Attention 8 sem. students please refer unit -2 and unit-3

Contents..

UNIT-II: Low Power design at Circuit and logic level

Power consumption in circuits

• The power reduction techniques at the circuit level are quite limited if compared with the other techniques at higher abstraction levels

• At the circuit level , percentage power reduction in the teens is considered good

• However, circuit techniques can have major impact because some circuits, especially in cell-based design, are repeated thousands of times on a chip

• Therefore, circuit techniques with a small percentage improvement should not be overlooked

• Circuits designed manually can often be analyzed in great details

- This allows us to optimize the circuit speed, power and area to suit our specification
- One important circuit technique is the reduction of operating voltage

– The general rule is to select the lowest voltage that is acceptable

Transistor and Gate Sizing

- At the circuit level, transistors are the basic building blocks and a digital circuit can be viewed as a network of transistors

with some additional parasitic elements such as capacitors and resistors

- Transistor sizes are the most important factor affecting the quality i,e area, and power dissipation of a circuit
- Some studies assume that the sizing problem is a convex function and linear programming can be used to solve the sizing problem optimally
- Another problem encountered in cell based design is gate sizing

• The goal is to choose a set of gate sizes that best fits the design constraints.

Flip Flops & Latches design

• Flip flops and latches are some of the most frequently used elements in digital VLSI

• In synchronous systems, they are the starting and ending points of signal delay paths, which decide the maximum speed of the system

• Typically, they consume more power because they are clocked at the system operating frequency

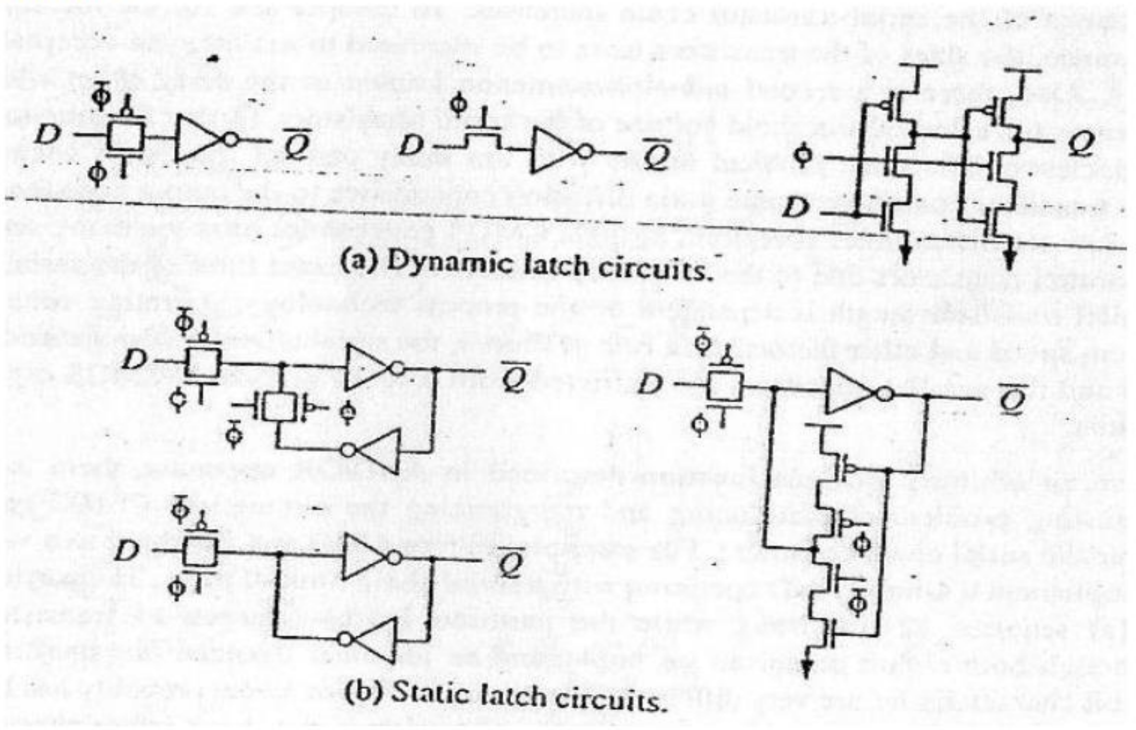
• Careful design of the flip flop and latch circuits is important to a low power VLSI design

• The energy dissipation of a flip flop can be divided into two components:

1. Clock energy
2. Data energy

Flip Flop & Latch Circuits

- The figure below shows various implementation of CMOS latches



(a) Dynamic latch circuits.

(b) Static latch circuits.

• The above circuits provides a different tradeoff among setup time, hold time, data to output and clock to output delay

• The use of NMOS pass transistors instead of transmission gates reduces the loading capacitance of the clock pin at the cost of reduced speed

• This eliminates the need for a two phase non-overlapping clock on the system or a phase splitter inverter that consumes power inside the cell

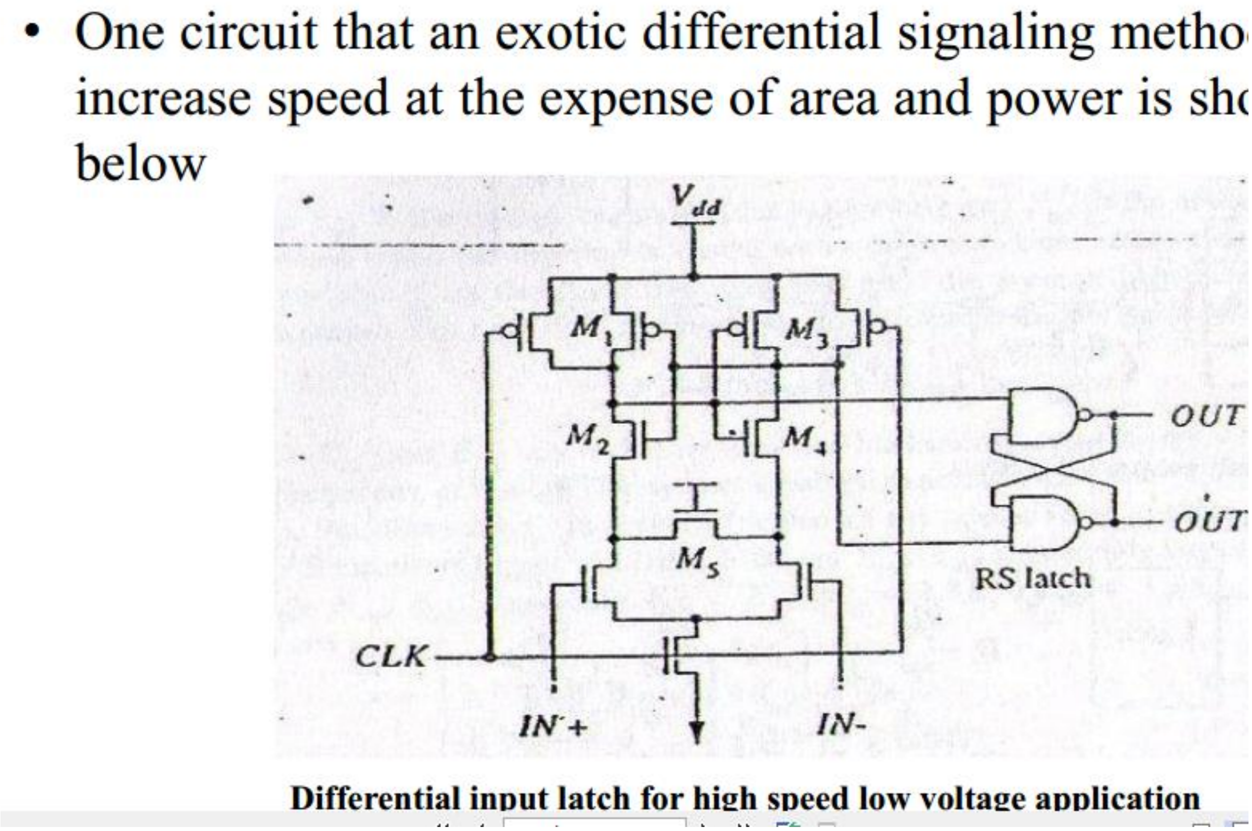
• The circuit suffers from threshold voltage loss when logic 1 is propagated through the NMOS pass transistor

- The *single phase latch circuit* avoids the threshold voltage problem but relies on charge storage effect to retain its data value
- This cause some loss in the noise margin but the circuit has been successfully used in high performance processor design.

Flip Flop & Latch Circuits

- A flip flop is typically constructed from two latches connected together
- The single phase dynamic flip flop at the top left is a cascaded version of two single phase latches
- It is suitable for some low power applications because it does not require internal phase splitting inverter for the clock pin.
- The circuit at the bottom was reported to achieve lower power at the same speed with more transistors, compared to a standard flip flop design on the top right.

Flip Flop & Latch Circuits



- One circuit that an exotic differential signaling metho increase speed at the expense of area and power is sh below

Differential input latch for high speed low voltage application

Self-gating Flip-flop

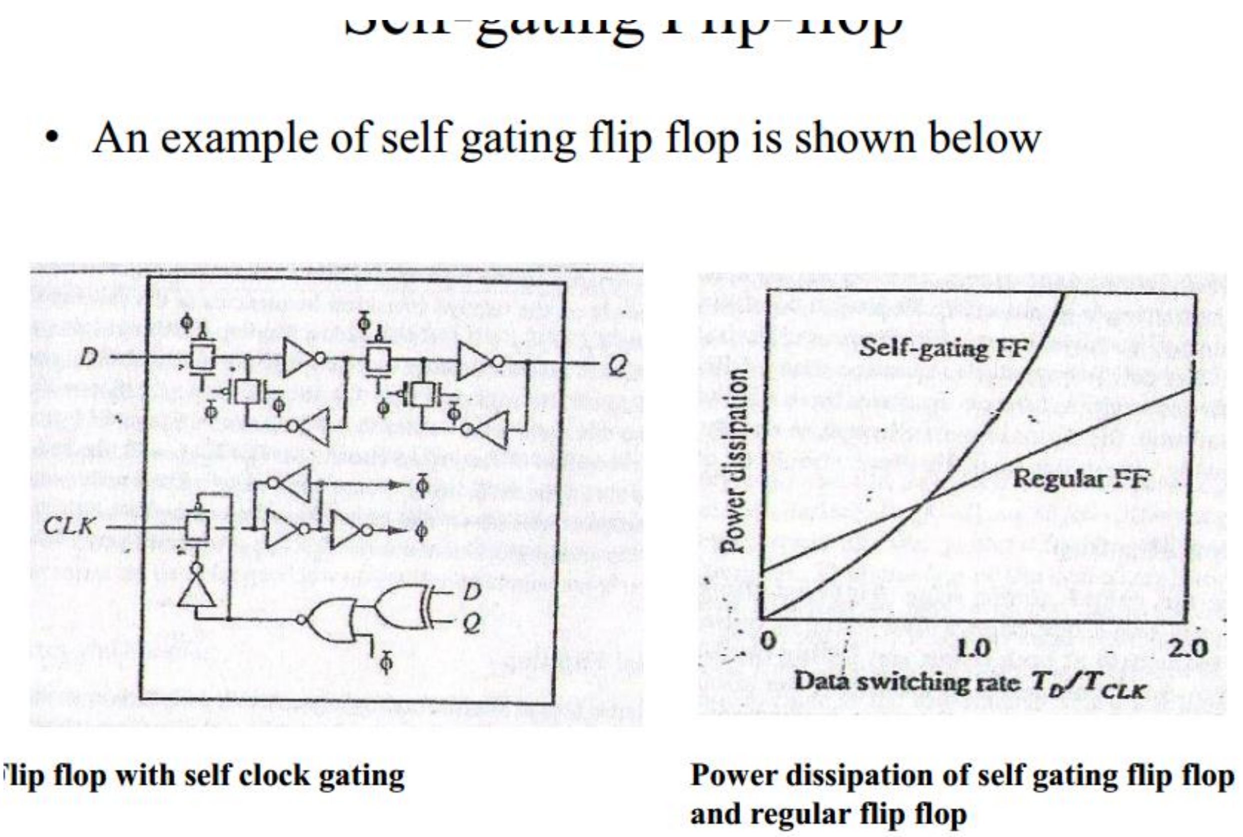
• Part of the clock energy of a flip flop is consumed by the internal clock buffer to control the transmission gates

• If the input of the flip flop is identical to its output, the switching of its clock signal can be suppressed to conserve power

• This is similar to clock gating techniques

• The difference is that the gating function is derived within the flip flop without any external control signal
• Power is saved by holding the internal clock signals of the flip flop when allowed
• The external clock signal of the flip flop still switches.

Self-gating Flip-flop



Self-gating Flip-flop

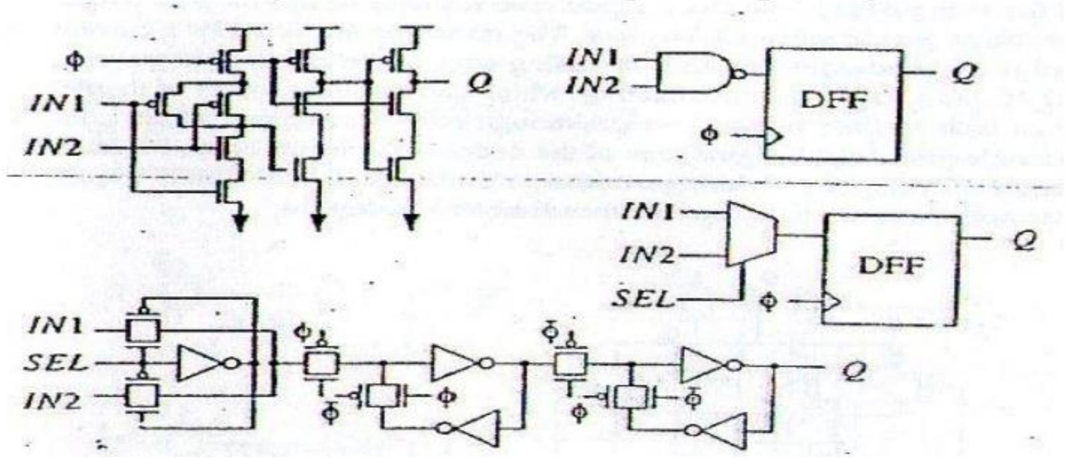
- An example of self gating flip flop is shown below

lip flop with self clock gating

Power dissipation of self gating flip flop and regular flip flop

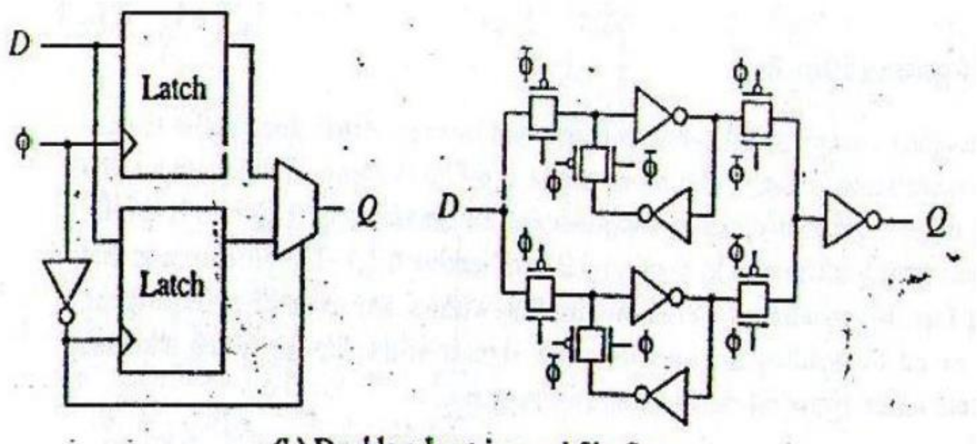
Combinational flip flop

- One way to reduce circuit size is to associate logic gates with a flip flop to produce a combinational flip flop
- Combinational flip flops are efficient because they are able to eliminate or share redundant gates
- In terms of area, power and delay, combinational flip flops are desirable but they increase the design complexity



Double Edge Triggered flip flop

- Compared to SETFF, the double edge triggered flip flop (DETFF) requires slightly more transistors to implement
- The flip flop retains its data when the clock signal is not toggling



(b) Double edge triggered flip-flop.

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

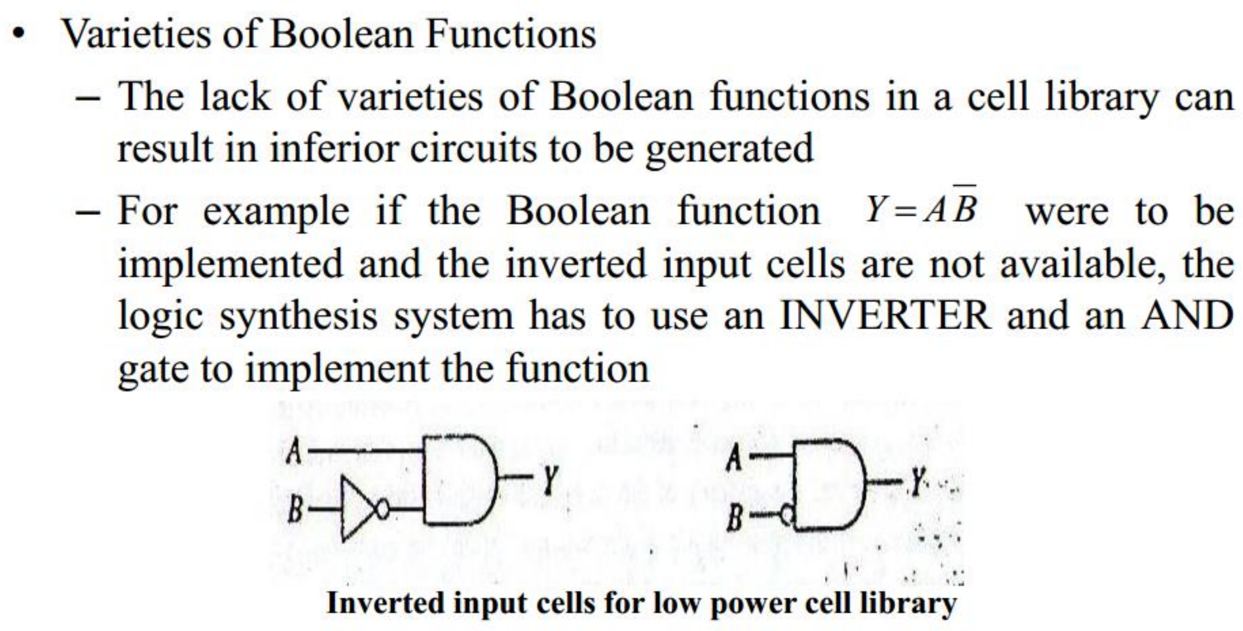
– In the top-down cell based design methodology, the tradeoff among
power, area and delay is performed by selecting the

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 $Y = A\overline{B}$ 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

Logic Design

- Logic design *was once the primary abstraction level where automatic design synthesis begins*
- *The most prevalent theme in logic level power optimization techniques is the reduction of* switching activities
- *Switching activities directly contribute to the* charging and discharging capacitance *and the* short circuit power
- *Some switching activities are the result of* unspecified or undefined behavior *of a logic system that are not related to its power operation*
- *Such* stray switching activities *should be eliminated or reduced if possible*
- *However, suppressing unnecessary activities usually requires* additional hardware logic *that increases the area and consumes power*
- *The challenge is to justify the low power techniques via* intelligent analysis.

Gate Reorganization

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 NAND gates 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.

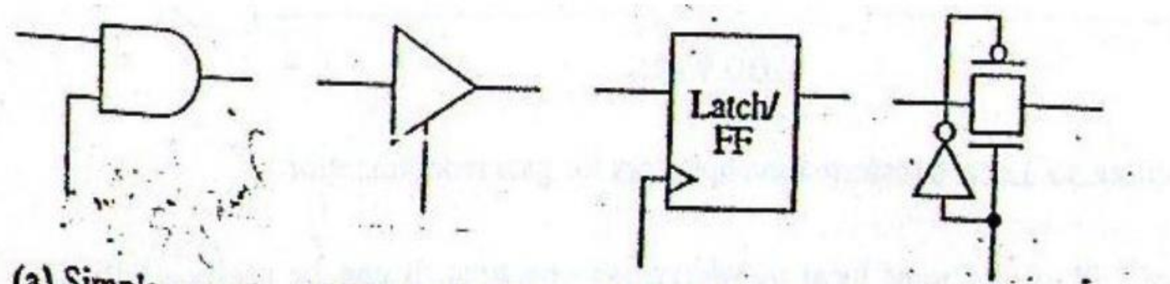
Signal Gating

- 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.

Signal Gating

- The various logic implementation of signal gating is shown below



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

- 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

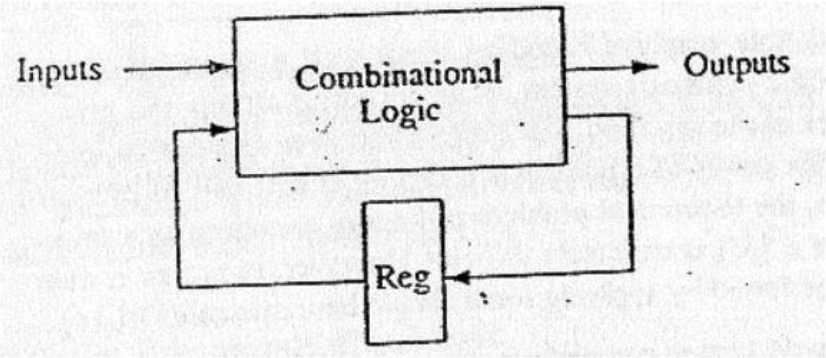
Logic Encoding

- 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 Eg. 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.

State Machine Encoding

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



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

State Machine Encoding

- 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

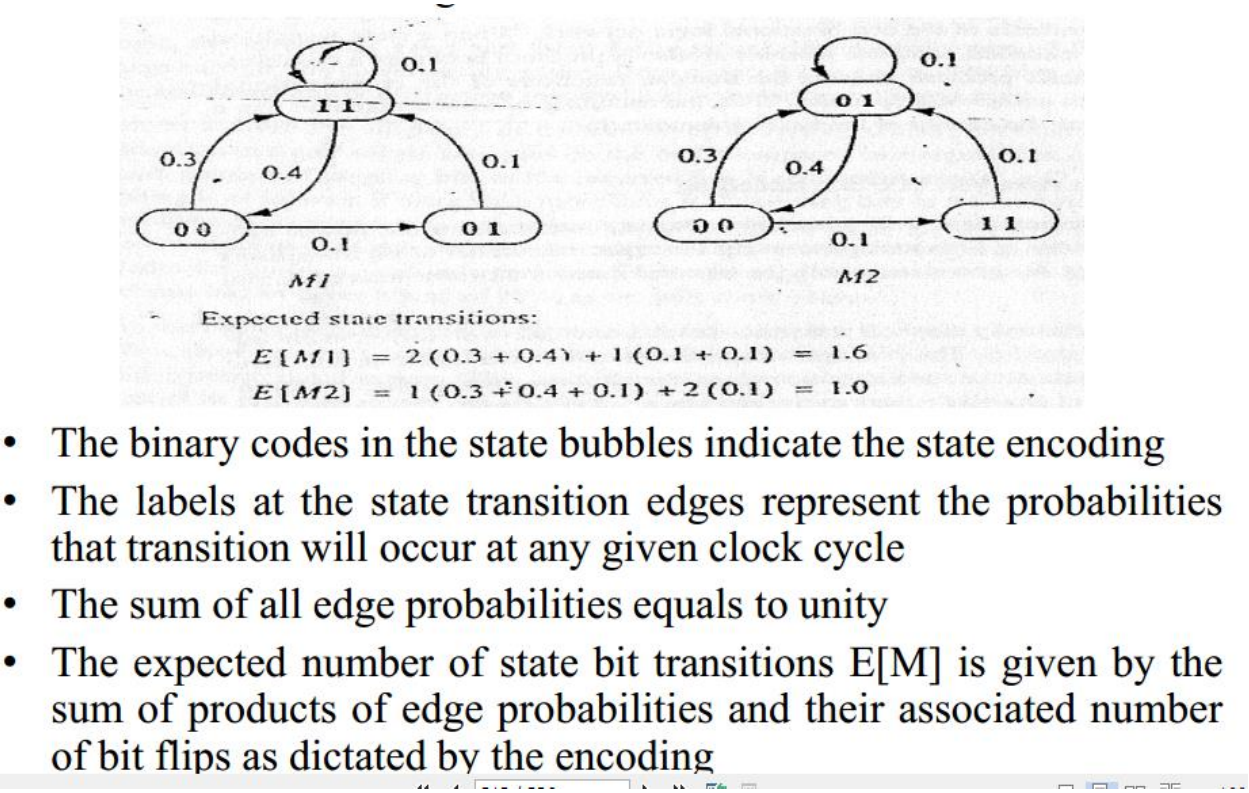
– Another parameter is the expected number of transitions of output signals

• 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:

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

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

- 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

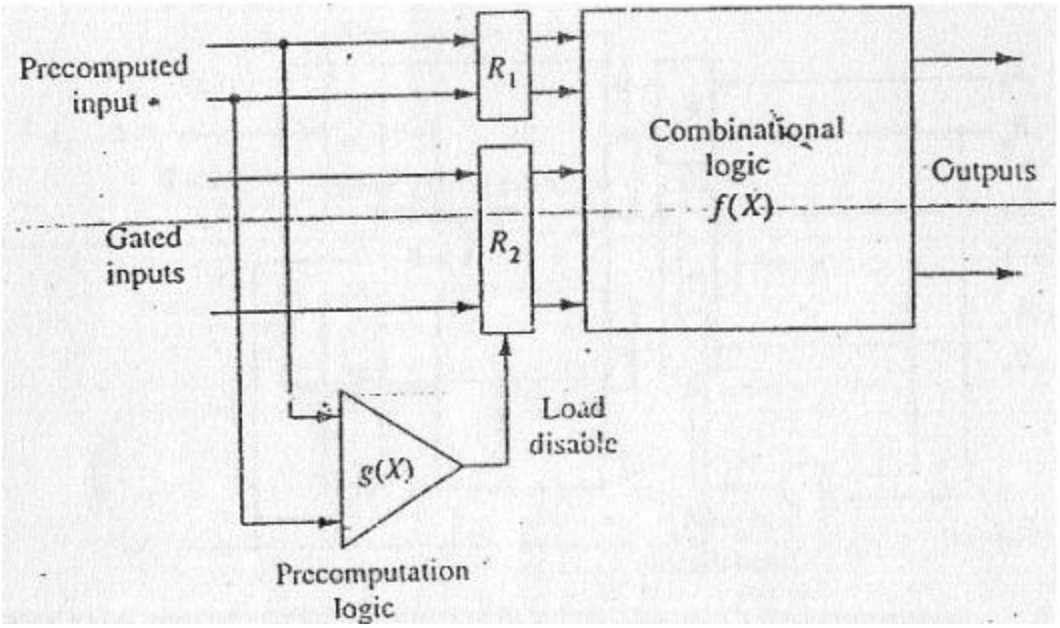
Precomputation Logic

• Precomputation logic optimization is a method to trade area for power in a synchronous digital circuit

• The principle of precomputation logic is to identify logical conditions at some inputs to a combinational logic that is invariant to the output

• Since those input values do not affect the output, the input transitions can be disabled to reduce switching activities

- One variant of precomputation logic is shown below.



Let R1 and R2 are registers with a common clock feeding a combinational logic circuit with a known Boolean function f(x)

- Due to the nature of the function f(x), there may be some conditions under which the output of f(x) is independent of the logic value of R2
- Under such conditions, we can disable the register

loading of R2 to avoid causing unnecessary switching activities, thus conserving power

- The Boolean function f(x) is correctly computed because it receives all required values from R1
- To generate the load disable signal to R2, a precomputation Boolean function g(x) is required to detect the condition at which f(x) is independent of R2
- g(x) depends on the input signals of R1 only because the load disable condition is independent t of R2, otherwise f(x) will depend on the inputs of R2 when the load disable signal is active.