WIRE operator reduces the circuit size.

ADD WIRE operator helps to provide an intermediate circuit that may eventually lead to a better one.

8. Discuss signal gating with relevant diagrams.

Signal gating refers to a class of general techniques to mask unwanted switching activities from propagating forward, causing unnecessary power dissipation.

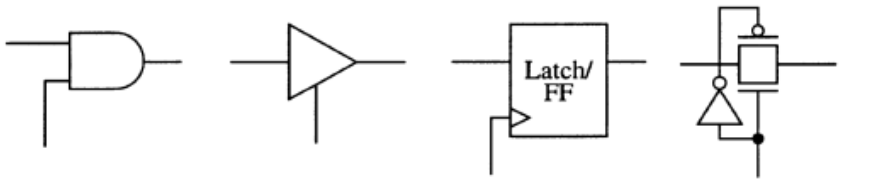
The probabilistic techniques are often used for switching activity analysis.

The simplest method to implement signal gating is to put an AND/OR gate at the signal path to stop the propagation of the signal when it needs to be masked.

Another method is to use a latch or flip flop to block the propagation of the signal.

Sometimes a transmission gate or tristate buffer can be used in place of a latch if charge leakage is not a concern.

The various logic implementation of signal gating is shown below



(a) Simple gate. (b) Tristate buffer. (c) Latch/FF. (d) Transmission gate.

FIGURE 5.2 Various logic implementations of signal gating.

The signals at the bottom of the circuits are control signals used to suppress the source signal on the left from propagating to the gated signal on the right.

9. Explain logic encoding with relevant logic table.

The logic designer of a digital circuit often has the freedom of choosing a different encoding scheme as long as the functional specification of the circuit is met. For E.g. An 8 bit counter can be implemented using the binary counting sequence or the gray code sequence. Different encoding implementation often lead to different power, area and delay tradeoff. The encoding techniques require the knowledge of signal statistics in order to make design decisions. The next slide discusses some techniques for using different logic encoding to achieve low power consumption.

BINARY VERSUS GRAY CODE COUNTING

Consider two n-bit counters implemented with Binary and Gray code counting sequences. The counting sequences of the two counters are shown below

TABLE 5.1 Binary and Gary code counting sequences.

Binary code		Gray code	
Sequence	No. toggles	Sequence	No. toggles
0 0 0	3	*0 0 0*	1
0 0 1	1	*0 0 1*	1
0 1 0	2	*0 1 1*	1
0 1 1	1	*0 1 0*	1
1 0 0	3	*1 1 0*	1
1 0 1	1	*1 1 1*	1
1 1 0	2	*1 0 1*	1
1 1 1	1	*1 0 0*	1

Toggle activities of binary versus gray code counter

TABLE 5.2 Toggle activities of Binary versus Gray code counter.

No. bits	No. of toggles		B_n / G_n
	Binary $B_n = 2(2^n - 1)$	Gray $G_n = 2^n$	
1	2	2	1
2	6	4	1.5
3	14	8	1.75
4	30	16	1.88
5	62	32	1.94
6	126	64	1.99
∞	-	-	2.00

Toggle activities of binary versus gray code counter

When n is large, the Binary counter has twice as many transitions as the Gray counter. Since the power dissipation is related to toggle activities, a Gray counter is generally more power efficient than a Binary counter.

BUS INVERT ENCODING

Bus invert encoding is a low power encoding technique that is suitable for a set of parallel synchronous signals e.g.: off-chip busses. The architecture of bus invert encoding is illustrated below

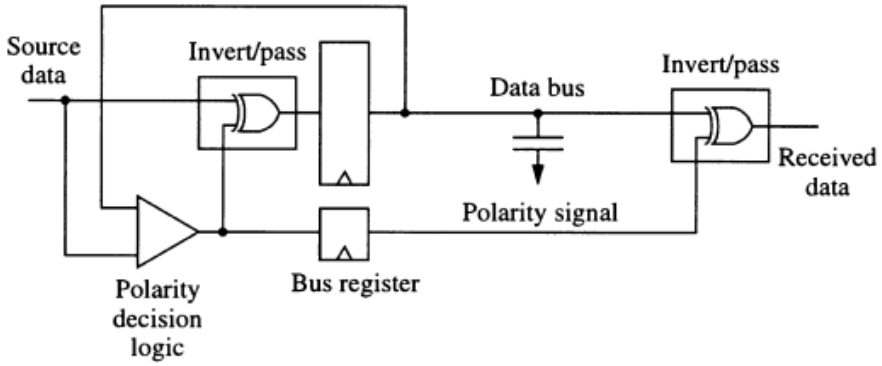


FIGURE 5.3 Architecture of bus invert encoding.

At each clock cycle, the data sender examines the current and next values of the bus and decides whether sending the true or the compliment signal leads to fewer toggles. Since the data signals on the bus may be complemented, an additional polarity signals is sent to the bus receiver to decode the bus data properly. The assertion of the polarity signal tells the receiver to invert the received bus signals.

TABLE 5.3 Efficiency of bus invert encoding under uniform random signal.

	Regular bus	Invert bus	Invert / Regular

Num. bits	Regular bus $E[P]$	Invert bus $E[Q]$	Invert / Regular $E[Q]/E[P]$
2	1	0.75	0.75
4	2	1.56	0.781
8	4	3.27	0.817
16	8	6.83	0.854
32	16	14.19	0.886
64	32	29.27	0.915
128	64	59.96	0.937
256	128	122.1	0.954
∞	-	-	1.00

10. Discuss state machine encoding techniques to reduce low power consumption.

A state machine is an abstract computation model that can be readily implemented using Boolean logic and flip flops as shown below:

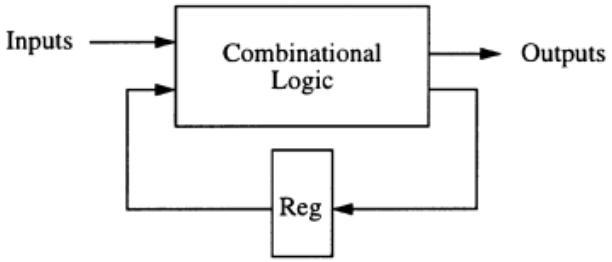


FIGURE 5.4 Hardware architecture of a synchronous state machine.

In today's logic synthesis environment, a state transition graph is specified by the designer and the synthesis system will produce a gate level circuit based on the machines specification. The state transition graph is a functional description of a machine specifying the inputs and outputs of the machine under a particular state and its transition to the next state. The very first step of a state machine synthesis process is to allocate the state register and assign binary codes to represent the symbolic states. This process is called the encoding of a state machine. The encoding of a state machine is one of the most important factors that determine the quality (area, power, speed etc.) of the gate level circuit. Transition Analysis of State Encoding

– The key parameter to the power efficiency of state encoding is the expected number of bit transitions E[M] in the state register.

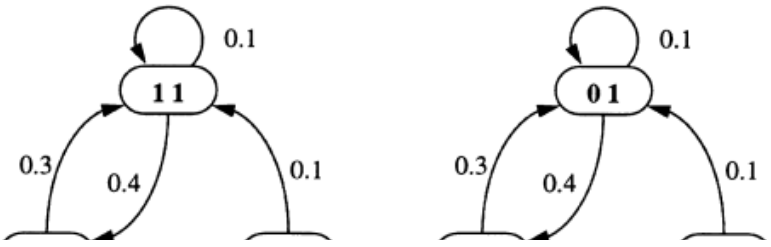
– Another parameter is the expected number of transitions of output signals.

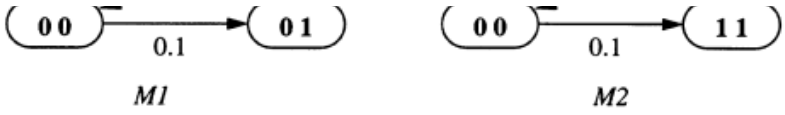
In general machines with lower E[M] are more power efficient because

– Fewer transitions of the state register lead to low power dissipation and

– Fewer transitions are propagated into the combinational logic of the machine

Consider two functionality identical state machines M1 and M2 with different encoding as shown below





Expected state transitions:

$$E[M1] = 2(0.3 + 0.4) + 1(0.1 + 0.1) = 1.6$$

$$E[M2] = 1(0.3 + 0.4 + 0.1) + 2(0.1) = 1.0$$

FIGURE 5.5 Functionally identical state machines with different encoding.

The binary codes in the state bubbles indicate the state encoding. The labels at the state transition edges represent the probabilities that transition will occur at any given clock cycle. The sum of all edge probabilities equals to unity. The expected number of state bit transitions E[M] is given by the sum of products of edge probabilities and their associated number of bit flips as dictated by the encoding. However, a state encoding with the lowest E[M] may not be the one that results in the lowest overall power dissipation. The reason is that the particular encoding may require more gates in the combinational logic, resulting in more signal transitions and power. The synthesized area and power dissipation of some randomly encoded state machines is shown below

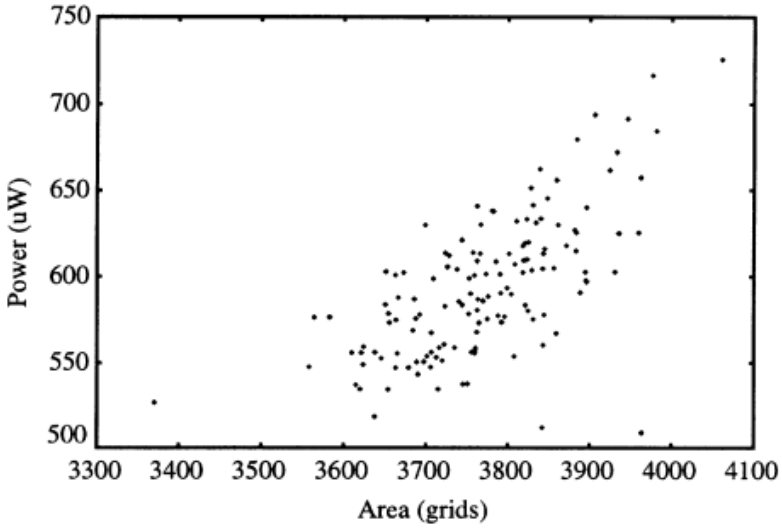


FIGURE 5.6 Effect of state encoding on synthesized area and power dissipation.

UNIT 3

15 August 2020 11:28

1. **Explain the characterization of continuous and discrete random logic Signals. [or]**

   Discuss the characterization of logic signals

   A logic signal only consists a waveform with zero-one voltage levels. The most precise way to describe a logic signal is to record all transitions of the signal at the exact times the transitions occur. To represent the signal, we write down the initial state of the signal and the time value when each transition occurs. Such description of the logic waveform allows us to analyze the signals for various purposes. For example, if we wish to compute the frequency of the signal, we count how many times the signal changes state and divide the number by the observation period. This exact characterization of the signal gives us the full details of the signal history, allowing precise reconstruction of the signal.

   For some purposes, the exact characterization of the signal is too cumbersome, inefficient and results in too much computation resource. For example, if we only wish to know the frequency of the signal, there is no need to know the initial state and the exact switching times; the number of switches should be sufficient.
   Digital signal analysis: Instead of using the exact historical time domain signal representation, we observe certain characteristic quantities of the signal over a period and use the quantities for power dissipation computation and other analysis. Often, such signal representation allows us to characterize a large number of different signals into a single class. For example, there are infinite number of logic signals with frequency I MHz
   But for the purpose of computing power dissipation using $P = CV^2f$ equation, all such logic signals result in identical power dissipation.
   Many quantities have been proposed and used for this purpose: signal probabilities, correlations, probability density functions, etc.

2. **Explain the expected frequency and static probabilities of discrete random logic signals.**

   The *static probability* of a digital signal is the ratio of the time it spends in logic 1 (t_1) to the total observation time $t_0 + t_1$ expressed in a probability value between zero and one, i.e.,

   $$p = \frac{t_1}{t_0 + t_1} \tag{3.3}$$

   This is a very fundamental characteristic quantity of a logic signal. It states how often the signal stays in logic 1 as opposed to logic 0. Again, when we refer to the static probability of a signal, there is also an implied period of observation. Often the period is infinity if it is not explicitly specified. By definition, the probability that the signal is at logic 0 is $1 - p$.

   The static probability and the frequency of a digital signal are related. If the static probability is zero or one, the frequency of the signal has to be zero because if the signal makes a transition, the ratio of logic 1 to logic 0 has to be strictly between zero and one.

   Expected frequency:

   solve the problem like this: Let's select a state at random. The probability that the state is logic 1 is $p^1 = p$ and the probability that it is logic 0 is $p^0 = (1 - p)$. Suppose that the state is logic 1, the *conditional probability* that the next state is also logic 1 is $p^{11} = p$ and the conditional probability that the next state is logic 0 is $p^{10} = (1 - p)$. This is a direct consequence of our memoryless assumption. Simi-

larly, we can compute p^{01}, p^{00} and summarize the results as follows:

$$p^0 = p^{00} = p^{10} = (1-p)$$

$$p^1 = p^{01} = p^{11} = p \tag{3.4}$$

The probability T that a transition occurs at a clock boundary is the probability of a zero-to-one transition T^{01} plus the probability of a one-to-zero transition T^{10}

$$T = T^{01} + T^{10} = p^0 p^{01} + p^1 p^{10} = 2p(1-p) \tag{3.5}$$

The expected frequency f is half the transition probability and we have

$$f = p(1-p) \tag{3.6}$$

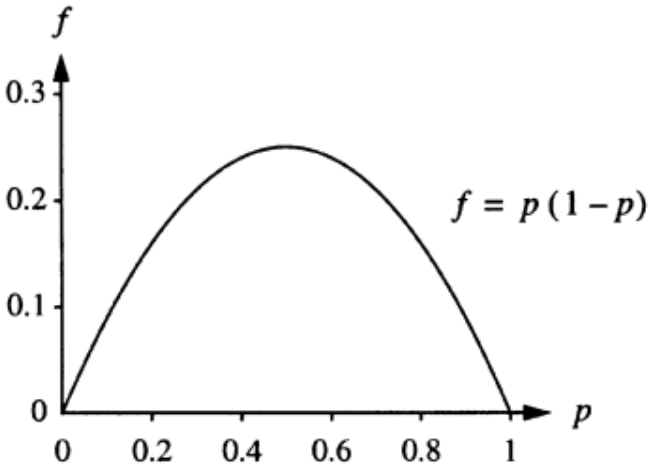


FIGURE 3.3 Expected frequency and static probability of discrete random signals.

3. **Discuss the conditional probability and frequency with necessary equations.**
In the memory less zero-one signal model, the current state of a logic signal is independent of its previous states. Relax the memory less condition such that the current state depends on the state immediately preceding it. Instead of having one independent variable to characterize the logic signal as in the memory less case, we now have two independent variables

We define p^{01} (p^{11}) to be the condition probabilities that the current state will be logic 1, given that the previous state was logic 0 (logic 1). Similarly, we define p^{00} (p^{10}) to be the condition probability that the current state is logic 0 given that the previous state was logic 0 (logic 1). The four variables are not independent but related by the following equations

$$p^{01} + p^{00} = 1 \tag{3.8}$$

$$p^{11} + p^{10} = 1 \tag{3.9}$$

as the probability that logic 0 is observed. The static probability $p^1(t)$ of the current state t is dependent on the static probability of the previous state $p^1(t-1)$ by

$$p^1(t) = p^0(t-1)\, p^{01} + p^1(t-1)\, p^{11} \tag{3.11}$$

When the zero-one sequence is time homogeneous (i.e., independent of the choice of time origin), we have $p^1(t) = p^1(t-1)$ and $p^0(t) = p^0(t-1)$. We can drop the time dependency of the static probability and rewrite Equation (3.11) as

$$p^1 = (1-p^1)\, p^{01} + p^1 p^{11} \tag{3.12}$$

which is equivalent to

$$p^1 = \frac{p^{01}}{1 - p^{11} + p^{01}} \tag{3.13}$$

Substituting Equation (3.9) into (3.13), we have

$$p^1 = \frac{p^{01}}{p^{10} + p^{01}} \tag{3.14}$$

The equation relates the static probability of the signal to the condition probabilities of transition. When the condition probabilities are given, the static probability of the random signal is determined.

Since p^{01} and p^{10} completely characterize the signal, the transition probability (and thus frequency) of the signal can be expressed by the two variables.

$$T = p^0 p^{01} + p^1 p^{10} = \frac{2p^{10}p^{01}}{p^{10} + p^{01}} \tag{3.15}$$

$$f = \frac{T}{2} = \frac{p^{10}p^{01}}{p^{10} + p^{01}} = p^1 p^{10} = p^0 p^{01} \tag{3.16}$$

The maximum frequency $f = 0.5$ is achieved when $p^{10} = p^{01} = 1$, which corresponds to a sequence of alternating 0's and 1's. When $p^{01} = p^{11} = p^1$, we have $p^{10} = p^{00} = p^0 = (1 - p^1)$ and the signal is reduced to the memoryless random signal discussed in the previous section.

4. Explain the probabilistic power analysis techniques with logic diagram

From the primary inputs, statistical quantities are propagated to the internal nodes and outputs of the circuit. The propagation of the statistical quantities is done according to a probabilistic signal propagation model.

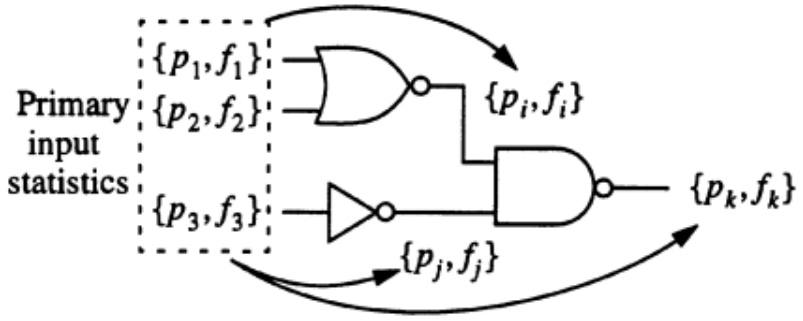
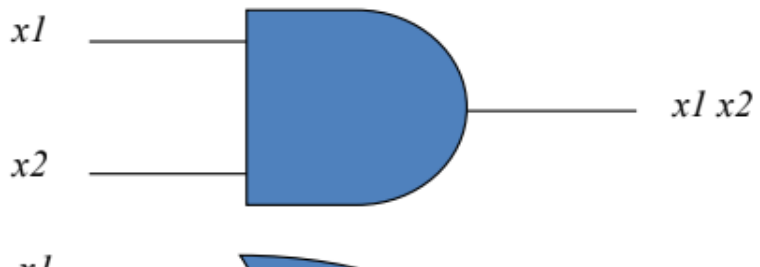


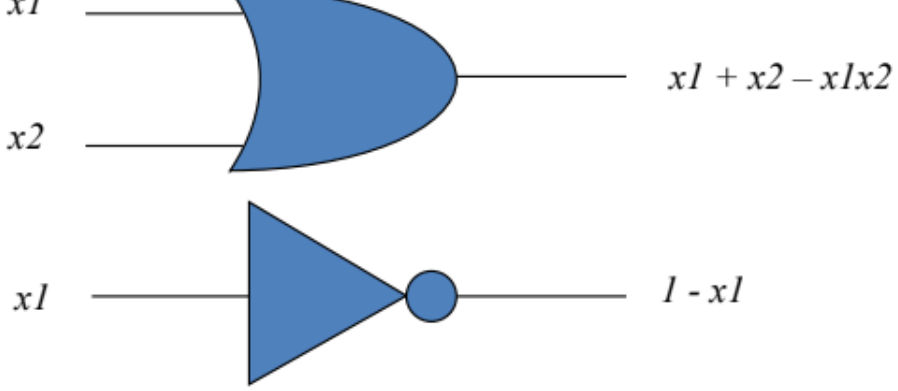
FIGURE 3.4 Propagation of statistical quantities in probabilistic power analysis.

Power can be estimated if transition density is known for all signals. Calculation of transition density requires

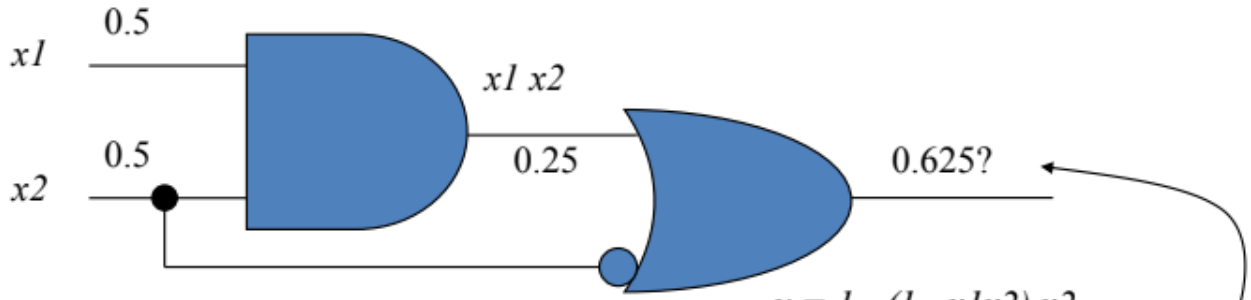
– Signal probabilities

– Transition densities for primary inputs; computed from vector statistics





Correlated Signal Probabilities



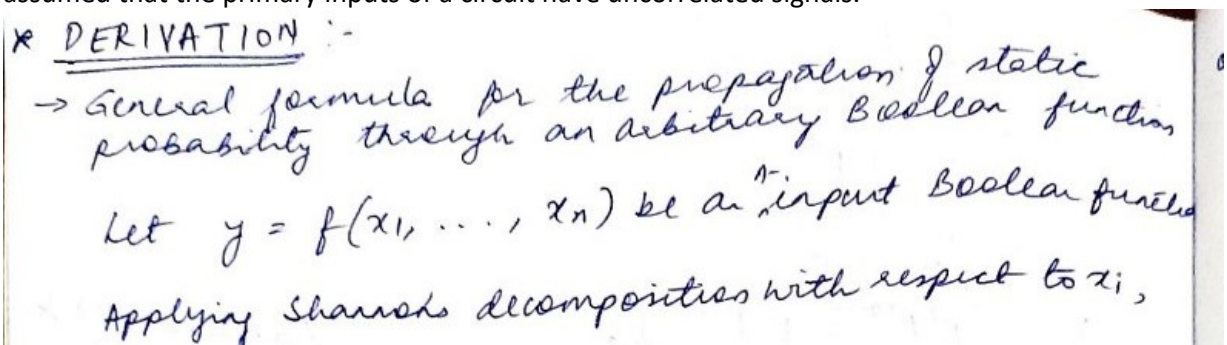
$y = 1 - (1 - x1x2)\, x2$
$= 1 - x2 + x1x2x2$
$= 1 - x2 + x1x2$
$= 0.75$

X1	X2	Y
0	0	1
0	1	0
1	0	1
1	1	1

5. Discuss the propagation of statical quantitates in probabilistic power analysis techniques

Use of this technique: We can find a propagation model for static probability, and can use it to derive the frequency of each node of a circuit, resulting in an efficient power analysis algorithm.

Static probability of a signal is the probability that the signal is at logic 1 at any given time. The propagation of static probability through a logic gate is quite simple. Consider a two-input AND gate. If the static probabilities of the inputs are PI and P2 respectively and the two signals are statistically uncorrelated, the output static probability is PIP2 because the AND-gate sends out a logic 1 if and only if its inputs are at logic 1. The uncorrelated input signal requirement is crucial for the correctness of this signal propagation model. This independence condition may not be satisfied for all logic gates. However, it is generally assumed that the primary inputs of a circuit have uncorrelated signals.


* DERIVATION :-

→ General formula for the propagation of static probability through an arbitrary Boolean function

Let $y = f(x_1, \ldots, x_n)$ be an n-input Boolean function

Applying Shannon decomposition with respect to x_i,

we have

$$y = x_i f_{xi} + \overline{x_i} f_{\bar{x}i} \quad \text{——} (1)$$

in which;

$f_{x_i}(f_{\bar{x}i})$ - is the new Boolean function obtained by setting $x_i = 1 (x_i = 0)$ in $f(x_1, \ldots, x_n)$

Let the static probabilities of the input variables be $P(x_1), \ldots, P(x_n)$

Since the 2 cum terms in the decomposition cannot be at logic 1 simultaneously, they are mutually exclusive. We can simply add their probabilities.

$$P(y) = P(x_i f_{xi}) + P(\bar{x}_i f_{\bar{x}i}) = P(x_i)P(f_{xi}) + P(\bar{x}_i)P(f_{\bar{x}i}) \quad \text{——} (2)$$

The new Boolean fns f_{x_i} and $f_{\bar{x}i}$ do not contain the variable x_i. The probabilities $P(f_{x_i})$ and $P(f_{\bar{x}i})$ are computed from recursive application of Shannon's decomposition to the new Boolean functions. At the end of recursion, $P(y)$ will be expressed as an arithmetic function of the input probabilities $P(x_i)$. Note that

$$P(\bar{x}_i) = 1 - P(x_i)$$

6. Compute the transition density and static probability of y=ab+c given P(a)=0.2, P(b)=0.3,P(c)=0.4,D(a)=1, D(b)=2,D(c)=3.

$$\begin{aligned} P(y) &= P(a)P(b) + P(c) - P(a)P(b)P(c) \\ &= 0.2 \times 0.3 + 0.4 - 0.2 \times 0.3 \times 0.4 \\ &= 0.436 \end{aligned} \quad \textbf{(3.31)}$$

To compute the transition density, we first find the Boolean difference functions with respect to the inputs

$$\begin{aligned} \frac{dy}{da} &= (b + c) \oplus c = b\bar{c} \\ \frac{dy}{db} &= (a + c) \oplus c = a\bar{c} \\ \frac{dy}{dc} &= 1 \oplus ab = \overline{ab} \end{aligned} \quad \textbf{(3.32)}$$

The probabilities of the Boolean difference functions are

$$P\left(\frac{dy}{da}\right) = P(b)\,[1 - P(c)] = 0.18$$

$$P\left(\frac{dy}{db}\right) = P(a)\,[1 - P(c)] = 0.12$$

$$P(\frac{dy}{dc}) = 1 - P(a)P(b) = 0.94 \quad (3.33)$$

and the transition density of the output is

$$D(y) = (0.18 \times 1) + (0.12 \times 2) + (0.94 \times 3) = 3.24 \quad (3.34)$$

UNIT 4

15 August 2020 11:28

1. **Discuss how the pipelining and parallelism techniques are used to reduce power dissipation at architecture level.**

Parallelism has been traditionally used to improve the computational throughput of high performance digital systems

Parallelism essentially trades area off for a lower operating frequency or higher throughput

The same tradeoff idea can also be applied to achieve power reduction

In a uniprocessing system, the power dissipation is given by

$$P_{uni} = CV^2f$$

Where C is the average capacitance switched and V is the operating voltage of the uniprocessing system. The power of parallel system is given by:

$$P_{par} = (2.2C)\,(0.6V)^2\,(0.5f) = 0.396\,P_{uni}$$

60% power reduction is achieved by parallelism

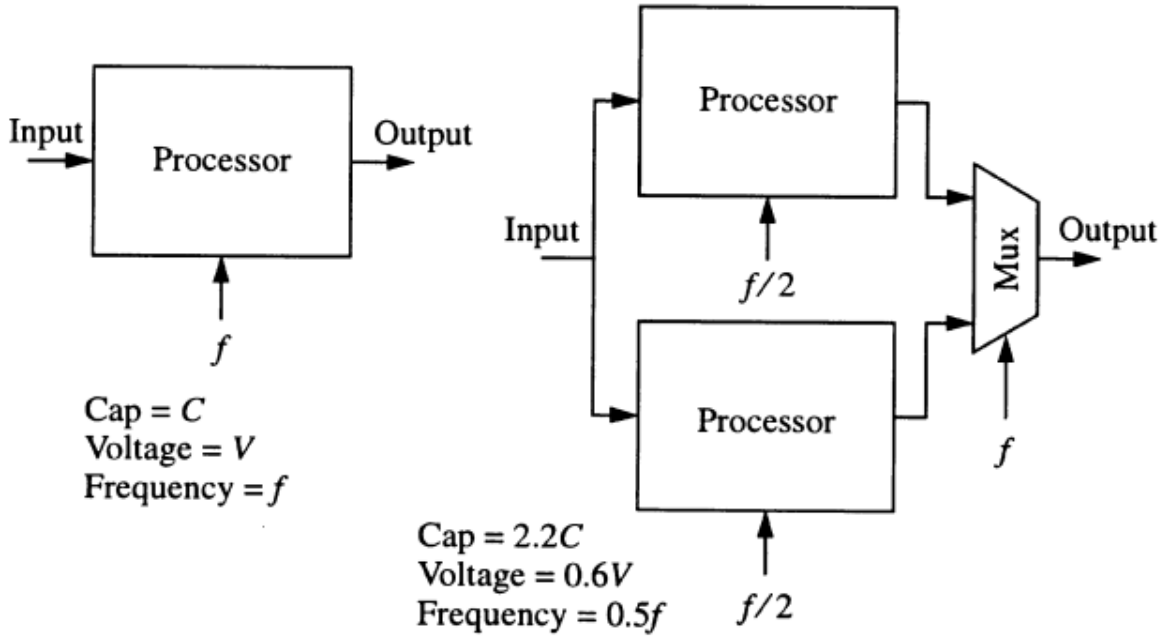
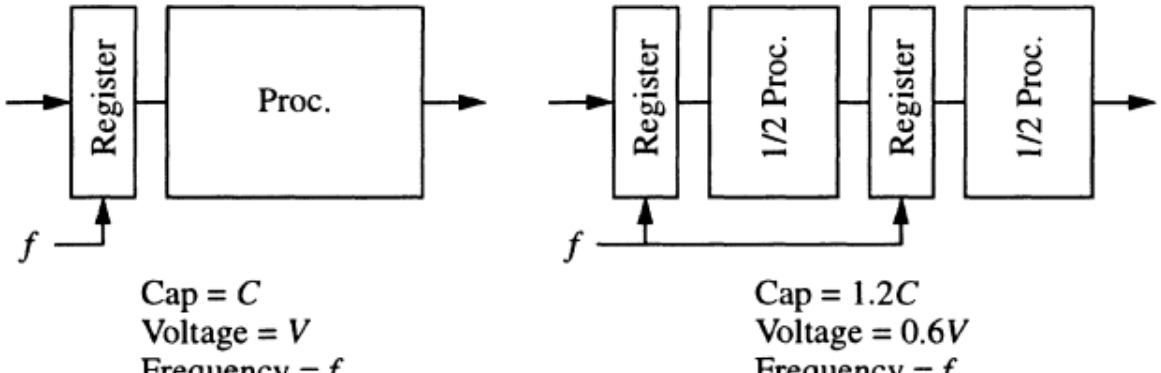


FIGURE 7.8 Power dissipation of uniprocessing and parallel processing systems.

PIPELINE: The above parallel technique increases the chip area by at least twice

If the area penalty of a parallel system is prohibitive, pipelining can offer similar tradeoff results with less area overhead but more complexity in controller design.



Frequency = f Frequency = f

FIGURE 7.9 Power efficiency of a pipelined system.

$$P_{pip} = (1.2C)(0.6V)^2 f = 0.432 P_{uni}$$

2. Explain with block diagram the adaptive performance management by voltage control.

The system consists of an asynchronous processing unit, a load sensor, a voltage regulator and input/output data buffers. The throughput of the data processing unit is dependent on the operating voltage, which is typical of an asynchronous processing system. The load sensor checks the FIFO buffer length to determine the workload of the system. If the queue is long, the voltage of the system is increased so that the throughput is accelerated. The voltage is scaled down during light loading to conserve power. This adaptive throughput adjustment technique is quite similar to the low power adaptive filtering technique

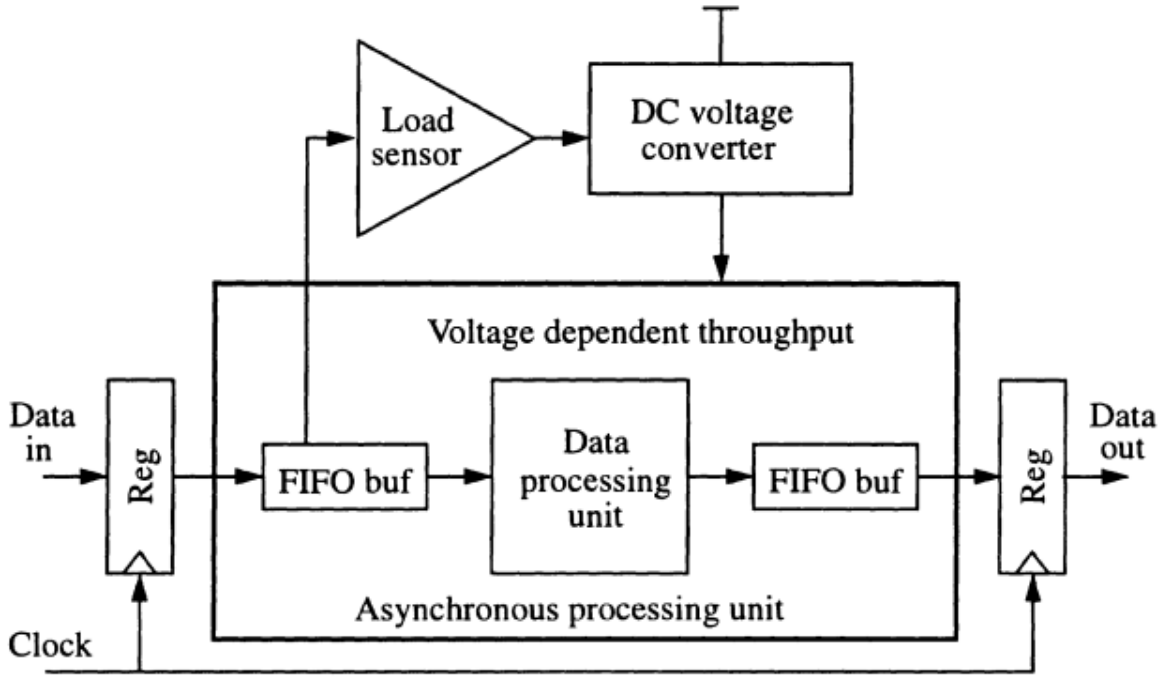


FIGURE 7.1 Adaptive performance management by voltage control.

3. Discuss switching activity reduction with block diagram

Switching activities are the biggest cause of power dissipation in most CMOS digital systems

In order to do computation, switching activities cannot be avoided

However, some switching activities do not contribute to the actual contribution and should be eliminated

The suppression of switching activities always involves some tradeoff decisions

In general hardware logic is required to suppress unwanted switching activities and the additional logic itself consumes power

- **Guarded Evaluation**

It is a technique to reduce switching activities by adding latches or blocking gates at the inputs of a combinational module if the inputs are not used.

As shown in figure below the result of multiplication may or may not be used depending on the condition selection of the multiplexer.

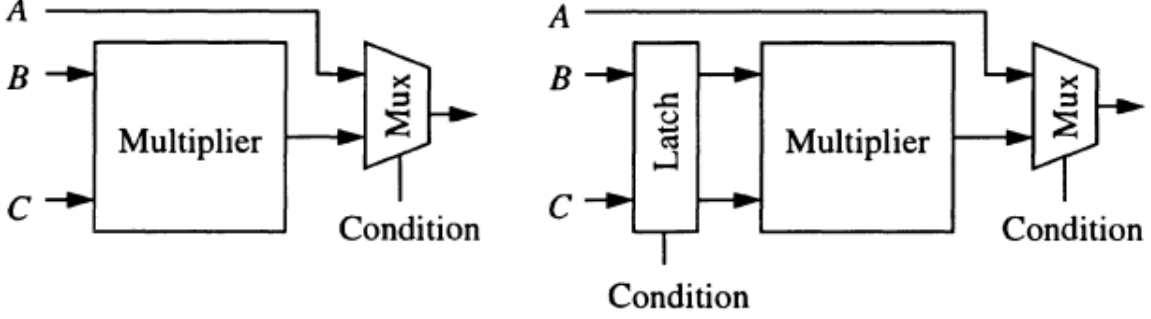


FIGURE 7.5 Guarded evaluation.

To reduce switching activities, latches are added at the input of the multiplexer

The latches are transparent when the result of multiplication is to be used

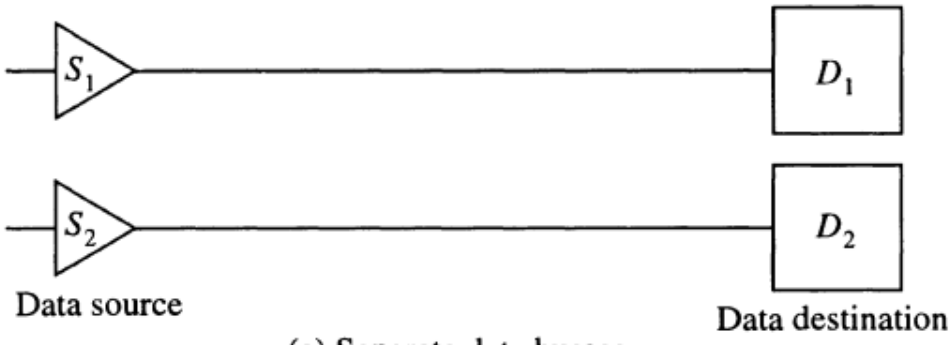
Instead of using latches, which increase the area substantially, we can also use AND gates to mask one or both inputs of the multiplier to zero

- **Bus Multiplexing**

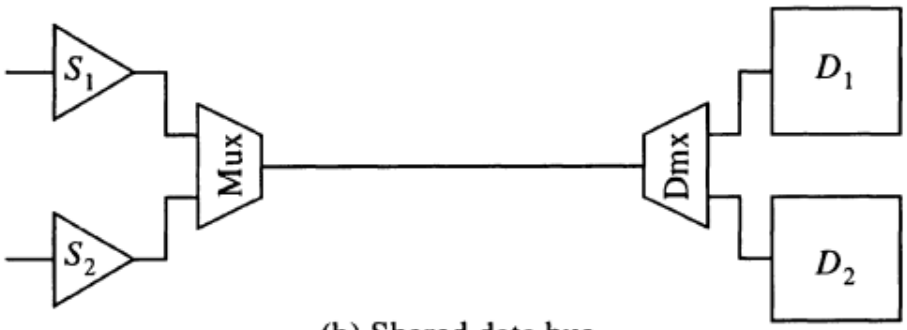
One common way to reduce hardware resources in a digital system is to share long data buses with time multiplexing.

At even clock cycles, S1 uses the shared bus to send data to destination D1 while at odd cycles, source S2 send data to D2

The switching activity characteristics of the shared and dedicated buses.

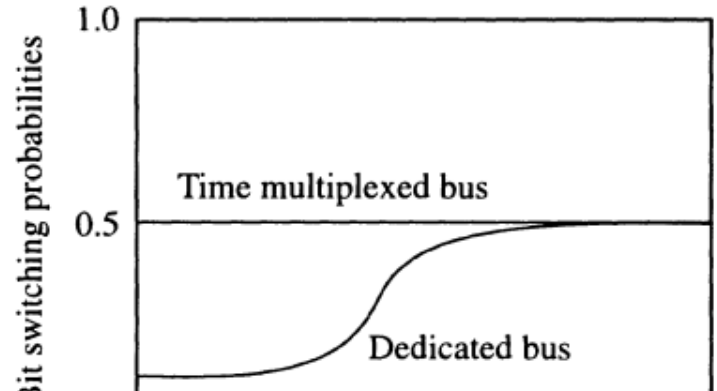


(a) Separate data busses.



(b) Shared data bus.

FIGURE 7.6 Data bus sharing with time multiplexing.



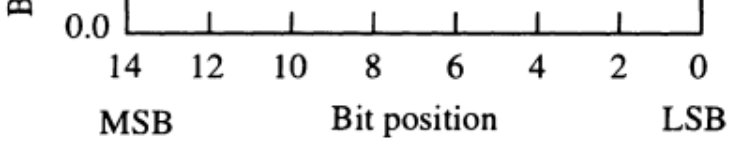


FIGURE 7.7 Switching activities of a positively correlated data stream.

4. Explain flow graph transformation with neat block diagram

Here, we focus on a system level technique for the design of special purpose architecture DSP systems

Such systems are characterized by computation intensive data path operations with simple control structures

The system architecture can be represented by a control data flow graph

The graph consists of control nodes and data nodes connected by directed edges

Control nodes change the flow of data that pass through it

Data nodes provide computation operators for the input data streams such as addition, multiplication, shift etc.

The graph edges represent the data streams of the system.

A control data graph expresses the conceptual computation

algorithm of the system

It provides basic information such as the number of control and data operators, their ordering dependencies and the inherent parallelism that may exist

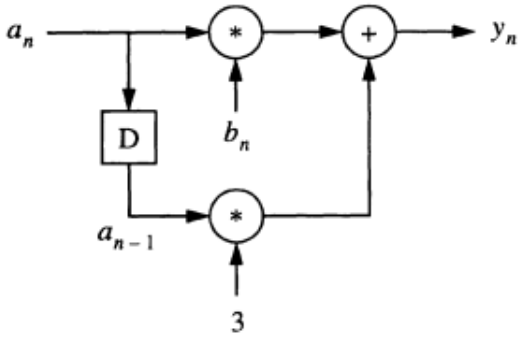
The path from system input to output is a good estimate of the delay of the system

The number of operator nodes is an estimate of the computation needs and the complexity of the system

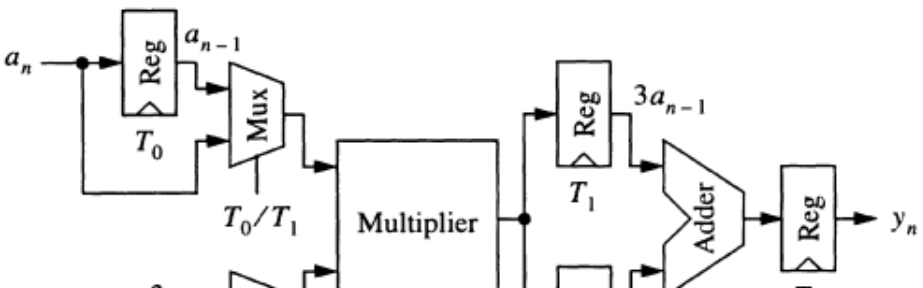
A control data flow graph is often the starting point to derive the actual hardware architecture of a system by mapping the operators and edges to actual hardware modules and busses respectively

A controller schedules the operators to perform the desired computation in the proper order

The below example illustrates a simple mapping from the control data flow graph to the system hardware architecture.



(a) Control data flow graph of $y_n = a_n b_n + 3a_{n-1}$.



(b) Hardware architecture and scheduling.

FIGURE 7.10 Control data flow graph and its mapping to hardware architecture.

- **Operator Reduction**

The transformations preserve the functionality of the graph so that the resulting graph is computationally equivalent to the original work.

The transformed graph represents an alternate hardware implementation with a different tradeoff.

If the system operating voltage is variable, the transformed graph may offer a lower voltage implementation by reducing the worst case delay path.

An example of operator transformation is illustrated below.

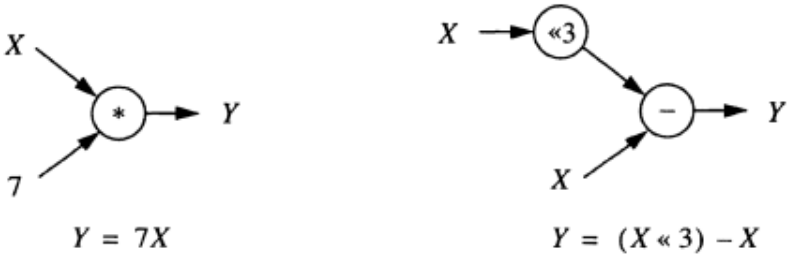


FIGURE 7.11 Constant multiplication versus shift and addition.

The multiplication by an integer constant can be replaced by binary shift (multiplication by powers of 2) and add / subtract operations, which may be power efficient

Rearranging the order of computation can also lead to fewer operations and lower power as shown below

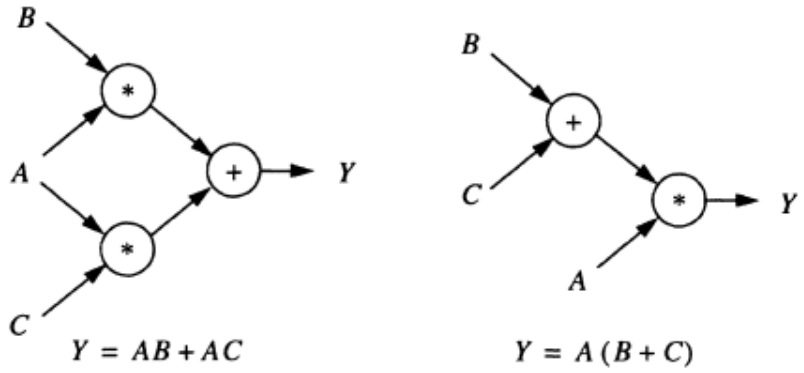
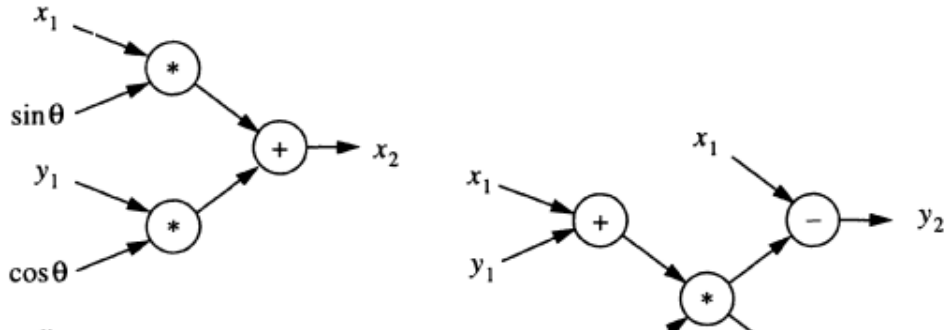


FIGURE 7.12 Associative transformation.

Also the direct and transformed flow graphs of coordinate rotation operation is shown below



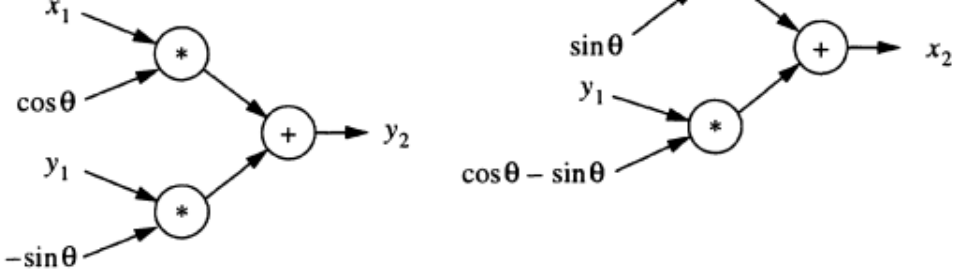


FIGURE 7.13 Direct and transformed flow graphs of coordinate rotation operation.

- **LOOP UNROLLING**

The control data flow graphs of DSP systems often contain loops, as a result of recursive computations

An important technique for flow graph transformation is to unroll the loop

Loop unrolling is a method to apply parallelism to the computation

Consider a simple recursive computation of an IIR filter

$$y_n = b_0 x_n + a_1 y_{n-1}$$

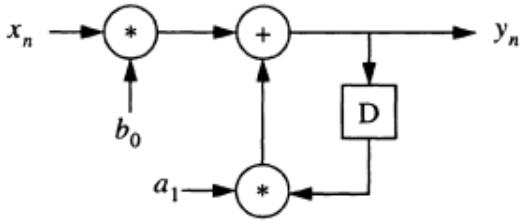
We can unroll the loop once and obtain

$$y_n = b_0 x_n + a_1 b_0 x_{n-1} + a_1^2 y_{n-2}$$
$$y_{n-1} = b_0 x_{n-1} + a_1 y_{n-2}$$

After loop unrolling, two output values are produced using two input values in a single computation cycle

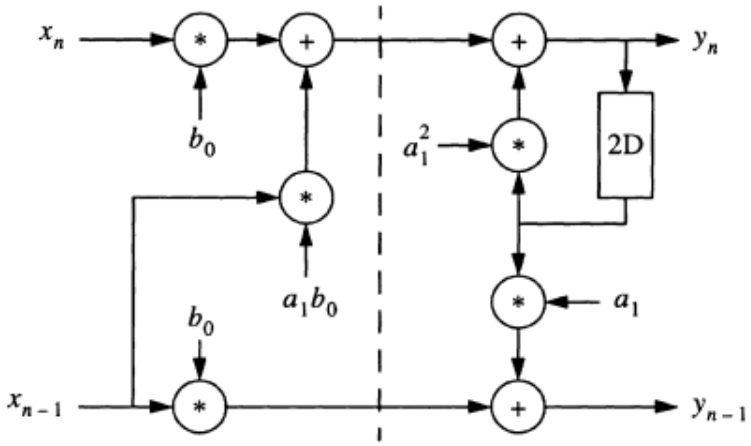
The unrolled computation structure increases the computation by more than twice

The data flow graphs of the original and unrolled computation are illustrated below



$$y_n = b_0 x_n + a_1 y_{n-1}$$

(a) Original control data flow graphs.



$$y_n = b_0 x_n + a_1 b_0 x_{n-1} + a_1^2 y_{n-2}$$
$$y_{n-1} = b_0 x_{n-1} + a_1 y_{n-2}$$

(b) Unrolled control data flow graph.

FIGURE 7.14 Loop unrolling transformation

FIGURE 7.14 Loop unrolling transformation.

If the unrolled computation structure is implemented directly, the power efficiency should be worse than the original implementation because of the increased computation

However, the unrolled structure allows us to apply pipelining by adding pipeline registers on the graph edges crossing the vertical dashed line

With pipelining , the longest delay path of the unrolled graph is identical to the original path (i.e. a multiplication followed by an addition)

Since the unrolled graph produces two outputs simultaneously, it can be implemented with half the operating frequency of the original path

Thus, its critical delay is identical but the operating frequency is halved

This allows us to lower the operating voltage to improve the overall system power efficiency

5. Explain with design flow and some supporting tools for proposed low power CAD frame work.

A design environment oriented towards power minimization must embody optimization and estimation tools at all levels of the design flow.

The most effective design decisions derive from choosing and optimizing algorithms at the highest levels. However, implementation details cannot be accurately modeled or estimated at this level of abstraction so relative metrics must be judiciously used in making design selections. More information is available at the architectural level, hence estimates are more accurate and the effectiveness of optimizations can be more accurately quantified.

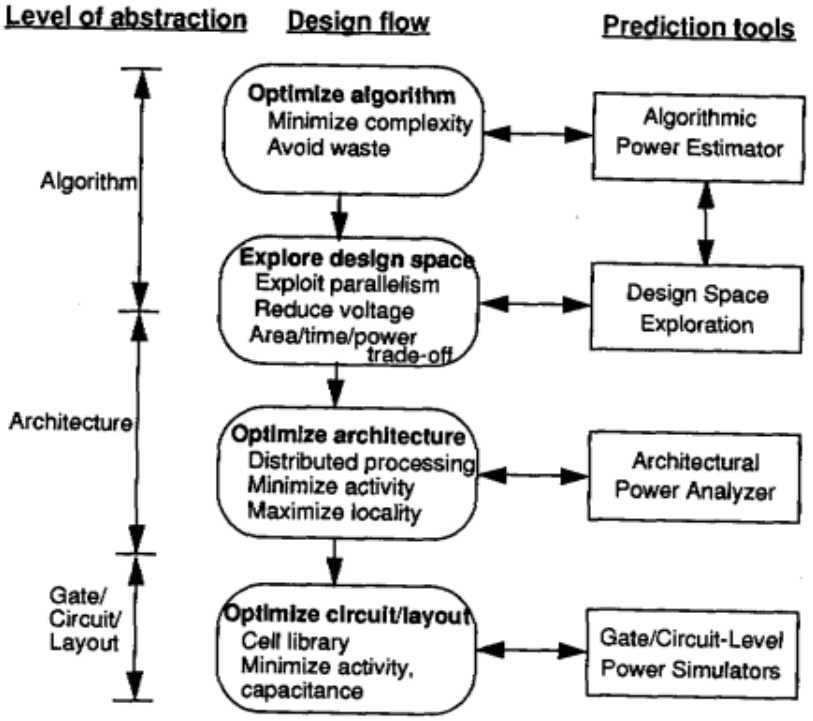


Figure 11.1 Design flow and some supporting tools for proposed low-power CAD framework

UNIT 5

15 August 2020 11:29

1. **Mention the advantage and limitations of the spice power analysis method.**

4. Explain the spice power analysis

SPICE operates by solving a large matrix of nodal current using the Krichoff's Current Law (KCL). The basic components of SPICE are the primitive elements of circuit theory such as resistors, capacitors, inductors, current sources and voltage sources. More complex device models such as diodes and transistors are constructed from the basic components. The device models are subsequently used to construct a circuit for simulation. Basic circuit parameters such as voltage, current, charge, etc. are reported by SPICE with a high degree of precision. Hence the circuit power dissipation can be directly derived from SPICE simulation.

SPICE offers several analysis modes but the most useful mode for digital IC power analysis is called *transient analysis*. The analysis involves solving the DC solution of the circuit at time zero and makes small time increments to simulate the dynamic behavior of the circuit over time. Precise waveforms of the circuit parameters can be plotted over the simulation time.

The strongest advantage of SPICE is of cause its accuracy. SPICE is perhaps the most versatile among all power analysis tools. It can be used to estimate dynamic, static and leakage power dissipation. MOS and bipolar transistor models are typically available and it also faithfully captures many low-level phenomena such as charge sharing, cross talk and transistor body effect. In addition, it can handle common circuit components such as diodes, resistors, inductors and capacitors. Specialized circuit components can often be built using the SPICE's modeling capability.

SPICE analysis requires intensive computation resources and is thus not suitable for large circuits. Most SPICE-based simulators start to experience memory or computation limitation at several hundred to several thousand devices. Some advanced SPICE simulators can handle circuits up to ten thousand devices but simulating the entire chip is not possible.

2. **Explain the effects of data correlation on bit switching frequency.**

The effect of data correlation on power dissipation often depends on the numerical representation of the digital system. In digital signal processing, the two's complement representation is the most widely used. Another popular numerical representation method is the signed magnitude. We will assume the use of two's complement representation as the analysis method for signed magnitude representation is similar.

Let us observe the toggle characteristics of the data signals under the influence of data correlation. If the data sample is positively correlated, successive data sample values are very close in their binary representation. This means that the least significant bits (LSB) of the data bus toggle frequently while the most significant bits (MSB) are relatively quiet. If we plot the bit-toggle frequencies of the signals, the characteristics shown in Figure 2.6 will be observed [2.13]. Some of the LSB bits toggle at approximately half the maximum frequency. This is called the *uniform white noise* region because the bits toggle in a random fashion. On the MSB side, the bits have a very low toggle rate and they are called the *sign bit* region. Most of the toggling at this region is the result of a sign change of the data samples, which occurs infrequently. There is also a grey area between the two regions where the toggle frequency changes from white noise to sign bit. In this region, the bit-toggle rate changes from near zero

to 0.5, typically in a linear fashion. Note that the switching frequency is normalized with respect to the maximum toggle rate, which is half of the sampling frequency. For a negatively correlated data stream, the converse is observed. The sign bit region has a very high switching frequency and the noise bit region remains at random. If a data stream exhibits no correlation, all bit switching characteristics will all look like uniform white noise.

The above observation allows us to characterize the data stream with only a few parameters:

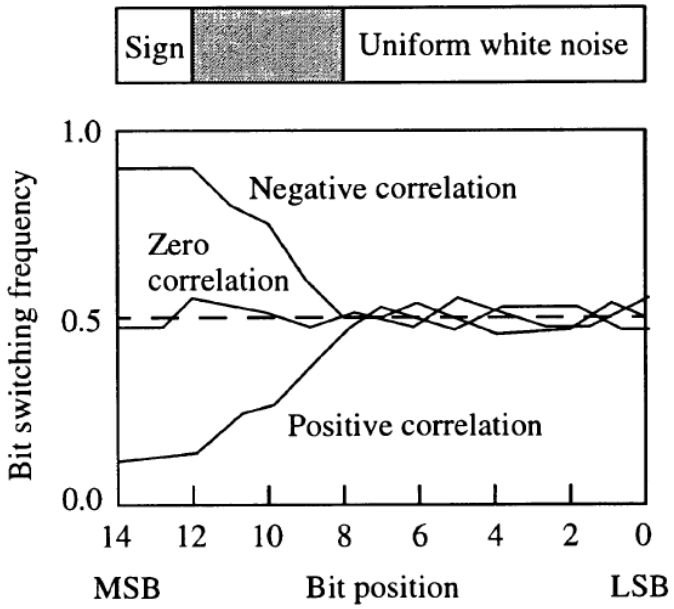


FIGURE 2.6 Effects of data correlation on bit switching frequency.

1. Sample frequency.
2. Data correlation factor from −1.0 to +1.0.
3. The sign bit and uniform white noise regions with two integers.

Such characterization of data signals is called the *dual bit type* model, proposed by Landman and Rabaey [2.13].

3. Mention the steps to obtain total power dissipation of the circuit after conducting the gate level power simulation

The event driven gate level power simulation is summarized as follows:

– Run logic simulation with a set of input vectors

– Monitor the toggle count for each net

– Obtain capacitive power dissipation P_{cap}

– Monitor the dynamic energy dissipation events of each gate

– Obtain internal switching power dissipation Pint

– Monitor the static power dissipation states of each gate

– Obtain static power dissipation P_{stat}

– Sum up all power dissipation components

$$P = P_{cap} + Pint + P_{stat}$$

4. **Discuss the gate level logic simulation with equations.**

Simulation based gate level timing analysis has been a very mature technique in today's VLSI design

The component abstraction at this level is logic gates and nets

The circuit consists of components having defined logic behavior at its input and output such as NAND gates, latches and flip flops

Gate level logic simulation software is one of the earliest CAD tools being developed

In present world, Gate level logic simulator can perform full chip simulation up to several million gates.

Event driven logic simulation: Events are zero-one logic switching of nets in a circuit at a particular simulation time point.

Cycle based simulators

Gate level simulators: Hardware acceleration, Hardware emulation.

VHDL and Verilog are two popular languages used to describe gate-level design.

CAPACITIVE POWER DISSIPATION: The major advantage of gate level power analysis is that the $P = CV^2f$ can be computed precisely

In non-logic abstraction such as SPICE, the notion of the frequency of a node is not well defined because it has an analog waveform that is potentially non-periodic and non-digital

In logic simulation, the switching activities of each node can be monitored to determine its frequency

The capacitive power dissipation of the circuit is

$$P_{cap} = \sum_{\text{net } i} C_i V^2 f_i \qquad (2.3)$$

$f_i = t_i/(2T)$ where t_i s the simulation time elapsed

INTERNAL SWITCHING ENERGY

The Equation (2.3) computes the power dissipated due to charging and discharging of node capacitance

Switching activities are not accounted

Short circuit power is also not captured

The dynamic power dissipated inside the logic cell is called internal power which consists of short circuit power and charging and discharging of internal nodes

Idea is to simulate the “dynamic energy dissipation events” of the gate with SPICE or other lower level power simulation tools

Switching from 1 to 0 or vice versa consumes some amount of dynamic energy internally

Computation of dynamic internal power uses the concept of logic events

Each gate has a pre-defined set of logic events in which a quantum of energy is consumed for each event

The energy value of each event can be computed with SPICE circuit simulation

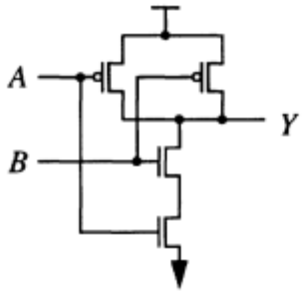
The total dynamic internal power dissipation is given by

$$P_{int} = \sum_{\text{gate } g} \sum_{\text{event } e} E(g, e) f(g, e) \qquad (2.4)$$

Where E(g, e) is the energy of the event e of gate g obtained from logic gate characterization, f(g, e) is the occurrence frequency of the event on the gate observed

from logic simulation, E(g, e) depends on process conditions, operating voltage, temperature, output loading capacitance, input signal slopes etc.

For example a simple 4 transistor NAND gate has four dynamic energy dissipation events as shown in below Fig.



(a) A 4-transistor CMOS NAND gate.

A	B	Y	Dyn energy (pJ)
1	r	f	1.67
1	f	r	1.39
r	1	f	1.94
f	1	r	1.72

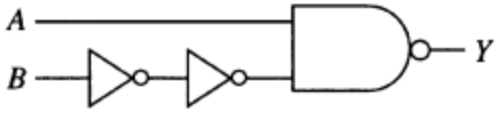
(b) Dynamic energy dissipation events.

A	B	Y	Static power (pW)
0	0	1	5.05
0	1	1	13.1
1	0	1	5.10
1	1	0	28.5

(c) Static power dissipation states.

FIGURE 2.3 Dynamic events and static states of a 2-input CMOS NAND gate.

The first implementation has only four energy dissipation events. Second implementation has two additional events due to switching of is internal nodes.



A	B	Y
1	r	f
1	f	r
r	1	f
f	1	r

A	B	Y
1	r	f
1	f	r
r	1	f
f	1	r
0	r	1
0	f	1

FIGURE 2.4 Two different implementations of NAND gate result in different dynamic energy dissipation events.

STATIC STATE POWER: A similar event characterization idea can also be used to compute the static power dissipation of a logic gate. Here, the power dissipation depends on the state of the logic gate. The total static power is

$$P_{stat} = \sum_{\text{gate } g} \sum_{\text{state } s} P(g, s) \frac{T(g, s)}{T} \qquad (2.5)$$

A	B	Y	Static power (pW)
0	0	1	5.05
0	1	1	13.1
1	0	1	5.10
1	1	0	28.5

(c) Static power dissipation states.

P(g, s) is the static power dissipation of gate "g" at state "s" obtained from characterization.

State duration T(g, s) is obtained from logic simulation: It is the total time gate :g" stays at state "s".

GATE LEVEL CAPACITANCE ESTIMATION: As discussed earlier, capacitance is the most important attribute that affects the power dissipation of CMOS circuits.

Capacitance also has an impact on delays and signal slopes of logic gates. Change in gate delay may affect the switching characteristics of the circuit and influence power dissipation. Short circuit current is affected by the input signal slopes and output capacitance loading. Thus, capacitance has a direct and indirect impact on power analysis. The accurate estimation of capacitance is important for power analysis and optimization.

Two types of parasitic capacitance exist in CMOS circuits:

– Device parasitic capacitance

– Wiring capacitance

Parasitic capacitance of MOS devices is associated with terminals. The gate capacitance depends on the oxide thickness of the gate that is process dependent. Design dependent factors are: Width, length and shape of the gate. In general a larger transistor has more capacitance in all its terminals. The second source of parasitic capacitance is wiring capacitance. Depends on the layer, area and shape of the wire.

GATE-LEVEL POWER ANALYSIS:

The event-driven gate-level power simulation is summarized as follows:

1. Run logic simulation with a set of input vectors.
2. Monitor the toggle count of each net; obtain capacitive power dissipation P_{cap} with Equation (2.3).
3. Monitor the dynamic energy dissipation events of each gate; obtain internal switching power dissipation P_{int} using Equation (2.4).
4. Monitor the static power dissipation states of each gate; obtain static power dissipation P_{stat} with Equation (2.5).
5. Sum up all power dissipation components.

The total power dissipation of the circuit is the sum of the three power components expressed in Equations (2.3), (2.4) and (2.5)

$$P = P_{cap} + P_{int} + P_{stat} \quad (2.7)$$

5. **Explain the architecture level analysis with respect to : Power models based on activities, Power models based on component operations**

Over the years, the design abstraction has moved from the mask, the transistors, to gates and now to the architecture level. Architecture level abstraction is called as block level or macro level design

Building blocks are : Registers, Adders, Multipliers, Buses, Multiplexers, Memories, State machines etc.

Today, architecture level power analysis is becoming more important because more digital circuits are now synthesized from architecture description.

Power model based on activities

One way to characterize the architectural components is to express the power dissipation as a function of the number of bits of the components and their operating frequencies

E.g.: the power dissipation of an adder can be expressed as:

$$P = (n\,K1 + K2)\,f$$

Where n – number of bits, f – frequency of the addition operation K1

and K2 – empirical coefficients derived from gate level simulation

The above model depends only on the operating frequency and size of the adder

The model does not account the data dependency of the power dissipation

A more accurate model that can capture data dependency is to characterize the power dissipation as

$$P = \sum_{input\ i} K_i f_i$$

For multipliers, the model is given by

$$P = \sum_{input\ i} K_i f_i + \sum_{input\ j} K_j f_j$$

Since it is tedious to characterize K_i for each input i, the P can be simplified as follows

$$P = K_1 \sum_{input\ i} f_i + K_2 \sum_{input\ j} f_j$$

$$P = K_1 f_{in} + K_2 f_{out}$$

Power dissipation based on Component operations

In this model the power dissipation is expressed in terms of the frequency of some primitive operations of an architecture component

• The power dissipation is given by

$$P = K_1 f_{read} + K_2 f_{write}$$

• The parameters f_{read} and f_{write} are the frequencies of READ and WRITE operations, respectively

• Coefficients K1 and K2 are obtained from characterization and properties of the component

6. Discuss the Data correlation analysis in DSP systems.

Sample correlation: Refers to the property that successive data samples are very close in their numerical values and consequently their binary representations have many bits in common.

Negative correlation (anti-correlation): Successive samples jump from a large positive value to a large negative value.

Positive or negative correlation has a significant effect on the power dissipation of a DSP system because of the switching activities on the system data path.

Hence, here we discuss how to estimate power of the architecture level component based on the frequency and correlation measures of the data stream.

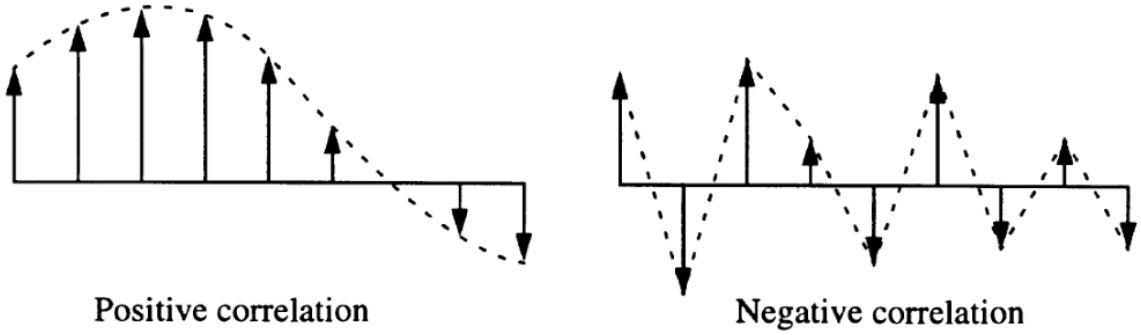


FIGURE 2.5 Correlation resulting from sampling analog signals.

Dual bit type signal model:

– The effect of data correlation on power dissipation depends on the numerical representation of the digital system

- Two's complement
- Signed magnitude

– Uniform white noise region

- Bits toggle in a random fasion

The data stream can be characterized with only a few parameters: Sample frequency, Data correlation factor from -1.0 to +1.0, The sign bit and uniform white noise regions with two integers.

Such characterization of data signals is called the dual bit type model proposed by Landman and Rabaey:

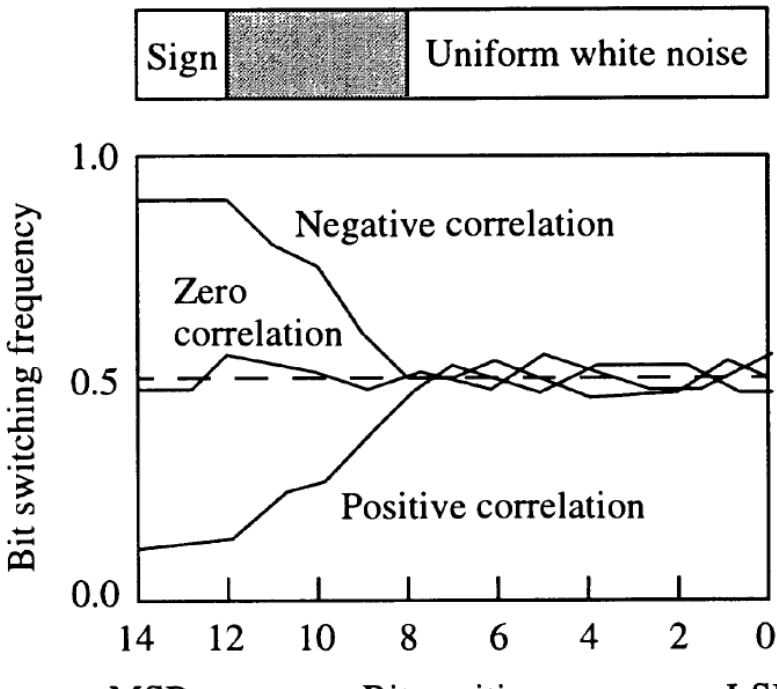


FIGURE 2.6 Effects of data correlation on bit switching frequency.

1. Distinguish between conventional charging and adiabatic charging of load capacitance

2.1. CMOS Logic Circuits Principal

Power dissipation in conventional CMOS circuits primarily occurs during device switching. As shown in Fig. 1, both PMOS and NMOS transistors can be modelled by including an ideal switch in series with a resistor in order to represent the effective channel resistance of the switch and the interconnect resistance [12].

The pull-up and pull-down networks are connected to the node capacitance C_L, which is referred to as the load capacitance in this paper.

When the logic level in the system is "1," there is a sudden flow of current through R. $Q = C_L V_{dd}$ is the charge supplied by the positive power supply rail for charging C_L to V_{dd}. Hence, the energy drawn from the power supply is $Q \cdot V_{dd} = C_L V_{dd}^2$. If it is assumed that the energy drawn from the power supply is equal to that supplied to C_L, the energy stored in C_L becomes one-half the supplied energy, i.e.

$$E_{stored} = 0.5\, C_L V_{dd}^2 \quad (1)$$

The remaining energy is dissipated in R. The same amount of energy is dissipated during discharging in the NMOS pull-down network when the logic level in the system is "0." Therefore, the total amount of energy dissipated as heat during charging and discharging is

$$\begin{aligned} E_{total} &= E_{charge} + E_{discharge} \quad (2) \\ &= 0.5 C_L V_{dd}^2 + 0.5 C_L V_{dd}^2 \\ &= C_L V_{dd}^2 \end{aligned}$$

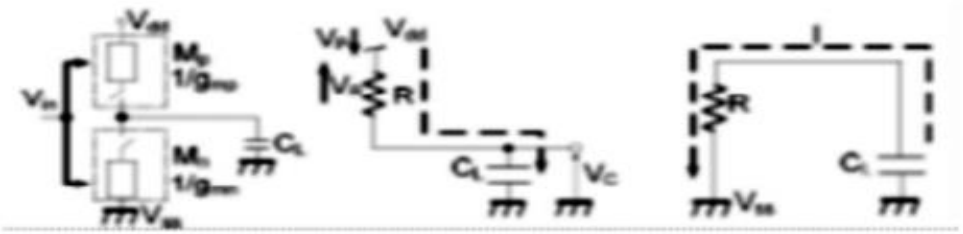


Fig. 1. A Conventional CMOS model along with charging and discharging

From the above equation, it is apparent that the energy consumption in a conventional CMOS circuit can be reduced by reducing V_{dd}. By decreasing the switching activity in the circuit, the power consumption ($P = dE/dt$) can also be proportionally suppressed.

2.2. Adiabatic Logic Circuits Principal

Switching circuit design around the adiabatic principle works in such a way that they reduces the overall power consumption by reducing the leakage current and the dissipation of power through parasitic capacitances as in the case of conventional CMOS This is accomplished by using AC power supplies to initially charge the circuit during specific adiabatic phases and then discharge the circuit to recover the supplied charge. The principle of adiabatic switching can be best explained by contrasting it with the conventional dissipative switching technique. Fig. 2 shows the manner in which energy is dissipated during a switching transition in adiabatic logic circuits.

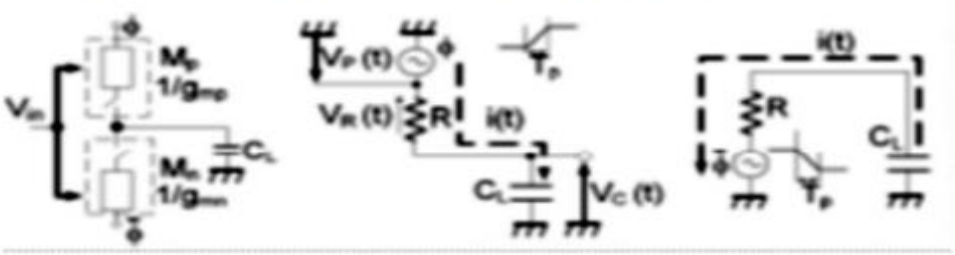


Fig. 2. An Adiabatic logic model along with charging and discharging

In contrast to conventional charging, the rate of switching transition in adiabatic circuits is decreased because of the use of a time-varying voltage source instead of a fixed voltage supply. The peak current in adiabatic circuits can be significantly reduced by ensuring uniform charge transfers over the entire available time. Hence, if Iˆ is considered as the average of the current flowing to CL, the overall energy dissipation during the transition phase can be reduced in proportion as follows [13, 14]:

In adiabatic switching circuits the parasitic capacitor charging, is done when the time for the driving voltage ö to change from zero to maximum voltage, charging time period is

long, power dissipation is nearly 0. When ö changes from 1 to 0 in the pulldown network, discharging via the nMOS transistor occurs. From Eq. (2), it is apparent that when power dissipation is minimized by decreasing the rate of switching transition, the system draws some of the energy that is stored in the capacitors during a given computation step and uses it in subsequent computations. The signal energy may be recycled instead of dissipated as heat [14]. It must be noted that systems based on the abovementioned theory of charge recovery are not necessarily reversible.

2. Explain how dynamic power dissipation is minimized using adiabatic switching.

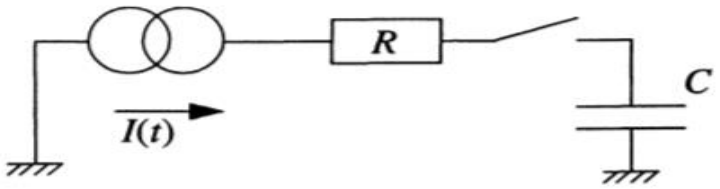


Figure 6.1: Current source charging a capacitance through a switch with a certain on-resistance.

The load capacitance is discharged at time 0. The capacitance voltage as a function of time, $V_C(t)$, is then given by:

$$V_C(t) = \frac{1}{C}\int_0^t I(\theta)d\theta = \frac{1}{C}\bar{I}(t)t \tag{170}$$

$\bar{I}(t)$ is the average current from 0 to t:

$$\bar{I}(t) = \frac{C \cdot V_C(t)}{t} \tag{171}$$

The energy dissipation in R from 0 to $t = T$ is then given by:

$$E_{diss} = R\int_0^T I(\theta)^2 d\theta \geq R\int_0^T \bar{I}(T)^2 d\theta = R\bar{I}(T)^2 T = \frac{RC}{T}CV_C(T)^2 \tag{172}$$

with equality when $I(t) \equiv \bar{I}(T)$, that is, when the current is constant. Other distributions of the current over time give higher dissipation. The influence of the current waveform may be quantified in a shape factor, ξ:

$$\xi = \frac{\int_0^T I(\theta)^2 d\theta}{\bar{I}(T)^2 T} \geq 1 \tag{173}$$

Then:

$$E_{diss} = \xi\frac{RC}{T}CV_C(T)^2 \tag{174}$$

This expression was introduced by Seitz and co-workers [Seitz85] (who apparently assumed constant-current charging and therefore omitted ξ).

The voltage on the output of the current generator in Figure 6.1, V_I, is connected to I (and thereby to V_C) by the following differential equation:

$$V_I = RI + V_C = RC\frac{d}{dt}V_C + V_C \tag{175}$$

A constant V_I (the "conventional" charging case described in Chapter 3) corresponds to an exponential current waveform with the time constant RC. A constant current corresponds to a linear voltage ramp.

3. Explain basic steps of battery aware task scheduling. How does it improves the life time of a battery

2.3. Voltage Scaling

The system configuration for a battery-operated device in a single processor system is described in figure 1. The battery voltage and current are denoted by V_{batt} and I_{batt} while the operating voltage of the processor and the current drawn by the processor are denoted by V_{proc} and I_{proc} respectively. The conversion efficiency of the DC-DC converter is $\eta = \frac{I_{proc}V_{proc}}{I_{batt}V_{batt}}$. For a $CMOS$ gate with threshold voltage V_t, supply voltage V_{dd} and velocity saturation index γ, the delay is proportional to $V_{dd}/[V_{dd} - V_t]^{\gamma}$. In our calculations, we assume γ to be 2. Assuming that the efficiency η is considered to be a constant in the region of operation, since I_{proc} is proportional to $(V_{proc})^2$, I_{proc} scales by s^2 as a result of voltage/clock scaling. Further since V_{batt} is considered constant, the battery current I_{batt} scales by s^3. Thus voltage scaling by a factor of s at the processor level leads to battery current scaling by a factor of s^3. Thus slack utilization by voltage scaling leads to significant battery lifetime improvement.

Fig. 1. System Level Configuration for a Single Processor System.

3.1. Scheduling Algorithm

A given task is associated with three parameters: the average current I_k, the average duration D_k and the start time t_k. Task scheduling with voltage scaling affects not only t_k but also I_k and D_k. Given the battery parameters α and β, the task profile and the set of discrete voltages that the processor can operate on, the scheduling algorithm provides a feasible schedule that maximizes the quality metric Q. The constraints are as follows: (1) The task deadlines are not violated, and (2) at any time the battery is alive. The assumption here is that the system can operate at any of the discrete voltages V_i from the set $S_V = \{V_0, V_1, ..., V_{m-1}\}$ as supported by the DC-DC converter.

The heuristic scheduling algorithms are based on the following three theorems derived from the properties of the battery model. The proofs of the theorems have been omitted due to lack of space.

Theorem 1: *For a fixed voltage assignment (only task start times can be changed), sequencing tasks in the non-*

Theorem 2: *If a battery fails during some task k, it is always cheaper to repair it by down-scaling its voltage than by inserting an off-line period before k.*

Theorem 3: *Given a pair of two identical tasks in the profile and a delay slack to be utilized by voltage down-scaling, it is always better to use the slack on the later task than on earlier task.*

The proposed scheduling algorithm operates in two phases. In the first phase, a feasible schedule is obtained by (i) using the earliest deadline first EDF algorithm, (ii) trying to generate a non-increasing order of loads, and (iii) ensuring that there is no failure during the battery discharge. In the second phase, the algorithm continues voltage down-scaling in order to fully utilize the available delay slack. Figure 2 describes the proposed algorithm .

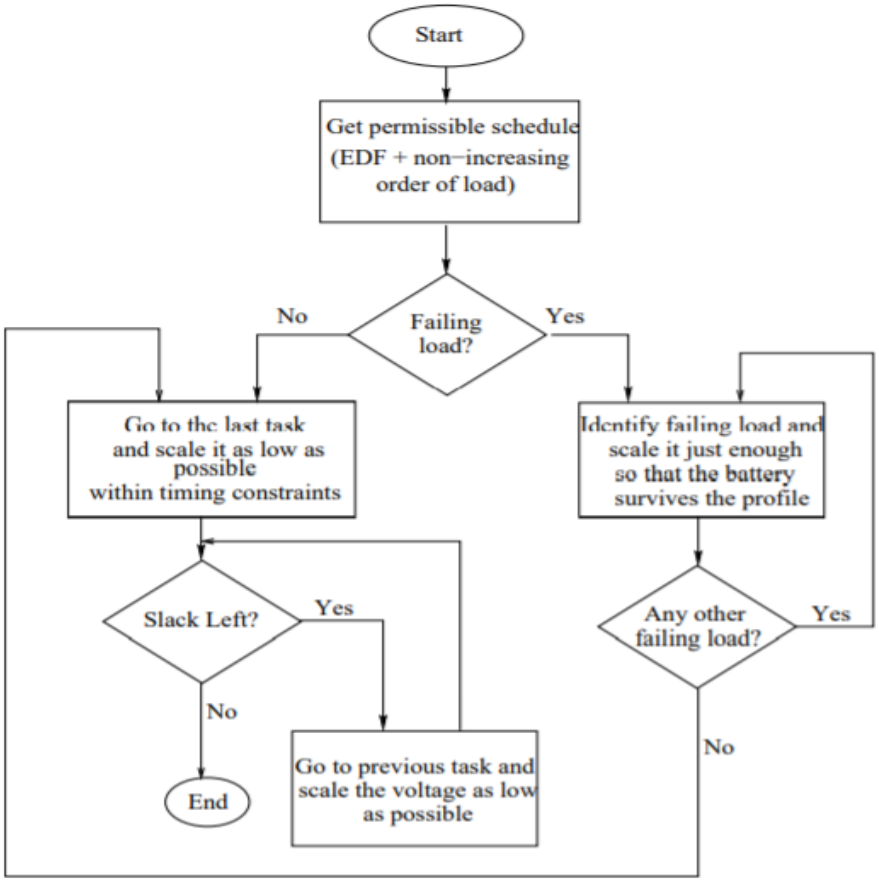


Fig. 2. Top level view of the algorithm for single processor system.

Phase One: Generating a Feasible Schedule

At the start of this phase, all the tasks are assigned to the highest voltage. First the tasks are arranged according to the EDF policy and then a greedy approach is used to reschedule them so that a non-increasing order of task currents is obtained if possible. This step is justified by Theorem 1. The algorithm then checks the Q value of the new schedule. If $Q < 0$ the failure recovery procedure is called to repair the battery failure. Once the Q value is positive, the slack utilization procedure is called.

Failure Recovery: This procedure reshapes the profile so that the battery survives the task set. In each call, the pro-

cedure repairs the earliest failing load as follows. It downscales the task voltage by the minimum amount such that the following conditions are met: (1) the task no longer fails, and (2) the deadlines are met (voltage task scaling increases the task delay). If condition (2) is violated the program control shifts to the previous task in the sequence and the same procedure is repeated. Minimum scaling is justified by Theorem 3 as it leads to delay slack being maximal for the later tasks. The above-mentioned process is repeated until all the failing loads are identified and the final Q at the end of the whole profile is positive. At this stage, the battery has survived the profile and now the slack utilization algorithm can be used to further increase the charge slack Q. Note that recovery insertion is not considered as an alternative to voltage scaling in keeping with Theorem 2.

Phase Two:Slack Utilization

This procedure is based on Theorem 3 which states that the greatest improvement in battery lifetime is obtained if the slack can be used as much as possible by the later tasks. The tasks are considered one by one, starting from the last tasks in the sequence. The last task is scaled to the lowest possible voltage in the set S_v subject to deadline constraints. The process is repeated until there is no slack available or none of the tasks can be assigned to a lower voltage.

4. Discuss adiabatic charging principle with equations

We may now make several observations:

- Of all possible distributions of charging current over time, a constant current causes the least dissipation. In this sense, constant-current charging is the most efficient way to charge a capacitance through a resistance to a certain voltage in a certain time.
- When the exponential current waveform of the conventional case is substituted in Equation 172, R and T cancel out, and the dissipation is once again given by $E_{diss} = (1/2)\,CV_C(T)^2$.
- The dissipation is lower than for the conventional case if the current is constant and $T > 2RC$.
- The dissipation may be made *arbitrarily small* by further extending the charging time: $E_{diss} \sim T^{-1}$.
- A smaller R also brings a lower dissipation. Again, this is in contrast to the conventional case, where dissipation depends only on the capacitance and the voltage swing.

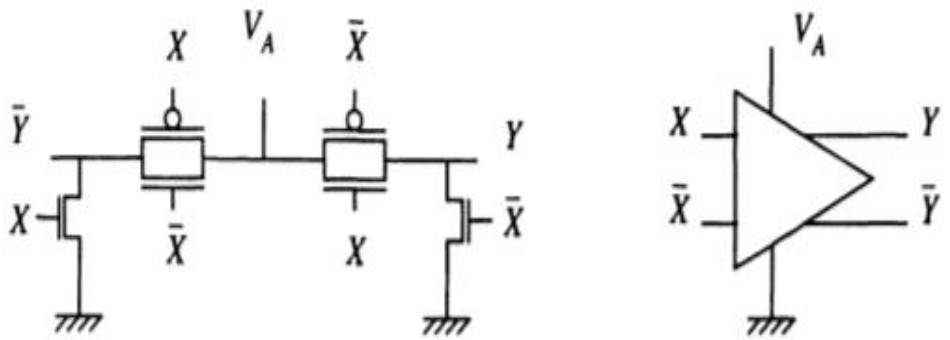


Figure 6.2: Adiabatic amplifier; circuit schematic and logic symbol.

We have efficiently moved energy from the power supply onto a load capacitance by using slow, constant-current charging. Reversing the current source will cause the energy to flow from the load capacitance back into the supply. Thus, in marked contrast to the conventional case, energy is not discarded (dissipated) after being used only once. The power supply must be designed to be able to retrieve the energy fed back to it; otherwise, only half of the potential benefit is realized.

It is clear from Equation 175 that to achieve the non-exponential current waveform desirable for low dissipation, it is necessary to provide a non-constant supply voltage. Adiabatic-switching circuits thus require non-standard power supplies with time-varying voltage and current. We will sometimes refer to these as "pulsed-power supplies." A low overall dissipation can be achieved only if the voltage and current waveforms of the supply both allow power-frugal logic circuits to be built and are possible to generate efficiently. A compromise between the conventional constant-voltage case and the ideal constant-current case may be reached by using sinusoidal waveforms, which can be efficiently generated with inductor-based pulsed-power supplies (as described in Section 6.5).

5. Explain adiabatic amplifier circuit with equations.

Figure 6.2 shows a simple adiabatic amplifier for capacitive loads. It consists of two transmission gates (T-gates) and two NMOS clamps. The input is

dual-rail encoded, since both signal polarities are needed to control the T-gates. The output is also dual-rail encoded, which is required when other T-gates (such as those in other amplifiers) are to be controlled by the output signal. Also, dual-rail signalling keeps the capacitive load on the power supply data-independent, which simplifies the power supply design (cf. Section 6.5).

The operation of the amplifier is straightforward. First, the input is set to a valid value: X and $\bar{X}$ cannot be equal. Next, the amplifier is "energized" by applying to V_A a slow voltage ramp from 0 to V_{dd}. The load capacitance connected to one of the outputs is adiabatically charged to V_{dd} through one of the T-gates, while the other output is clamped to ground. When charging is complete, the output signal pair is valid and can be used as an input to other circuits. Next, the amplifier is de-energized by ramping the voltage on V_A back to 0. The signal energy that was stored on the load capacitance flows back into the power supply connected to V_A. The input signal pair must be stable throughout the process.

The dissipation caused by the operation of the amplifier is easy to determine. As shown in Section 6.1, the dissipation caused by adiabatically charging and discharging a capacitance depends on the on-resistance of the switch. The analysis assumed that the resistance was linear. MOS devices are highly non-linear, but a T-gate can be linearized to a first approximation by carefully selecting the device widths, as is shown next.

A T-gate is turned on with minimal on-resistance when the gate of the PMOS device is grounded and the gate of the NMOS device is connected to V_{dd}. Both devices are in the triode region when the voltage drop across the T-gate is small, which is the intended region for adiabatic circuits. Following Mead and Conway [Mead80], we model the conductance of the NMOS device, G_n, as:

$$G_n = \frac{C_n}{K_n}(V_{dd} - V_{ch} - V_{th}) \tag{176}$$

$$K_n = \frac{L^2}{\mu_n} \tag{177}$$

V_{ch} is the average channel voltage, V_{th} is the threshold voltage, and C_n is the gate capacitance of the device. K_n is a process constant that combines mobility,

μ_n, and channel length, L (the minimum channel length allowed in the process is used for all devices). Likewise, for the PMOS device, we get:

$$G_p = \frac{C_p}{K_p}(V_{ch} - V_{th}) \tag{178}$$

These equations do not take into account body effects nor the difference in threshold voltage between NMOS and PMOS devices. Accuracy is therefore limited to within a factor of two.

The sum of the two conductances may be simplified by selecting the widths of the two MOS devices such that $C_n/K_n = C_p/K_p$:

$$G_p + G_n = \frac{C_n}{K_n}(V_{dd} - V_{ch} - V_{th} + V_{ch} - V_{th}) = \frac{C_n}{K_n}(V_{dd} - 2V_{th}) \tag{179}$$

The on-resistance of the T-gate is then independent of the channel voltage:

$$R_{TG} = \frac{K_n}{C_n(V_{dd} - 2V_{th})} \tag{180}$$

Equations 179 and 180 are valid only when both devices are conducting, which is not the case when V_{ch} is within one threshold voltage of either supply rail. At these extremes, the on-resistance will be less than the value given by Equation 180, so the formulation can be used as an estimate of the upper bound on the resistance. In practice, the body effect increases the resistance further.

The energy efficiency of the amplifier may now be analyzed with the help of Equations 174 and 180. The dissipation in the amplifier caused by first charging and then discharging one load capacitance, C_L, is:

$$E_{load} = 2\xi\frac{R_{TG}C_L}{T}C_L V_{dd}^2 = \frac{2\xi}{T}\frac{K_n}{C_n(V_{dd} - 2V_{th})}C_L^2 V_{dd}^2 \tag{181}$$

Additionally, parasitic effects such as diffusion capacitance of the T-gates and the clamp NMOS devices will contribute to the total load capacitance. Terms modelling these effects may be easily added to Equation 181. For simplicity, parasitics are neglected in this analysis.

Let $V_{dd} = m \cdot V_{th}$, and collect all process constants in one parameter, $\tau_n = K_n/V_{th}$. Then:

$$E_{load} = \left(2\xi\frac{\tau_n}{T}\frac{1}{(m-2)}\frac{C_L}{C_n}\right)C_L V_{dd}^2 = \left(2\xi\frac{\tau_n}{T}\frac{m^2}{(m-2)}\frac{C_L}{C_n}\right)C_L V_{th}^2 \tag{182}$$

We see that the dissipation decreases linearly with increasing T. Also, since $V_{dd} = m \cdot V_{th}$, the dissipation increases only linearly with the voltage swing, as opposed to the V_{dd}^2 dependence of the conventional case.

Energy is also dissipated to drive the input capacitances. When this energy is taken into account, the dependence on T and V_{dd} is affected, as is shown in the following sections.

6. Explain variation tolerant design techniques

Variation-Tolerant Circuits: Circuit Solutions and Techniques

Jim Tschanz, Keith Bowman, Vivek De
Circuit Research Lab, Intel Corporation
JF3-334, 2111 N.E. 25th Avenue, Hillsboro, OR 97124
(503) 712-4360, james.w.tschanz@intel.com

ABSTRACT
Die-to-die and within-die variations impact the frequency and power of fabricated dies, affecting functionality, performance, and revenue. Variation-tolerant circuits and post-silicon tuning techniques are important for minimizing the impacts of these variations. This paper describes several circuit techniques that can be employed to ensure efficient circuit operation in the presence of ever-increasing variations.

Categories and Subject Descriptors
B.7.1 [**Integrated Circuits**]: Hardware – *Types and design styles*.

General Terms: Design, Reliability, Performance

Keywords: Parameter variation, high-performance design, body bias.

1. INTRODUCTION
Process variations are expected to worsen in future technology generations due to difficulties with printing nanometer-scale geometries using standard lithography [1]. A major source of variation in CMOS circuits is due to variation in the channel length of the devices, although other factors such as dopant fluctuation and non-uniformity in etching and polishing of interconnect layers also play an important role. These variations include die-to-die as well as within-die components, and impact the maximum clock frequency (Fmax) and leakage of microprocessor dies. For some variation-sensitive circuits, such as SRAM arrays and dynamic logic, process variations can result in functionality issues and yield loss.

In addition to process variations – which are mainly static – circuits also need to operate correctly under dynamic fluctuations of supply voltage, temperature, and noise [2]. As processor power consumption and frequency continue to rise, it is becoming increasingly difficult to deliver the required power with minimum voltage transients. These voltage transients, which occur when there is a step change in the current demand for the processor, reduce the effective supply voltage, and hence, the processor Fmax. This effect has typically been mitigated by a combination of circuit techniques (e.g., decoupling capacitors), architectural techniques (e.g., staged clock gating) and simple margining of the frequency. Similarly, microprocessors can experience a wide range of operating temperatures, but must be designed to operate correctly under the worst-case condition. If the frequency is set by the worst-case temperature, the processor is operating sub-optimally whenever the temperature is lower.

As both static and dynamic variations increase, techniques are necessary at the system, architecture, and circuit level to reduce the impact of these variations while providing the highest performance for the given power constraints. This paper gives an overview of the effect of variations, and then describes several circuit techniques that can be applied post-silicon for variation tolerance.

2. VARIATION TRENDS
Functionality, Fmax, and power consumption of individual dies are influenced by both die-to-die (D2D) and within-die (WID) variation components. The impact of within-die variation, which causes differences in path delays fabricated on the same die, is heavily influenced by circuit optimization decisions such as transistor sizing, threshold voltage assignment, and number of critical paths in the design. Figure 1 shows that as the number of independent critical paths increases, the mean of the maximum critical path delay (which corresponds to the Fmax) increases as well. The magnitude of the WID variation also depends on critical path depth, where paths with fewer logic stages experience less averaging of random variations resulting in larger variability. Due to increasing complexity and performance requirements for microprocessor designs, the number of critical paths increases with each generation while the logic depth typically decreases. Both trends worsen the impact of within-die variations.

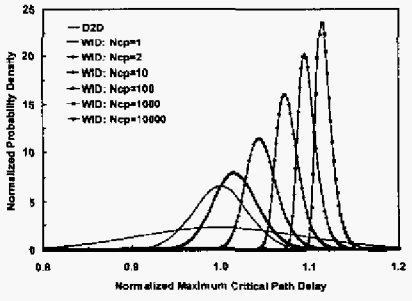


Figure 1. Impact of within-die variations on product performance, as a function of the number of statistically-independent critical paths (N_{CP}).

DAC 2005, June 13–17, 2005, Anaheim, California, USA.
ACM 1-59593-058-2/05/0006.

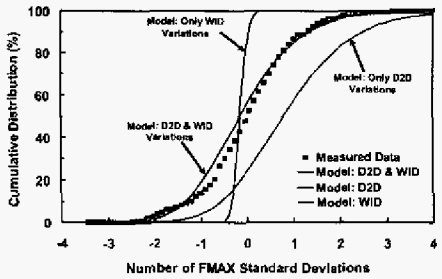


Figure 2. Individual contributions of D2D and WID variations.

Figure 2 demonstrates the interaction between die-to-die and within-die variation components. The variance of the combined distribution is determined mainly by the die-to-die component, while the mean Fmax is primarily a function of within-die variations. These variations combine to affect the frequency and power distributions for the fabricated dies, and therefore both are important to consider when optimizing a design for performance, power, and revenue. Typically, these variations are handled by a combination of design margining (which can lead to a worst-case design which operates inefficiently under normal conditions) and frequency binning (which impacts revenue and yield). When frequency binning is done, dies with a slow Fmax are either discarded or sold at a reduced price, while dies with excessive leakage or total power will violate the system power specification and must be discarded. Thus, the amount of process variations directly impacts the revenue.

3. CIRCUIT TECHNIQUES

One useful technique for reducing the impact of these variations at the circuit level is substrate or body biasing, where a non-zero voltage is applied between the body and source of a transistor. Depending on the voltage applied, the threshold voltage is either increased (which reduces the leakage) or reduced (which increases the Fmax). Thus, adaptive body bias (ABB) can be used after fabrication to compensate for the effects of process variations – each die receives a unique bias voltage which maximizes the frequency of that die subject to power constraints. Figure 3 shows the native leakage vs. Fmax distribution as well as the resulting distribution after ABB is applied. ABB reduces the sigma of the frequency variation by 6X and moves over 30% of the dies into the highest frequency bin.

ABB is effective at compensating for die-to-die variations, but within-die variations cannot be handled using only a single bias value per die. Instead, the die can be divided into multiple regions, each of which can potentially receive a different body bias voltage after fabrication. Figure 3 shows that this within-die ABB technique further reduces the frequency variation and moves 97% of the dies into the highest bin.

It is possible to use supply voltage as a method of reducing the impacts of process variations as well. Both switching and leakage power have a super-linear dependence on supply voltage; therefore, total power and frequency can be modulated by choosing the optimum supply voltage. Figure 4 demonstrates the binning improvement possible with an adaptive V_{DD} technique, where the number of dies in the top two frequency bins improves by 45%. Since switching power and leakage power respond differently to supply voltage and threshold voltage, the combination of ABB and adaptive V_{DD} is more beneficial. As process variations become more important, circuit designers will likely include additional circuit features which may be tuned post-silicon for variation tolerance.

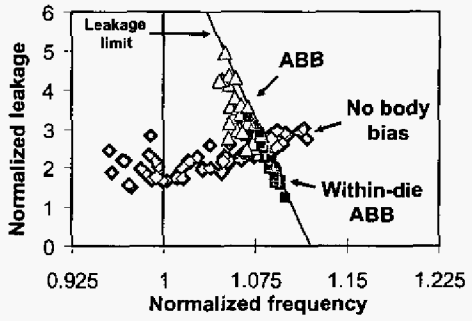


Figure 3. Leakage vs. Fmax distribution for dies without body bias, ABB, and with WID-ABB.

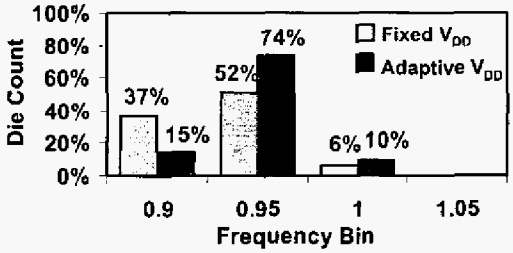


Figure 4. Binning improvement for adaptive V_{DD}.

4. CONCLUSION

Circuit design techniques that account for variation effects will increasingly become important as variations worsen. Optimizations at different levels of the design – system, architecture, and circuits – provide opportunities to reduce the impact of these variations. At the circuit level, adaptive body bias and adaptive supply voltage techniques have been shown to reduce the variation in frequency of fabricated dies, improving the mean frequency and number of dies in the highest bin.

5. REFERENCES

[1] S. Borkar et. al., "Design and reliability challenges in nanometer technologies," *Proc. DAC 2004*, p. 75.

[2] S. Borkar et. al., "Parameter variations and impact on circuits and microarchitecture," *Proc. DAC 2003*, pp. 338-342.

[3] K. A. Bowman et. al., "Impact of die-to-die and within-die parameter fluctuations on the maximum clock frequency distribution for gigascale Integration," *IEEE J. Solid-State Circuits*, pp. 183-190, Feb. 2002.

[4] J. Tschanz et. al., "Adaptive body bias for reducing impacts of die-to-die and within-die parameter variations on microprocessor frequency and leakage", *IEEE J. Solid-State Circuits*, pp. 1396-1402, Nov. 2002.

[5] J. Tschanz et. al., "Effectiveness of adaptive supply voltage and body bias for reducing impact of parameter variations in low power and high performance microprocessors," *IEEE J. Solid-State Circuits*, pp. 826-829, May 2003.

1.Draw the energy band diagram of MIS (metal insulator semiconductor) structure for accumulation region operation of PMOS device.

2.Draw the energy band diagram of MIS (metal insulator semiconductor) structure when negative bias applied. **(ANSWER SAME FOR BOTH)**

When the voltage V is negative, the holes in the p-type semiconductor are attracted to and accumulate at the semiconductor surface in contact with the insulator layer. Therefore this condition is called *accumulation*. In the absence of a current flow, the carriers in the semiconductor are in a state of equilibrium and the Fermi level appears as a straight line. The Maxwell–Boltzmann statistics relates the equilibrium hole concentration to the intrinsic Fermi level:

$$p_0 = n_i\, e^{(E_i - E_F)/kT} \tag{2.2}$$

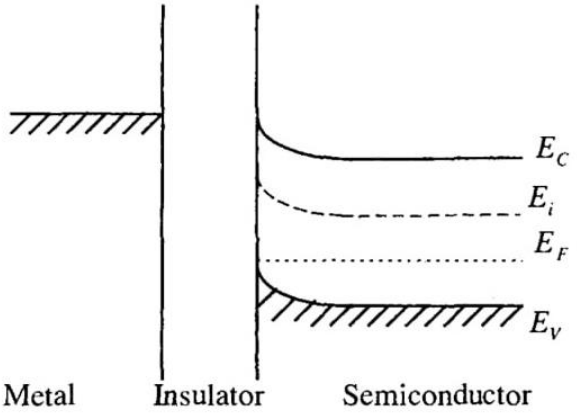


Figure 2.3 Energy bands when a negative bias is applied.

3.Draw the energy band diagram of MIS (metal insulator semiconductor) structure when positive bias applied

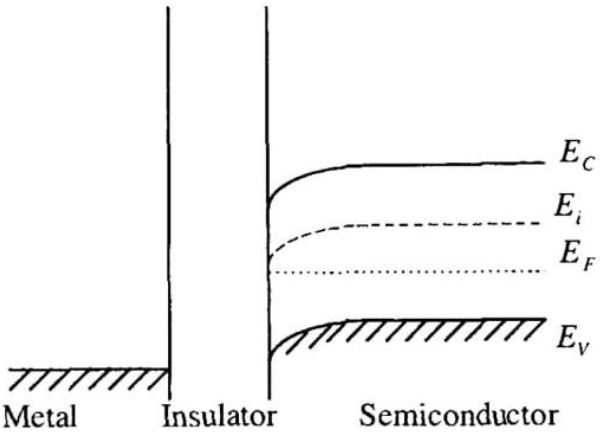


Figure 2.4 Energy bands when a small positive bias is applied.

4. Mention how precomputation logic helps to reduce power dissipation at gate level design with an example.

- A method to trade area for power in synchronous digital circuits.
- **Identify and disable some inputs that are invariant to output. No change in output, but switching activities reduced.**

Examples:

a.

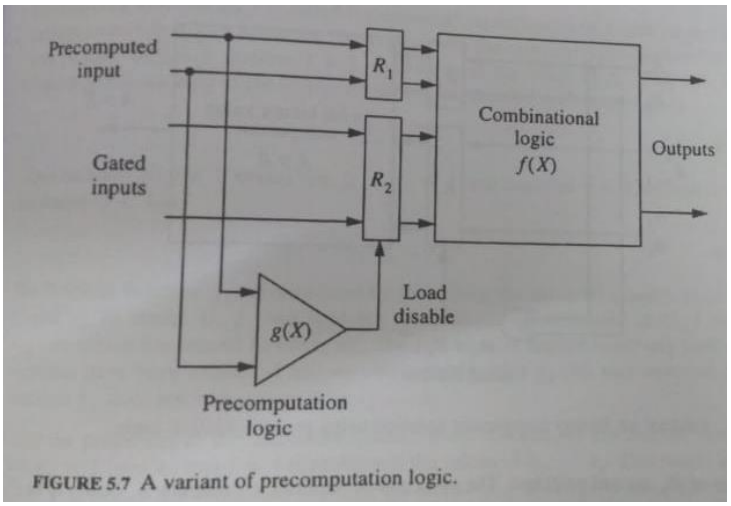


FIGURE 5.7 A variant of precomputation logic.

Due to nature of f(x), output is independent of R2 under certain conditions. In such conditons, we can disable register R2 in order to minimise switching activities.

b. Binary Comparator Function:

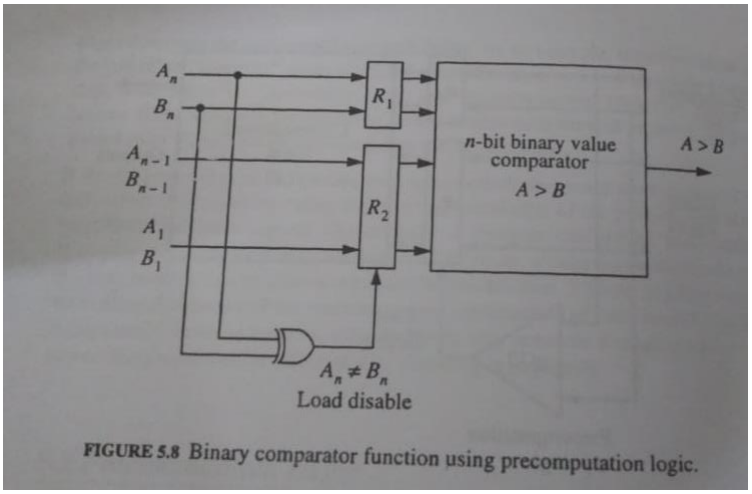


FIGURE 5.8 Binary comparator function using precomputation logic.

When An=1 and Bn=0, output=1 and when An=0 and Bn=1, output= 0 regardless of the inputs of R2, and hence R2 can be disables whenever An≠Bn.

5.The standard deviation of the power samples measured from a circuit has been observed to have +/-20% fluctuation from the mean. Compute how many samples are required so that we are 99% confidence that the error of sample mean is within +/-5%.

Example:

From experience, the standard deviation of the power samples measured from a circuit has been observed to have ±20% fluctuation from the mean. How many samples are required so that we are 99% confidence that the error of sample mean is within ±5%?

We have $\sigma/\mu = 0.2$, $\varepsilon = 0.05$. Since $(1-\alpha) = 0.99$, we have $z_{\alpha/2} = 2.58$ from Table 2.1. Therefore

$$N = \left(\frac{z_{\alpha/2}\,\sigma}{\varepsilon\,\mu}\right)^2 = \left(\frac{2.58 \times 0.2}{0.05}\right)^2 = 107 \qquad (2.25)$$

6. What is the limitation of contemporary CAD tools.

a. In RTL coding there is no provision to use Multi-Vt , Multi-Vdd, Body biasing and power gating in RTL synthesis.
b. Static power reduction techniques cannot be used.
c. Dynamic power can be reduced primarily by reducing the switching activity α. Commonly used techniques in RTL synthesis to reduce α are:
    - Bus encoding
    - Clock gating
    - FSM state assignment

7. Discuss the importance of Monte Carlo simulation

- Better than probabilistic methods for the estimation of power since they achieve superior accuracy with comparable speeds.
- Easier to implement and can be added to existing timing or logic simulation tools.
- Accuracy can be specified up-front with any desired confidence.
- Although the type of circuit may affect the amount of power drawn or the number of samples needed to converge, it will not affect the accuracy or reliability of these methods.
- Monte Carlo methods are based on simple timing or logic simulation techniques and, therefore, experience very few difficulties with feedback circuits

8. Discuss the application of signal entropy

Signal Entropy correlated directly to average switching frequency of signals.

This relations means signal entropy can be used for **power estimation**.

Experimental results have shown that entropy method is acceptable for high-level power analysis.

This method is most suitable for circuits with a high degree of randomness and fails with circuits exhibiting structural regularity, such as multipliers, counters and decoders.

in Figure 3.7. Assuming constant V_{dd}, let f_i be the switching frequency of a node capacitance C_i in the circuit and let N be the total number of nodes. Applying Equation (1.8) in Section 1.2, the power dissipation of the circuit is

$$P = \sum_{i=0}^{N} C_i V_{dd}^2 f_i \quad (3.40)$$

If we assume that $f_i = F$ is constant for all nodes i, we can write

$$P \propto F \sum_{i=0}^{N} C_i = FA \quad (3.41)$$

where A is the sum of node capacitance, proportional to the area or size of the circuit.

9. Discuss precomputation logic with necessary schematic diagrams

SEE QUESTION 4

VLSI SLE:

U1:

Sources of Power Dissipation in Digital ICs.

→ Static Power: Ideally no static loss, since no direct path from Vdd to GND

Practically, switch not perfect.

i) Leakage currents
ii) Substrate Injection Currents
iii) Ratio-ed Logic

→ Dynamic Power: Due to transient switching behaviour of CMOS devices.

i) Capacitance charging of parasitic caps.

Depends on switching activity involved

$P_{dyn} = \alpha C V_{dd}^2 f$

α → Activity, No: of 0-1 transitions / sec

f → Avg. data rate.