**UNIT2 VLSI Design**

**Capacitances of an EMOSFET**

****

****

 **Fig2.1. Overlap capacitance**

****

More accurate modeling of the gate capacitance may be achieved by using a charge basedmodel .For the purpose of delay calculation of digital circuits, we usually approximate *Cg* *Cgs* *Cgd* *Cgb* 􀁾 *C*0 2*CgolW* or use an effective capacitance extracted from simulation. It is important to remember that this model significantly overestimates the capacitance of transistors operating just below threshold.

**Data-dependent gate capacitance**

****

 **Fig2.2 Data-dependent gate capacitance**

**DIFFUSION REGION GEOMETRY**

****

**Fig2.3**

**Total source parasitic capacitance**

****

**Where** *AS* *WD, PS* 2*W* 2*D*

***Built-in potential***

****

**Area junction capacitance term with *MJ* is the *junction grading coefficient***,

****

**Sidewall capacitance**

****

**Capacitance of nMOS transistor**

****

**Fig2.4**

**Delay definitions**

We begin with a few definitions illustrated in Figure 2.5:

1. *Propagation delay time, tpd* = maximum time from the input crossing 50% to the output crossing 50%

2. *Contamination delay time, tcd* = minimum time from the input crossing 50% to the output crossing 50%

3. *Rise time, tr* = time for a waveform to rise from 20% to 80% of its steady-state value

4 .*Fall time, tf* = time for a waveform to fall from 80% to 20% of its steady-state value

5. *Edge rate, trf* = (*tr* + *tf* )/2

****

 **Fig2.5** propagation delay and rise and fall time

Transient response

1.DC analysis tells us Vout if Vin is constant.

2.Transient analysis tells us Vout(t) if Vin(t) changes

3. Requires solving differential equations .

4 Input is usually considered to be a step or a ramp from 0 to VDD or vice versa

**Transient Response of An Inverter to A Ramp Input**



Fig2.5 .Inverter driving a load capacitance







Assuming *Vtn* |*Vtp*| *VDD*, the ramp response includes three phases, as shown in

Table 2.2 **Phases of Inverter Ramp Response**



**Transient Response of Single Order RC- Delay Model**



Fig 2.6 RC circuit

Transfer function



Output voltage



Propagation delay



Approximated propagation delay (now onwards it will only be considered for convenience)





 Fig2.7 Transient Response of 2nd Order RCModel.

**Second order RC-delay model**



Fig.2.8 second order RC-circuit

Transfer function



 Output voltage



Delay



Total propagation delay





Fig 2.9 Comparision of second order to first order RC-circuit

**RC Delay Models**

RC delay models approximate the nonlinear transistor I-V and C-V characteristics with an average resistance and capacitance over the switching range of the gate. This approximation works remarkably well for delay estimation despite its obvious limitations in predicting detailed analog behavior.RC delay model showed that delay is a linear function of the fanout of a gate

 

Fig. 2.10 RC equivalent models of NMOS and PMOS









Fig 2.11 Equivalent circuit of RC Model

**Elmore Delay**

In general, most circuits of interest can be represented as an *RC tree*, i.e., an RC circuit with no loops. The root of the tree is the voltage source and the leaves are the capacitors atthe ends of the branches. The Elmore delay model [Elmore48] estimates the delay from a source switching to one of the leaf nodes changing as the sum over each node *i* of the capacitance *Ci* on the node, multiplied by the effective resistance *Ris* on the shared path from the source to the node and the leaf. Application of Elmore delay is best illustrated through examples

Q1.Estimate *tpd* for a unit inverter driving *m* identical unit inverters in following figure 2.12



*tpd* = (3 + 3*m*)*RC*.

Fig.2.12

Q2 Repeat Example Q1 if the driver is *w* times unit size. in following Fig.2.13



Fig.2.13

*tpd* = ((3*w* + 3*m*)*C*)(*R*/*w*) = (3 + 3*m*/*w*)*RC*

Q3 If a unit transistor has *R* = 10 kῺand *C* = 0.1 fF in a 65 nm process, compute the delay,

in picoseconds, of the inverter in following Fig.2.14 with a fanout of *h* 4.



Fig2.14

*tpd* =(3 + 3*h*)(1 ps) = 15 ps.

Q4 Estimate *tpdf* and *tpdr* for the 3-input NAND gate if the output is loaded with *h* identical NAND gates.

Solution Each NAND gate load presents 5 units of capacitance on a given input. Figure 2.15 (a) shows the equivalent circuit including the load for the falling transition. Node n1 has capacitance 3C and resistance of R/3 to ground. Node n2 has capacitance 3C and resistance (R/3 + R/3) to ground. Node Y has capacitance (9 + 5h)C and resistance (R/3 + R/3 + R/3) to ground.



Figure 2.15 Falling circuits

The Elmore delay for the falling output is the sum of these RC products, tpdf = (3C)(R/3) + (3C)(R/3 + R/3) + ((9 + 5h)C)(R/3 + R/3 + R/3) = (12 + 5h)RC. Figure 2.15 (b) shows the equivalent circuit for the falling transition. In the worst case, the two inner inputs are 1 and the outer input falls. Y is pulled up to VDD through a single pMOS transistor. The ON nMOS transistors contribute parasitic capacitance that slows the transiton. Node Y has capacitance (9 + 5h)C and resistance R to the VDD supply. Node n2 has capacitance 3C. The relevant resistance is only R, not (R + R/3), ecause the output is being charged only through R. This is what is meant by the resistance on the shared path from the source (VDD) to the node (n2) and the leaf (Y). Similarly, node n1 has capacitance 3C and resistance R. Hence, the Elmore delay for the rising output is tpdr = (15 + 5h)RC. The R/3 resistances do not contribute to this delay. Indeed, they shield the diffusion capacitances, which don’t have to charge all the way up before Y rises. Hence, the Elmore delay is conservative and the actual delay is somewhat faster. Although the gate has equal resistance pulling up and down, the delays are not quite equal because of the capacitances on the internal nodes.

Q5.Estimate the contamination delays *tcdf* and *tcdr* for the 3-input NAND gate if the output is loaded with *h* identical NAND gates.

SOLUTION**:** The contamination delay is the fastest that the gate might switch. For the falling transition, the best case is that the bottom two nMOS transistors are already ON when the top one turns ON. In such a case, the diffusion capacitances on *n*1 and *n*2 have already been ischarged and do not contribute to the delay. Figure 2.16(a) shows the equivalent circuit and the delay is *tcdf* = (9 + 5*h*)*RC*. For the rising transition, the best case is that all three pMOS transistors turn on simultaneously. The nMOS transistors turn OFF, so *n*1 and *n*2 are not connected to the output and do not contribute to delay. The parallel transistors deliver three times as much current, as shown in Figure 2.16(b), so the delay is *tcdr* = (3 + (5/3)*h*)*RC*.





Fig.2.16 Contamination delay circuits

Q6 Sketch a 3-input NAND gate with transistor widths chosen to achieve effective rise and fall resistance equal to that of a unit inverter (R). Annotate the gate with its gate and diffusion capacitances. Assume all diffusion nodes are contacted. Then sketch equivalent circuits for the falling output transition and for the worst-case rising output transition.

 Solution: Q6

   

 Fig.2.17 three input NAND gate with capacitances

NMOS and PMOS models for calculation of K-value by placing in any gate for analysis of delay and capacitance.

Figure 2.18.RC models

L**inear Delay Model**



Fig2.19 Normalised delay vs.fan out.

Figure 2.19 plots normalized delay vs. electrical effort for an idealized inverter and 3-input NAND gate. The *y*-intercepts indicate the parasitic delay, i.e., the delay when the gate drives no load. The slope of the lines is the logical effort. The inverter has a slope of 1 by definition. The NAND has a slope of 5/3.

d =f + p

p= parasitic delay

p is the parasitic delay inherent to the gate when no load is attached, The parasitic delay of a gate is the delay of the gate when it drives zero load. It can be estimated with RC delay models. The parasitic delay also depends on the ratio of diffusion capacitance to gate capacitance.

f= effort delay

f is the effort delay or stage effort that depends on the complexity and fan out of the gate:

f =gh

g=logical Effort

Logical effort of a gate is defined as the ratio of the input capacitance of the gate to the input

capacitance of an inverter that can deliver the same output current

h=electrical effort

h=Cout/Cin

*  *

Figure 2.20 logical effort of NOT,NAND and NOR

***Logical* effort** :Logical effort of a gate is defined as the ratio of the input capacitance of the gate to the input capacitance of an inverter that can deliver the same output current

Table 2.3.Logical efforts of Gates

 **

***Parasitic delay ;***The parasitic delay of a gate is the delay of the gate when it drives zero load. It can be estimated with RC delay models. A crude method good for hand calculations is to count only diffusion capacitance on the output node

 Table2.4. Parasitic delay of gates

**

**Drive**

A good standard cell library contains multiple sizes of each common gate. The sizes are typically labeled with their drive. For example, a unit inverter may be called inv\_1x. An inverter of eight times unit size is called inv\_8x. A 2-input NAND that delivers the same current as the inverter is called nand2\_1x.It is often more intuitive to characterize gates by their drive, *x*, rather than their input capacitance. If we redefine a unit inverter to have one unit of input capacitance, then the drive of an arbitrary gate is

 x = Cin / g

Delay can be expressed in terms of drive as

d= Cout  /x +p

**Logical Effort of path**

**

 *Fig 2.21path logical effort G*

**Q7**.Estimate the minimum delay of the path from *A* to *B* in Figure 2.22 and 2.23 and choose transistor sizes to achieve this delay. The initial NAND2 gate may present a load of 8 λ of transistor width on the input and the output load is equivalent to 45 λ of transistor width.

**Solution:** The path logical effort is *G* =(4/3) .(5/3) .(5/3) =100/ 27. The path electrical effort is *H* =45/8. The path branching effort is *B* =3 .2 =6. The path effort is *F* =*GBH* =125. As there are three stages, the best stage effort is . The path parasitic delay is *P* =2 +3 +2 =7. Hence, the minimum path delay is *D* =3 .5 +7 =22 in units of λ, or 4.4 FO4 inverter delays. The gate sizes are computed with the capacitance transformation from Cin= Couti\* gi/f working backward along the path: *y* =45 .(5/3)/5 =15. *x* (15 15) (5/3)/5 10. We verify that the initial 2-input NAND gate has the specified size of (10 10 10) (4/3)/5 8. The transistor sizes in Figure 2.23 are chosen to give the desired amount of input capacitance while achieving equal rise and fall delays. For example, a 2-input NOR gate should have a 4:1 P/N ratio. If the total input capacitance is 15, the pMOS width must be 12 and the nMOS width must be 3 to achieve that ratio. We can also check that our delay was achieved. The NAND2 gate delay is *d*1 *g*1*h*1 *p*1 (4/3) (10 10 10)/8 2 7. The NAND3 gate delay is *d*2 *g*2*h*2 *p*2 (5/3) (15 15)/10 3 8. The NOR2 gate delay is *d*3 *g*3*h*3 *p*3 (5/3) 45/15 2 7. Hence, the path delay is 22, as predicted. Recall that delay is expressed in units of λ In a 65 nm process with τ = 3 ps, the delay is 66 ps. Alternatively, a fanout-of-4 inverter delay is 5τ, so the path delay is 4.4 FO4s.

* *

Figure 2.22 Figure 2.23

**Choosing Best Number of Stagest**

 **Q8**.A control unit generates a signal from a unit-sized inverter. The signal must drive unit-sized loads in each bitslice of a 64-bit datapath. The designer can add inverters to buffer the signal to drive the large load. Assuming polarity of the signal does not matter, what is the best number of inverters to add and what delay can be achieved?

**SOLUTION :** Figure 2.24 shows the cases of adding 0, 1, 2, or 3 inverters. The path electrical effort is *H* =64. The path logical effort is *G* =1, independent of the number of inverters. Thus, the path effort is *F* = 64. The inverter sizes are chosen to achieve equal stage effort. The total delay is . The 3-stage design is fastest and far superior to a single stage. If an even number of inversions were required, the two- or four-stage designs are promising. The four-stage design is slightly faster, but the two-stage design requires significantly less area and power.

**

Figure 2.24.1,2,3, and 4 inverters

**Table2.5 Effort and Delay**



**Limitations of Logical Effort**

Limitations of Logical Effort is based on the linear delay model and the simple premise that making the effort delays of each stage equal minimizes path delay. This simplicity is the method’s greatest strength, but also results in a number of limitations

1. Logical Effort does not account for interconnect. Logical Effort is most applicable to high-speed circuits with regular layouts where routing delay does not dominate. Such structures include adders, multipliers, memories, and other data paths and arrays.

2. Logical Effort explains how to design a critical path for maximum speed, but not how to design an entire circuit for minimum area or power given a fixed speed constraint.

3 Paths with nonuniform branching or reconvergent fanout are difficult to analyze by hand.

4. The linear delay model fails to capture the effect of input slope. Fortunately, edge rates tend to be about equal in well-designed circuits with equal effort delay per stage

The circuit in Figure 2.25 has nonuniform branching, reconvergent fanout, and a wire load in the middle of the path, all of which stymie back-of-the-envelope application of Logical Effort. The wire load is given in the same units as the gate capacitances (i.e., multiples of the capacitance of a unit inverter). Assume the inputs arrive at time 0. Write an expression for the arrival time of the output as a function of the gate drives. Determine the sizes to achieve minimum delay

**Q9**.The circuit in Figure 2.25 has nonuniform branching, reconvergent fanout, and a wire load in the middle of the path, all of which stymie back-of-the-envelope application of Logical Effort. The wire load is given in the same units as the gate capacitances (i.e.,multiples of the capacitance of a unit inverter). Assume the inputs arrive at time 0.Write an expression for the arrival time of the output as a function of the gate drives.

**

***Figure2.25***

**Solution**

The delay equations for each gate are obtained

D1 =1+4/3(.x2)+5/3(x3)

D2=2+7/3(x4/x2)

D3=2+ 7/3(x4/x3)

D4=3+(10/x4) +(x5/x4)

D5=1+ 12/x5

Arrival times

*a1=d1*

*a2=d2+a1*

*a3=a1+d3*

*a4=d4 +max(a2,a3)*

*a5=a4 + d5*

***Timing Analysis Model***

To handle a chip with millions of gates, the delay model for a timing analyzer must be easy

enough to compute that timing analysis is fast, yet accurate enough to give confidence.

**Slop based timing model**

A simple approach is to extend the linear delay model by adding a term reflecting the

input slope. Assuming the slope of the input is proportional to the delay of the previous

stage, the delays for rising and falling outputs can be expressed as:

delay\_rise = intrinsic\_rise + rise\_resistance \*capacitance +slope\_rise \*delay\_previous

delay\_fall = intrinsic\_fall +fall\_resistance \*capacitance +slope\_fall \*delay\_previous

Linear delay models are not accurate enough to handle the wide range of slopes and

loads found in synthesized circuits, so they have largely been superseded by nonlinear

delay models.

**Nonlinear delay models.**

nonlinear delay model looks up the delay from a table based on the load capacitance and

the input slope. Separate tables are used to lookup rising and falling delays and output

slopes. The timing analyzer uses interpolation when a specific load capacitance or slope

is not in the table

Table 2.6 Rise time and Cou



Nonlinear delay models are widely used at the time of this writing. However, they do not contain enough information to characterize the delay of a gate driving a complex RC interconnect network with the accuracy desired by some users. They also lack the accuracy to fully characterize noise events. A different model must be created for each voltage and temperature at which the chip might be characterized.

**Current Source Model**

The limitations of nonlinear delay models have motivated the development of current source models.

**1.A *current source model***theoretically should express **the output DC current** as a nonlinear function of the input and output voltages of the cell.

**2**. **A timing analyzer** numerically integrates the output current to find the voltage as a function of time into an arbitrary RC network and to solve for the propagation delay.

**3**. Liberty ***Composite Current Source Model* (CCSM**) instead stores output current as a function of time for a given input slew rate and output capacitance.

4. Competing ***Effective Current Source Model* (ECSM)** stores output voltage as a function of time. The two representations are equivalent, and can be synthesized into a true current source model

 ***Introduction to power***

Today, we are interested in power from a number of points of view. In portable applications, products normally run off batteries. While battery technology has improved markedly over the years, it remains that a battery of a certain weight and size has a certain energy capacity.

The instantaneous power P (t)

 *P* *t* *I* *t* *V* *t* 

 *Energy* consumed

*Average power* over this interval

 

1. Resistor (b)Voltage source (c) Capacitor

Fig.2.26

1. Power dissipated across resistor



(b) Voltage source delivered power



(c) Capacitor: Energy stored



 **Inverter Showing Energy Delivered and Dissipated**



 Fig.2.27

Energy stored



Energy supplied by an inverter



50% energy is wasted only half of the energy from the power supply is stored in the capacitor because the transistor has a voltage across it at the same time a current flows through it. The power dissipated depends only on the load capacitance, not on the size of the transistor or the speed at which the gate switches. Suppose that the gate switches at some average frequency *f*sw. Over some interval *T*,the load will be charged and discharged *Tf*sw times.

Average power dissipation is

 =C *fSw*

Pswirtching=α C *f*

This is called the ***dynamic power***because it arises from the switching of the load. Because

most gates do not switch every clock cycle, it is often more convenient to express switching

frequency *f*sw as an *activity factor* α times the clock frequency *f*. Now, the dynamic power dissipation may be rewritten as

 *Where fsw =*αf

α =Activity factor

The **activity factor** is the probability that the circuit node transitions from 0 to 1, because

that is the only time the circuit consumes power. A clock has an activity factor of α = 1

because it rises and falls every cycle.

**Sources of power Dissipation**

Power dissipation in CMOS circuits comes from two components:

 Dynamic dissipation due to

1 charging and discharging load capacitances as gates switch

2 “short-circuit” current while both pMOS and nMOS stacks are partially ON

*P* dynamic = *P* switching + *P* short circuit

Static dissipation due to

1 Subthreshold leakage through OFF transistors

2.Gate leakage through gate dielectric

3. Junction leakage from source/drain diffusions

4 Contention current in ratioed circuits

Putting this together gives the total power of a circuit

 *P*static=( *I*sub +*I* gate + *I* junct + *I*contention )*VDD*

*P* total = *P* dynamic + *P* static

**Power can also be considered in active, standby, and sleep modes.**

1. *Active power* is the power consumed while the chip is doing useful work. It is usually dominated by *P*switching.

*2.Standby power* is the power consumed while the chip is idle. If clocks are stopped and ratioed circuits are disabled, the standby power is set by leakage

3.In sleep mode, the supplies to unneeded circuits are turned off to eliminate leakage.

This drastically reduces the *sleep power* required, but the chip requires time and energy to wake up so sleeping is only viable if the chip will idle for long enough. Dynamic power also includes a short-circuit power component caused by power rushing from *VDD* to GND when both the pullup and pulldown networks are partially ON while a transistor switches

Q10 A digital system-on-chip in a 1 V 65 nm process (with 50 nm drawn channel lengths and λ = 25 nm) has 1 billion transistors, of which 50 million are in logic gates and the remainder in memory arrays. The average logic transistor width is 12 λ and the average memory transistor width is 4 λ. The memory arrays are divided into banks and only the necessary bank is activated so the memory activity factor is 0.02. The static CMOS logic gates have an average activity factor of 0.1. Assume each transistor contributes 1 fF/m of gate capacitance and 0.8 fF/µm of diffusion capacitance. Neglect wire capacitance for now (though it could account for a large fraction of total power). Estimate the switching power when operating at 1 GHz.

Solution

There are (50 x106 logic transistors)(12 λ)(0.025 μm/ λ )((1 + 0.8) fF/μm) = 27 nF of logic transistors and (950 x 106 memory transistors)(4 λ(0.025 μm/ λ)((1 + 0.8) fF/μm) = 171 nF of memory transistors. The switching power consumption is [(0.1)(27 x 10–9) + (0.02)(171 x10–9)](1.0 V)2(109 Hz) = 6.1 W.

Q11. Consider the system-on-chip from above Example10. Subthreshold leakage for OFF devices is 100 nA/μm for low-threshold devices and 10 nA/μm for high-threshold devices. Gate leakage is 5 nA/μm. Junction leakage is negligible. Memories use low leakage devices everywhere. Logic uses low-leakage devices in all but 5% of the paths that are most critical for performance. Estimate the static power consumption.

**Solution** There are (50 x 106 logic transistors)(0.05)(12 λ)(0.025 μm/λ) = 0.75 x106 μm of low-threshold devices and [(50 x 106 logic transistors)(0.95)(12 λ) + (950 x 106 memory transistors)(4 λ)](0.025 μm/λ) = 109.25 x 106 μm of high-threshold devices. Neglecting the benefits of series stacks, half the transistors are OFF and contribute subthreshold leakage. Half the transistors are ON and contribute gate leakage. *I*sub = [(0.75 x 106 μm)(100 nA/μm) + (109.25 x 106 μm)(10 nA/μm)]/2 = 584 mA. *I*gate = ((0.75 + 109.25) x 106 μm)(5 nA/μm)/2 = 275 mA. *P*static = (584 mA + 275 mA)(1 V) = 859 mW. This is 15% of the switching power and is enough to deplete the battery of a hand-held device rapidly.

 **Sources of Dynamic Power**

**Activity Factor**

The activity factor is a powerful and easy-to-use lever for reducing power. If a circuit can

be turned off entirely, the activity factor and dynamic power go to zero. Blocks are typically

turned off by stopping the clock; this is called *clock gating*

**1.Clock Gating** Clock gating ANDs a clock signal with an enable to turn off the clock to idle blocks The clock enable must be stable while the clock is active (i.e., 1 for systems using positive edge-triggered flip-flops). Following figure shows how an enable latch can be used to ensure the enable does not change before the clock falls. When a large block of logic is turned off, the clock can be gated early in the clock distribution network, turning off not only the registers but also a portion of the global network. The clock network has an activity factor of 1 and a high capacitance, so this saves significant power.

**

***Fig 2.28***

**Switching Probability of logic gates**

 **Activity factor** can be calculated as

**Probability of Gates**

**Table2.7**

**

 Q12. Following figure shows a 4-input AND gate built using a tree (a) and a chain (b) of gates. Determine the activity factors at each node in the circuit assuming the input probabilities

*PA* = *PB* = *PC* = *PD* = 0.5.



Fig2.29

Solution\*



Fig 2.29A .Answer of Q12.

**2.Glitches** :switching probabilities computed in the previous section are only valid if the gates have zero propagation delay .In reality, gates sometimes make spurious transitions called glitches when inputs do not arrive simultaneously.

For example, in following figure2.30, suppose *ABCD* changes from 1101 to 0111. Node *n*4 was 1 and falls to 0. However, nodes *n*5, *n*6, *n*7, and *Z* may glitch before *n*4 changes, as shown in Figure 2.30, The glitches cause extra power dissipation. Chains of gates are particularly prone to this problem. Glitching can raise the activity factor of a gate above 1 and can account for the majority of power in certain circuits such as ripple carry adders and array multipliers

**

*Figure (a) Logic circuit Figure(b )glitches*

 Fig2.30

**3. Switching capacitance**

1. Switching capacitance comes from the wires and transistors in a circuit. Wire capacitance

is minimized through good floorplanning and placement.

2 .Device-switching capacitance is reduced by choosing fewer stages of logic and smaller

 transistors.

3 .Minimum-sized gates can be used on non-critical paths

4. Gates that are large or have a high activity factor and thus dominate the power can be downsized with only a small performance impact.

 5. large energy savings can be made by relaxing a circuit a small amount from the minimum delay point

**4.Gate Sizing Under a Delay Constraint**

In many cases, we are willing to increase delay to save energy.consider a model to compute the energy of a circuit. If a unit inverter has gate capacitance 3*C*, then a gate with logical effort *g*, parasitic delay *p*, and drive *x* has *gx* times as much gate capacitance and *px* times as much diffusion capacitance. The switching energy of each gate depends on its activity factor, the diffusion capacitance of the gate, the wire capacitance *C*wire, and the gate capacitance of all the stages it drives. The energy of the entire circuit is the sum of the energies of each gate.

**

If wire capacitance is expressed in multiples of the capacitance of a unit inverter as *c* = *C*wire /3*C* and we normalize energy for the capacitance and voltage of the process, above equation becomes the sum of the effective capacitances of the nodes

**

we seek to minimize *E* such that the worst-case arrival time is less than some delay *D*.

**Voltage**

1. Voltage has a quadratic effect on dynamic power. The chip may be divided into multiple v*oltage domains*, where each domain is optimized for the needs of certain circuits. For example a system on- chip might use a high supply voltage for memories to ensure cell stability, a medium voltage for a processor, and a low voltage for I/O peripherals running at lower speeds.

2.Voltage also can be adjusted based on operating mode. for example, a laptop processor may operate at high voltage and high speed when plugged into an AC adapter, but at lower voltage and speed when on battery power. If the frequency and voltage scale down in proportion, a cubic reduction in power is achieved.

**Voltage Domains**

Figure2.31 given below shows direct connection of inverters in two domains using high and low

supplies, *VDDH* and *VDDL*, respectively. A gate in the *VDDH* domain can directly drive a

gate in the *VDDL* domain. However, the gate in the *VDDL* domain will switch faster than it

would if driven by another *VDDL* gate

 

 *Fig2.31* (a) *Fig2.31* (b)

**Clustered Voltage Scaling (CVS)**

In this ,two supply voltages can be used in a single block. Figure 2.32 shows an example of lustered voltage scaling. Gates early in the path use *VDDH*. Noncritical gates later in the path use *VDDL*. Voltages are assigned such that a path never crosses from a *VDDL* gate to a *VDDH* gate within a block of combinational logic, so level converters are only required at the registers. CVS requires that two power supplies be distributed across the entire block. This can be done by using two power rails. Note that many processes require a large spacing between n-wells at different potentials, which limits the proximity of the *VDDH* and *VDDL* gates.



Fig2.32

**Dynamic Voltage Scaling (DVS)**

Systems that can save large amounts of energy by reducing the clock frequency to the minimum sufficient to complete the task on schedule, then reducing the supply voltage to the minimum necessary to operate at that frequency. This is called *dynamic voltage scaling* (DVS) or *dynamic voltage/frequency scaling* (DVFS).

Figure drawn below shows a block diagram for a basic DVS system. The DVS controller takes information from the system about the workload and/or the die temperature. It determines the supply voltage and clock frequency sufficient to complete the workload on schedule or to maximize performance without overheating. A switching voltage regulator efficiently steps down *V*in from a high value to the necessary *VDD*. The core logic contains a phase-locked loop or other clock synthesizer to generate the specified clock frequency. The DVS controller determines the operating frequency, then chooses the lowest supply voltage suitable for that frequency. One method of choosing voltage is with a precharacterized table of voltage vs. frequency.



Figure2.33 DVS

**Short-Circuit Power Dissipation**

1. Short-circuit power dissipation occurs as both pullup and pulldown networks are partially

ON while the input switches.

2. It increases as the input edge rates become slower because both networks are ON for more time.

3.It decreases as load capacitance increases because with large loads the output only switches a small amount during the input transition, leading to a small *Vds* across the transistor that is causing the short-circuit current.

 4. Unless the input edge rate is much slower than the output edge rate, short-circuit current is a small fraction (< 10%) of current to the load and can be ignored in hand calculations.

5.It is good to use relatively crisp edge rates at the inputs to gates with wide transistors to minimize their short-circuit current

6.Short-circuit power is strongly sensitive to the ratio *v* = *Vt* / *VDD*.

**Resonant Circuits**

Resonant circuits seek to reduce switching power consumption by letting energy slosh

back and forth between storage elements such as capacitors and inductors rather than

dumping the energy to ground. The technique is best suited to applications such as clocks

that operate at a constant frequency





Figure 2.34 Resonant clock network

**Static Power Sources**

Static power arises from subthreshold, gate, and junction leakage currents and contention current. Entire books have been written about leakage, but this section summarizes the key effects

1. Subthreshold leakage current flows when a transistor is supposed to be OFF. For *Vds* exceeding a few multiples of the thermal voltage (e.g., *Vds* > 50 mV)



where *I*off is the subthreshold current at *Vgs* = 0 and *Vds* = *VDD*, and *S* is the subthreshold slope (about 100 mV/decade). *I*off is a key process parameter defining the leakage of a single OFF transistor. It ranges from about 100 nA/μm for typical low-*V*t devices to below **1 nA/μm** for high-*Vt* devices. *I*off is usually specified at 25 °C and increases exponentially with temperature because *V****t***decreases with temperature and *S* is directly proportional to temperature.

1. The leakage through two or more series transistors is dramatically reduced on account

 of the *stack effect*

Figure 2.35 given below shows two series OFF transistors with gates at 0 volts. The drain of *N*2 is at *VDD*, so the stack will leak. However, the middle node voltage *Vx* settles to a point that each transistor has the same current. If *Vx* is small, *N*1 will see a much smaller DIBL effect and will leak less. As *Vx* rises, *Vgs* for *N*2 becomes negative, reducing its leakage. Hence, we would expect that the series transistors leak less.



 Series OFF transistors

 Fig.2.35

3. Subthreshold leakage cannot be reduced without consideration of other forms of leakage.

**Gate Leakage**

Gate leakage occurs when carriers tunnel through a thin gate dielectric when a voltage is applied across the gate. Gate leakage is an extremely strong function of the dielectric thickness. pMOS gate leakage is an order of magnitude smaller in ordinary SiO2 gates and can often be ignored, but it can be significant for other gate dielectrics. Gate leakage also depends on the voltage across the gate. gate leakage can be alleviated by stacking transistors such that the OFF transistor is closer to the rail

**Junction Leakage**

Junction leakage occurs when a source or drain diffusion region is at a different potential from the substrate. Although the ordinary leakage of reverse-biased diodes is usually negligible

**Contention Current**

Static CMOS circuits have no contention current. However certain alternative circuits inherently draw current even while quiescent

Q13.Express an energy as delay varies from the minimum possible (*D*min = 23.44 τ) to 50 τ. Assume that the input probabilities are 0.5.

Solution: following Figure 2.36 Logic circuit shows the activity factors of each node. Hence, the energy of this circuit is E =(1/4)[1+4/3(x2)+5/3(x3)] + (3/16)[2x2+7/3(x4)] + (3/16)[2 X3+ 7/3(x4)] + (87/1024)[3x4+(10) +(x5/x4)] +(87/1024) [x5+ 12]



Figure 2.36 Logic circuit