Clock Gating

Clock Gating

- Power has become a primary consideration during hardware design. Dynamic power can contribute up to 50% of the total power dissipation.
- Clock gating is one of the power-saving techniques used in many synchronous circuits.
- Clock gating refers to adding additional logic to a circuit to prune the clock tree, thus disabling portions of the circuitry where flip flops do not change state.
- Clock-gating is the most common, and yet very effective RTL optimization for reducing dynamic power.

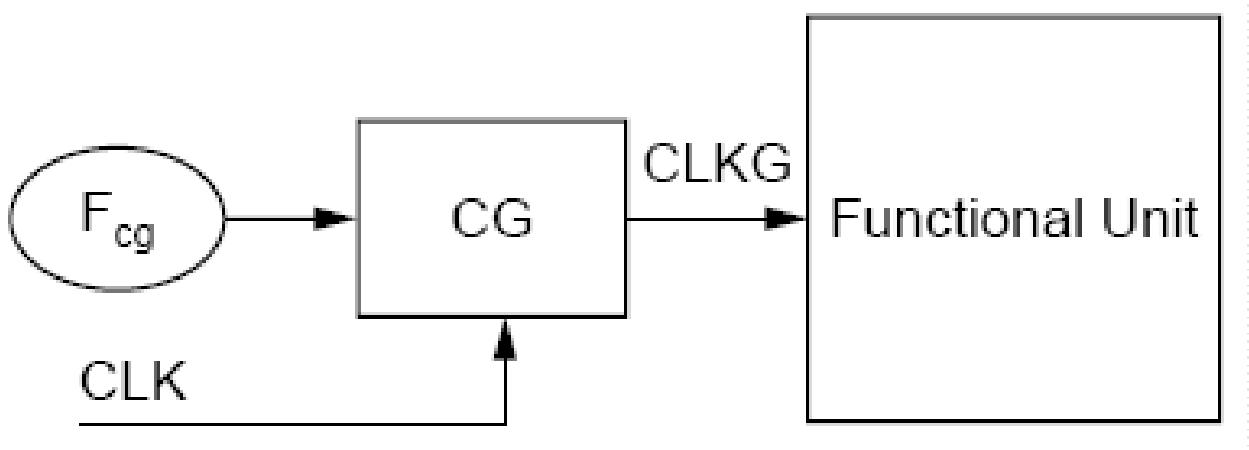
Clock Gating

- Effective clock-gating implementation requires skillful application and comprehensive verification.
- The savings are mainly due to the switching capacitance reduction in the clock network and the switching activity in the logic fed by the storage elements.
- There is a vast array of clock-gating techniques available to designers.
- Most clock-gating is done at the Register Transfer Level.

Clock Gating

- Clock-gating algorithms can be grouped into three categories:
  - **System-level**
  - **Sequential**
  - **Combinational**
- System-level clock-gating stops the clock for an entire block, effectively disabling all functionality.
- On the contrary, combinational and sequential clock-gating selectively suspend clocking while the block continues to produce output.

Clock Gating Principle



- A block CG, which inhibits the clock signal when the idle condition is true, is associated with each sequential functional unit.
- The clock signal is controlled by function F_{cg}.
- CLK is the system clock and CLKG the gated clock.
- Clock-gating techniques have been successfully implemented in many microprocessors.

Implementation of F_{CG}

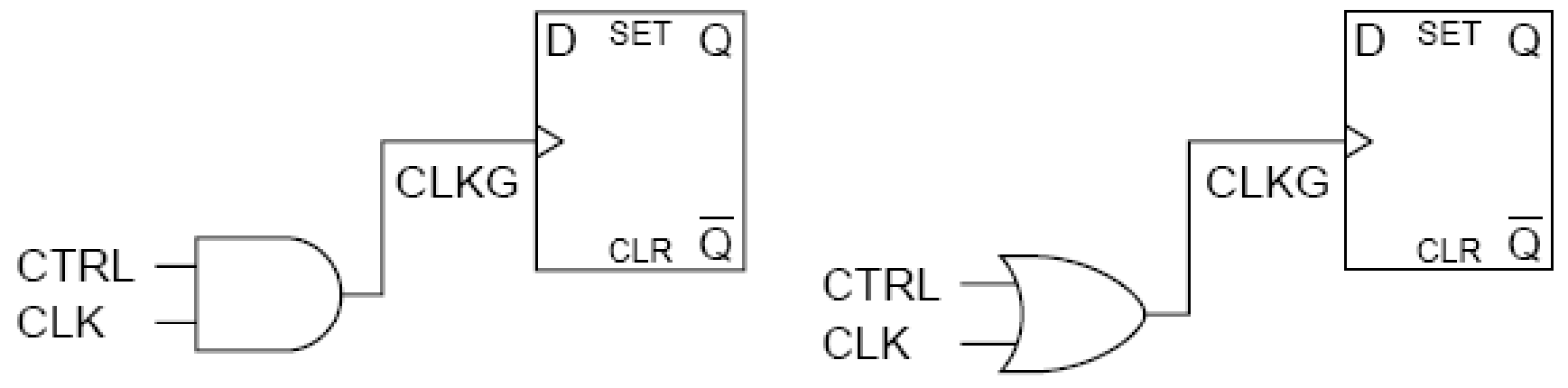
- Many implementations have been proposed for function CG (F_{CG}).

Combinational Clock Gating

- Combinational clock-gating is a straightforward substitution to the RTL code. It reduces power by disabling the clock on registers when the output is not changing.
- Opportunities to insert combinational clock-gating can be found by looking for conditional assignments in the code. Clock-gating logic is substituted when code like

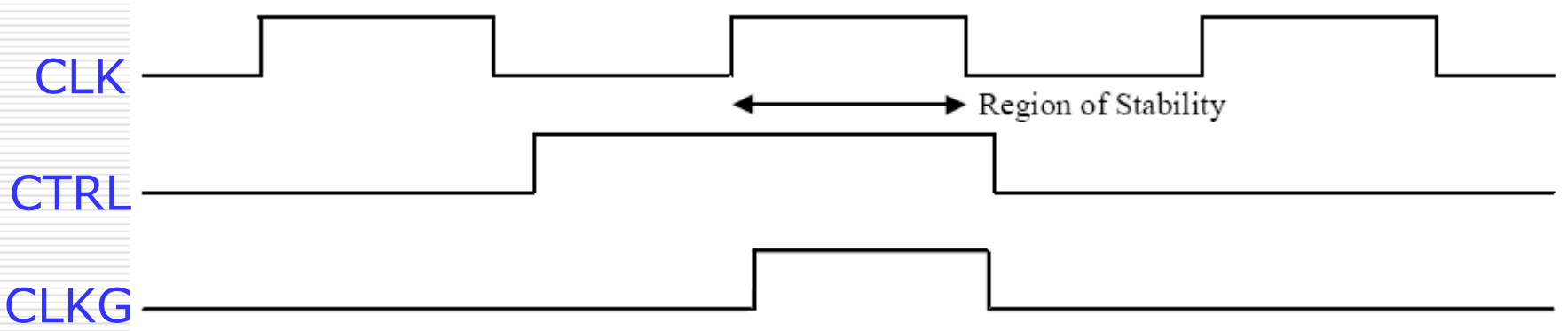
  "if (condition) out <= in" is present.
- Combinational clock-gating is now a feature in the RTL compilers. Power aware synthesis tools identify RTL coding patterns and make the appropriate substitution.
- H/w designers only need to understand some simple RTL coding guidelines to gain the benefits of combinational clock-gating.

Latch-Free Implementation



- The simplest one uses an AND or an OR gate

Latch-Free Implementation

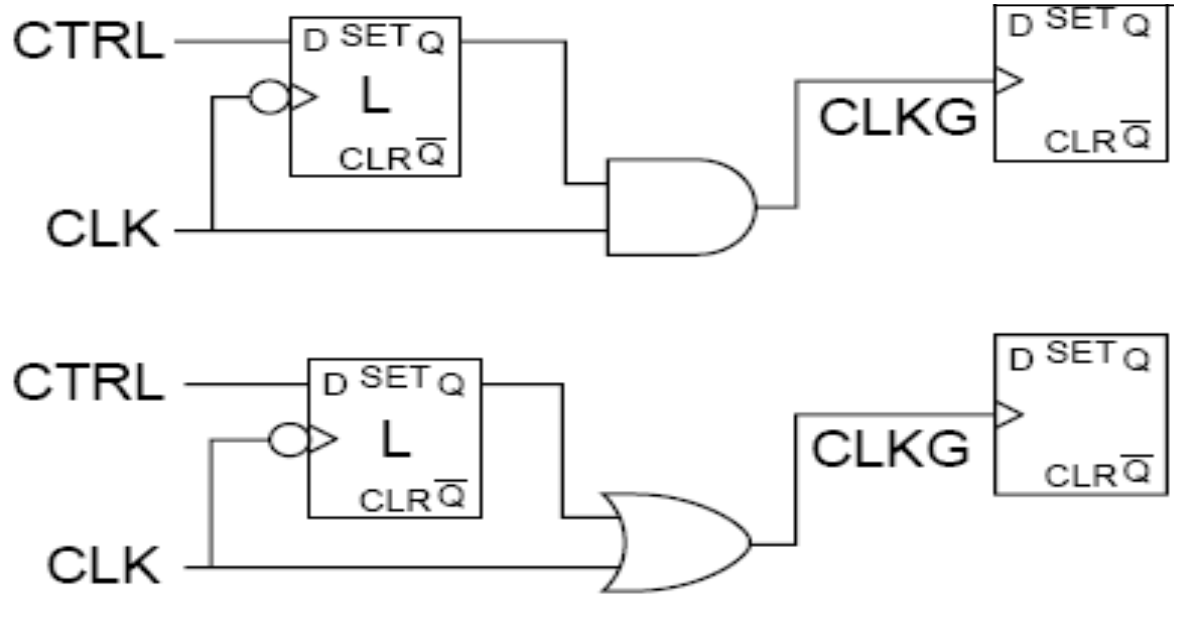


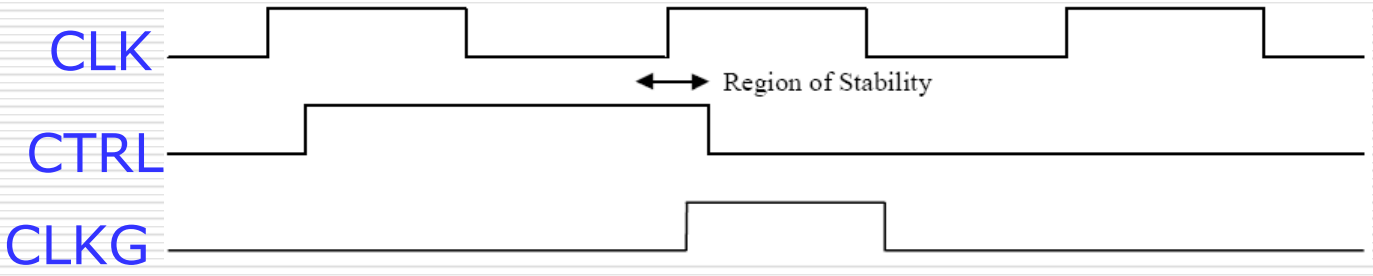
- This imposes a requirement on the circuit that all enable (CTRL) signals be held constant from the active (rising) edge of the clock until the inactive (falling) edge of the clock in order to avoid truncating the generated clock pulse prematurely or generating multiple clock pulses where one is required.
- This is not efficient because of the possible spikes or glitches at the output of the gate.

Latch-based CG block Implementation

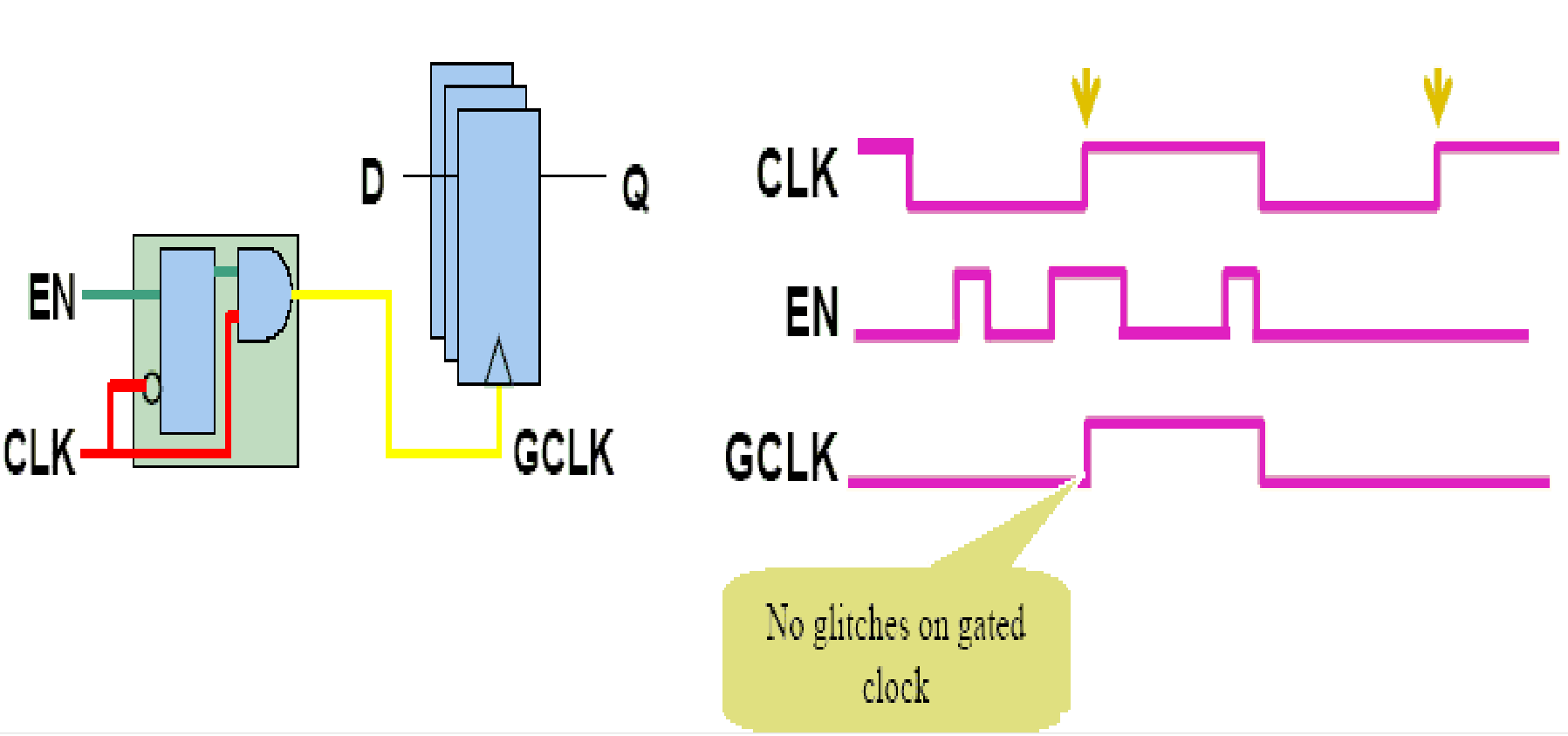
- An alternative and better solution is shown next. With this configuration, the spurious transitions generated by function F_{cg} are filtered.
- Since the latch captures the state of the enable (CTRL) signal and holds it until the complete clock pulse has been generated, the enable signal need only be stable around the rising edge of the clock.

Latch-based CG block Implementation



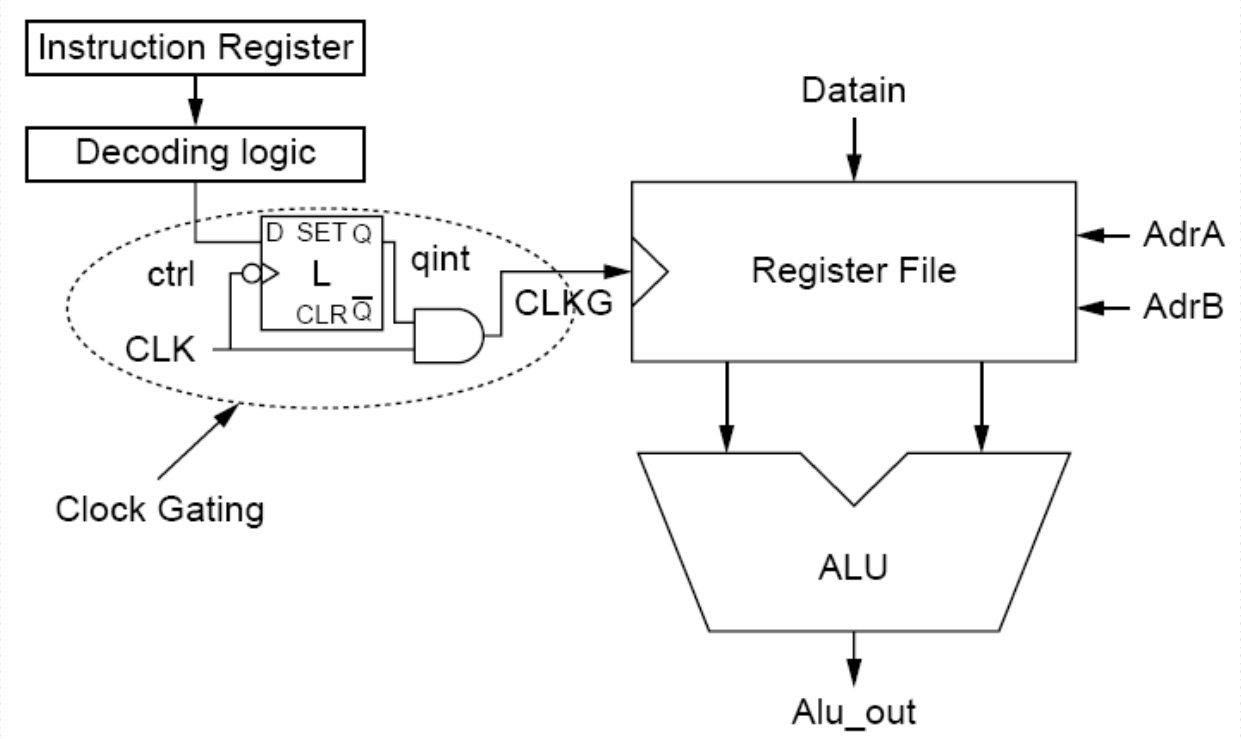


Latch-based CG block Implementation



Combinational Clock Gating Example:

Clock Gating of datapath register:

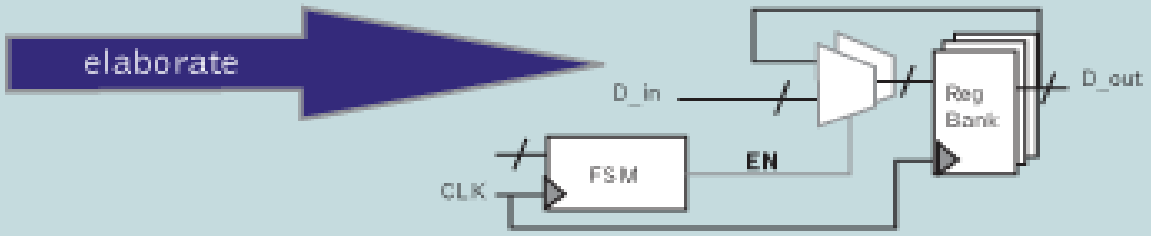


In some applications, conditionally executed parts of the HDL code can be identified and separated. Clock gating can be applied for each part.

Example: Power optimization during synthesis — Power Compiler automatically inserts clock-gating circuits.

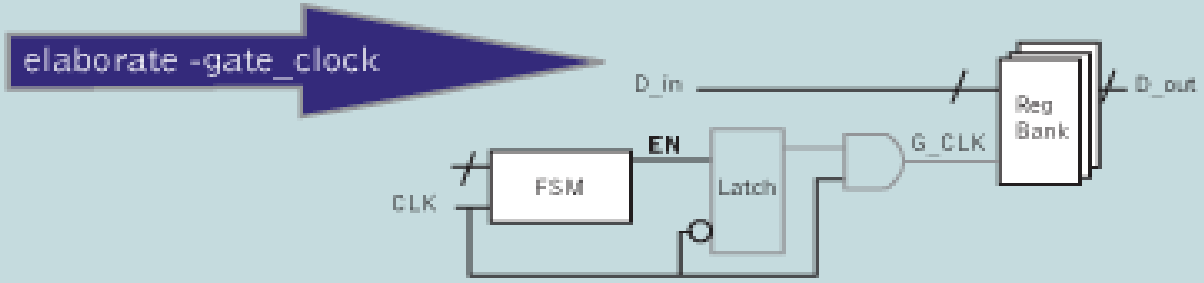
```
always@(posedge CLK)
  if (EN)
      D_out = D_in
```

Synchronous Load-enable implementation without Clock Gating




```
always@(posedge CLK)
  if (EN)
      D_out = D_in
```

Synchronous Load-enable implementation with Clock Gating



Combinational Clock Gating

– Merits and Demerits

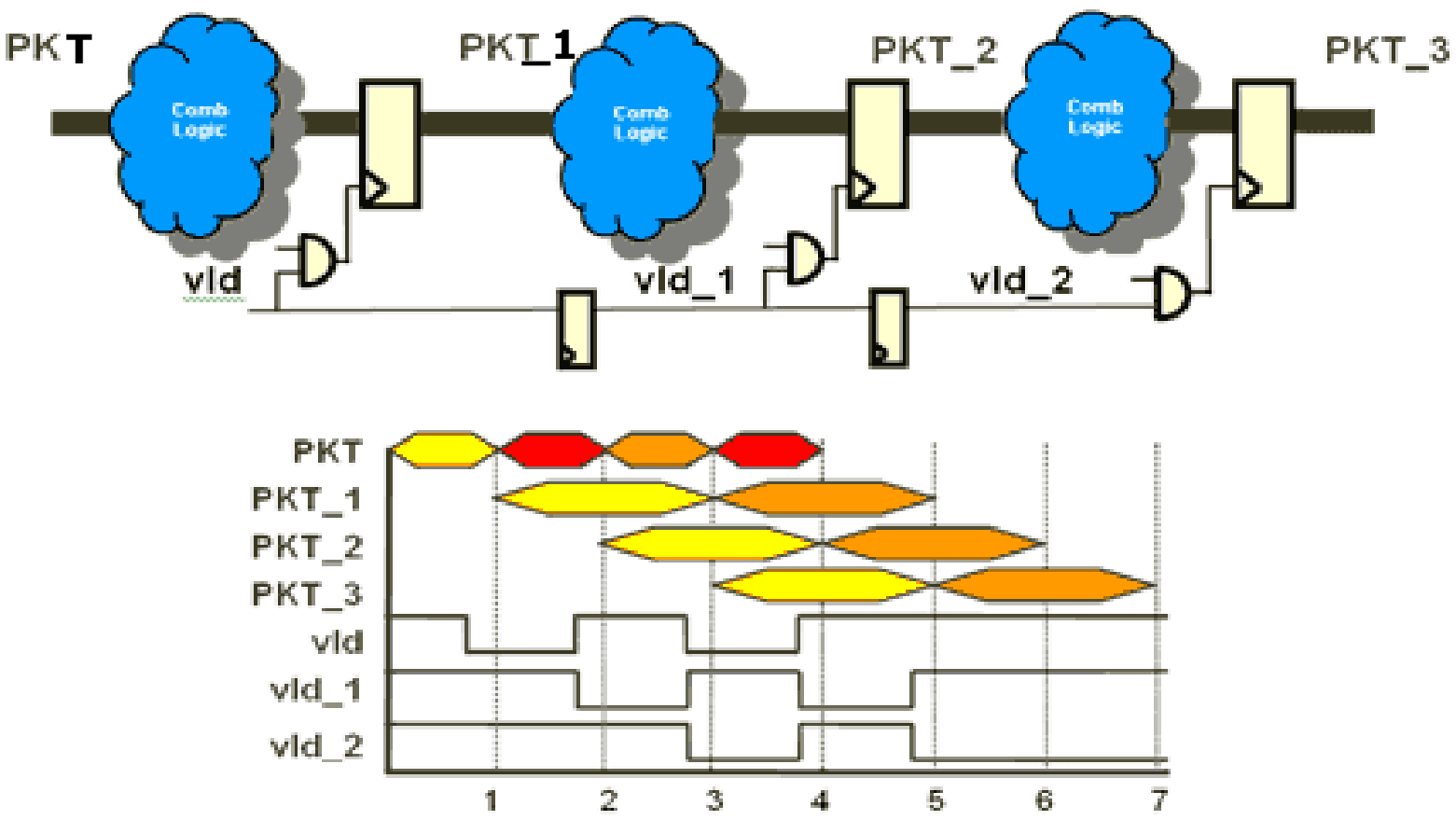
- Combinational clock gated flops maintain a one to one state mapping with the original RTL. This makes verification simple to setup and comprehensive.

- On the other hand because switching activity is eliminated only when data is not changing, the actual power savings is limited. In typical designs, combinational clock-gating can reduce dynamic power by about 5% to 10%.

Sequential Clock Gating

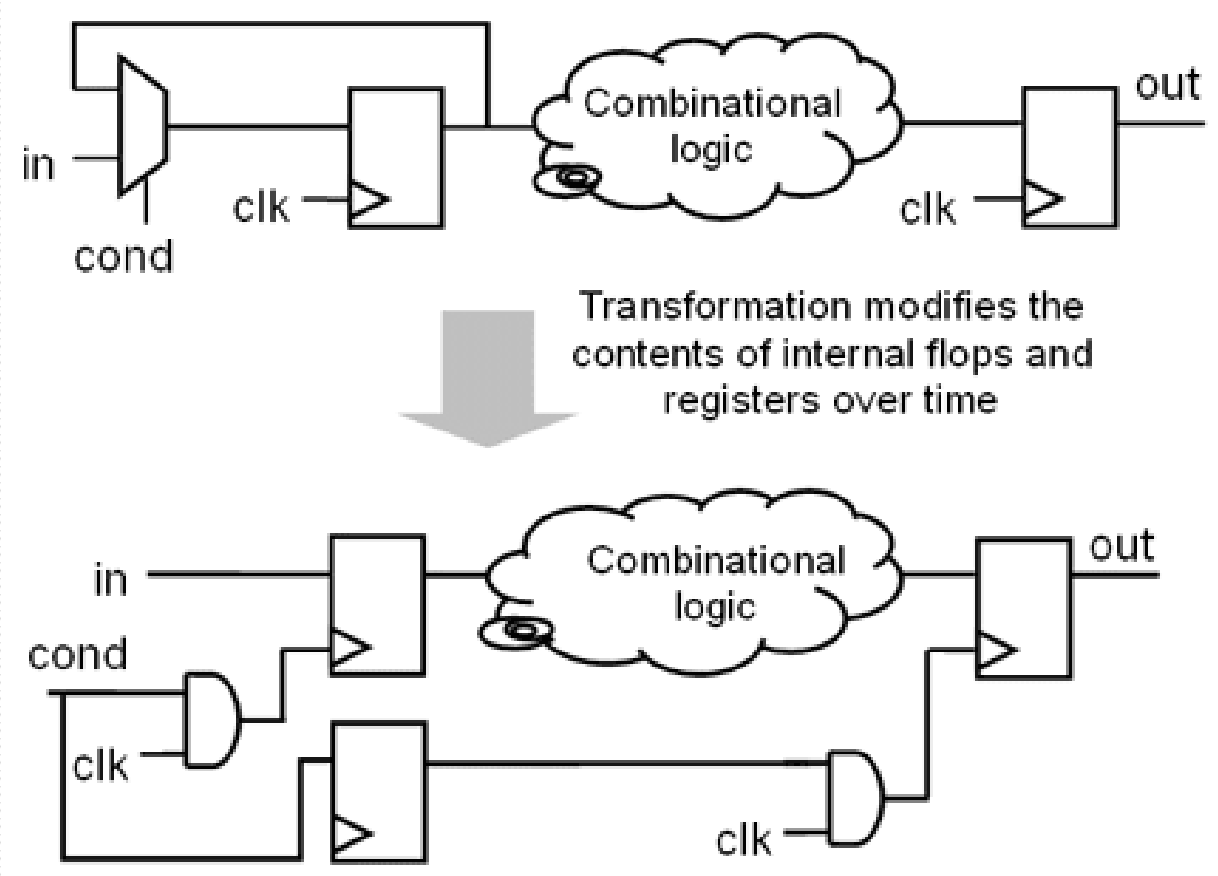
- Power is optimized by identifying unused computations, data dependent functions and don't-care cycles in the original code.
- There are many types of sequential clock-gating transformations.
- Identifying opportunities for sequential clock-gating is difficult, requiring sequential analysis.
- One example of a sequential optimization is turning off subsequent pipeline stages based on a propagated valid condition.

Disabling pipeline stages when the output is not used



Note : The active clock edge occurs just before the ‘vld’ signal changes state

Sequential Clock Gating



- sequential clock-gating can save significant power, typically reducing switching activity by 15-to-25% on a given block.

Clock-Gating Efficiency

- A typical metric used to measure the effectiveness of clock gating is the percentage of registers in a design that are clock gated.
- This gives designers an indication of the number of clock-gated registers in the design.
- Clock-gating efficiency is defined as the percentage of time a register is gated for a given stimulus or switching activity.
- The average clock-gating efficiency can be computed as the average of all clock-gating efficiencies in the design.

Clock-Gating Efficiency

Example: