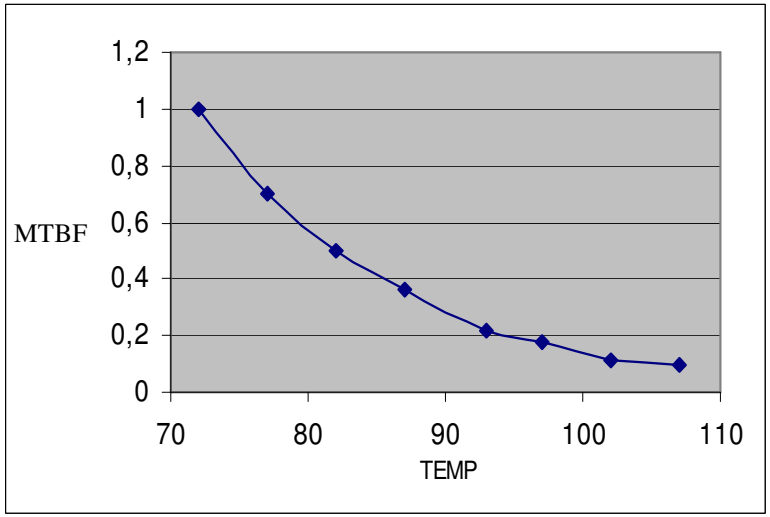
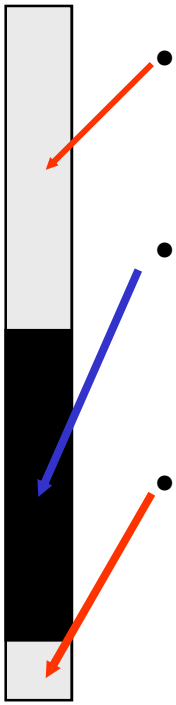
Principles of low power digital design

- Why low power?
  - Decrease cost of packaging, power supply, cooling system,...
  - Increase autonomy and portability
  - Save energy
  - Limit the maximum power demand (or current) from the power supplies
  - Increase reliability: MTBF (*Mean Time Between Failure*) decreases with T

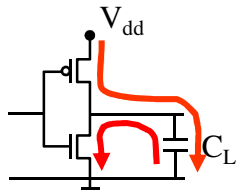


Sources of energy consumption

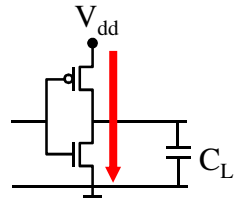
- CMOS digital tecnology
  - Average electric power: Ptotal = Pswitch + Pleak + Pshort



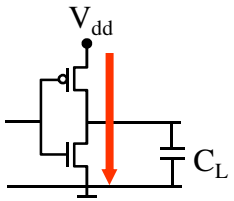
- Pswitch – charging/discharging of the load capacitances



- Pleak – transistor leakage

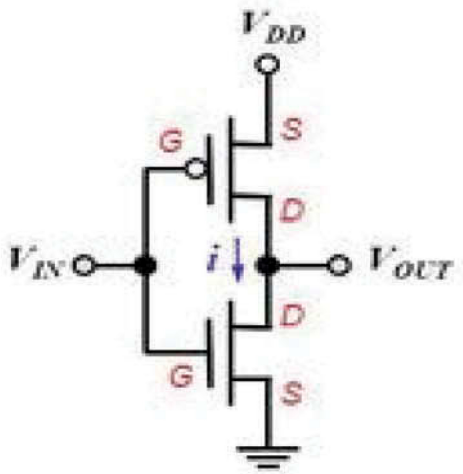


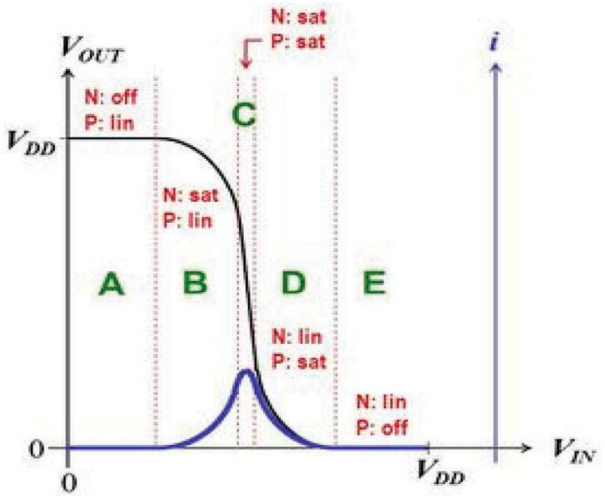
- Pshort – short circuits V_{dd} to Gnd during switching



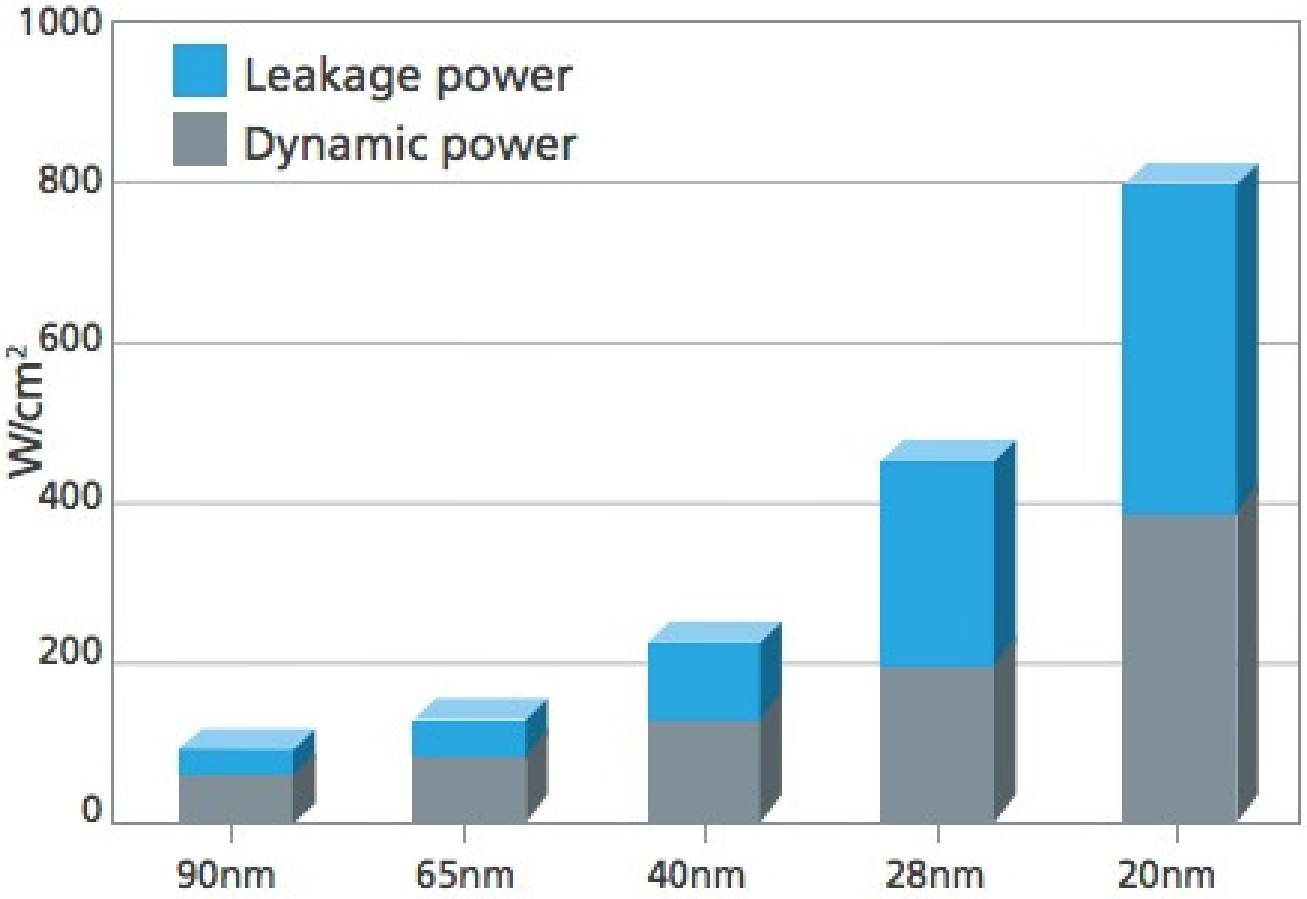
How to reduce the power consumption?

- **Pleak (or static power)**
  - Subthreshold leakage, gate to substrate reverse-bias leakage
  - Depends on the physical characteristics of technology
    - Reducing the technology node from 28nm (planar) to 12 nm (finFET): -50% of leakage
  - Pleak is roughly proportional to the *size* of the circuit
- **Pshort**
  - Due to the limited transition time between ON an OFF states
  - Pshort depends on the switching activity and rise/fall times of inputs



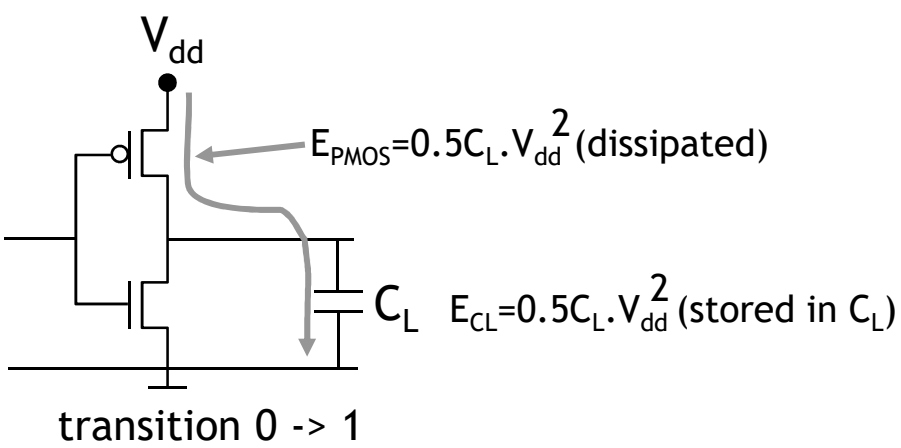


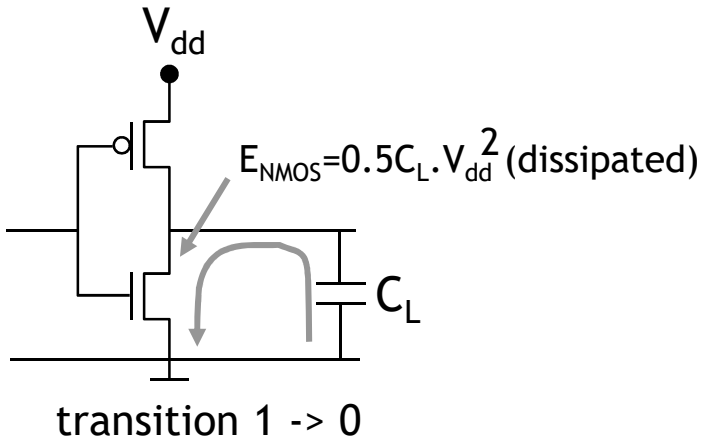
Leakage and active power



Low power design

- Dynamic power consumption – depends on the circuit activity
  - Pswitch = α . $C_L.V_{dd}^2.f_{clk}$
    - C_L = load capacitance
    - V_{dd} = power supply
    - α = average transitions per clock cycle
    - f_{clk} = clock frequency
  - α . C_L = $C_{efective}$ effective load capacitance





How to reduce Pswitch?

$Pswitch = \alpha \cdot C_L.V_{dd}^2.f_{clk}$

- Vdd: power depends on Vdd^2
  - 5V to 3.3V: 43%$Pswitch_{5V}$
  - 3.3V to 1.8V: 31% $Pswitch_{3.3V}$
  - 5V to 1.2V: 6% $Pswitch_{5V}$
- Switching activity: α and/or f_{clk}
  - Use clock frequency as low as possible (gated clocking is mandatory!)
  - Isolate glitching propagation through combinational circuits
  - Exploit low switching topologies and appropriate data encoding
- C_L: Load capacitance C_L

Switching activity and logic function

- NOR gate with random inputs
  - P(0->1) = P(1->0) = p(1)*p(0) = 1/4*3/4 = 3/16 = **0.19**
  - P(0->0) = 3/4* 3/4 = 9/16
  - P(1->1) = 1/4 * 1/4 = 1/16
- XOR gate with random inputs
  - P(1->0) = P(0->1) = 1/2 * 1/2 = 1/4 = 0.25
  - P(0->0) = P(1->1) = 1/2 * 1/2 = 1/4 = 0.25
- A complex logic function can be realized with different circuits
  - The transition probabilities of the internal nodes may vary
- In general the inputs have different probability of switching

Switching activity and logic function

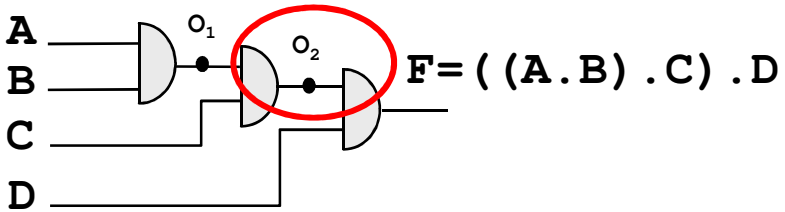
- NOR gate with random inputs
  - **P(0->1) = P(1->0) = p(1)*p(0) = 1/4*3/4 = 3/16 = 0.19**
  - P(0->0) = 3/4* 3/4 = 9/16
  - P(1->1) = 1/4 * 1/4 = 1/16

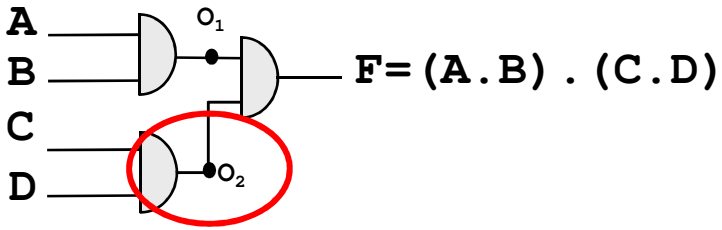
Output of a XOR gate has higher activity

- XOR gate with random inputs
  - **P(1->0) = P(0->1) = 1/2 * 1/2 = 1/4 = 0.25**
  - P(0->0) = P(1->1) = 1/2 * 1/2 = 1/4 = 0.25
- A complex logic function can be realized with different circuit topologies
  - Transition probabilities of the internal nodes may vary
- In general the inputs have different switching probabilities
  - Compare two's complement to sign and magnitude in a digital signal

Static switching analysis

two topologies for the same function

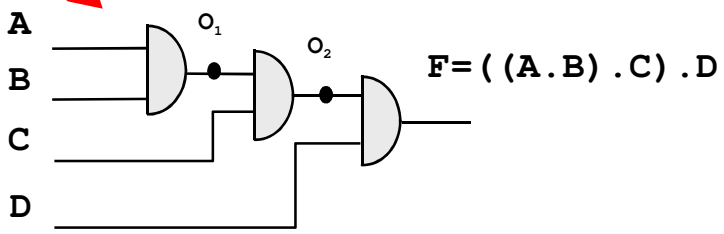


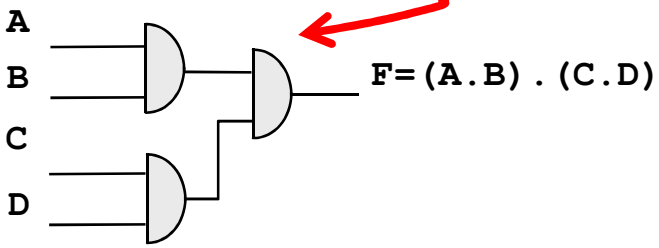


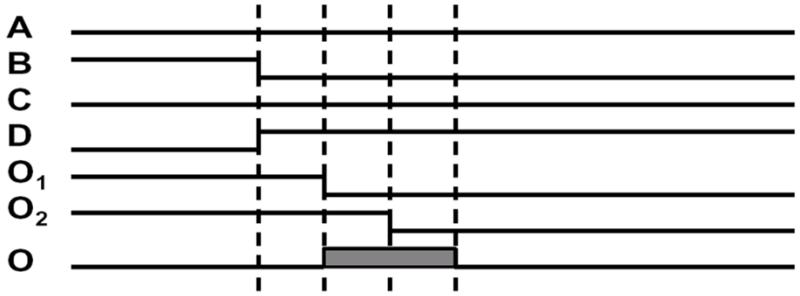
	O1	O2
P(0->1) cascaded	3/16=0.19	7/64=0.11
P(0->1) tree	3/16=0.19	3/16=0.19

Dynamic switching analysis

- Delays induce glitching in the outputs (unwanted transitions)
- Glitches propagate through the circuits
  - 4-input AND implemented as **((A.B).C).D**
    - The transition **1110->1011 generates a glitch**
  - 4-input AND implemented as **(A.B).(C.D)**
    - If the AND gates are ballanced, **1110->1011 is glitch free**

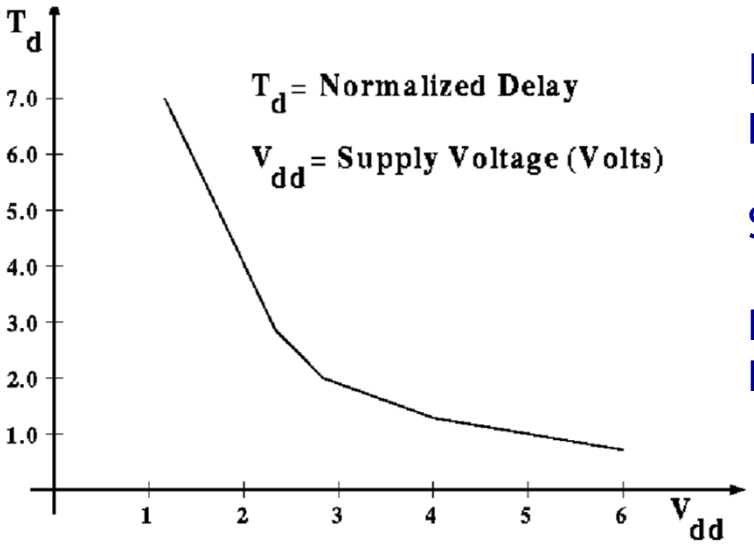






Reducing Vdd

- Pswitch depends on V_{dd}^2
  - But the propagation delay decreases with V_{dd} : $$T_d = \frac{C_L V_{dd}}{K(W/L)(V_{dd} - V_{th})^2}$$



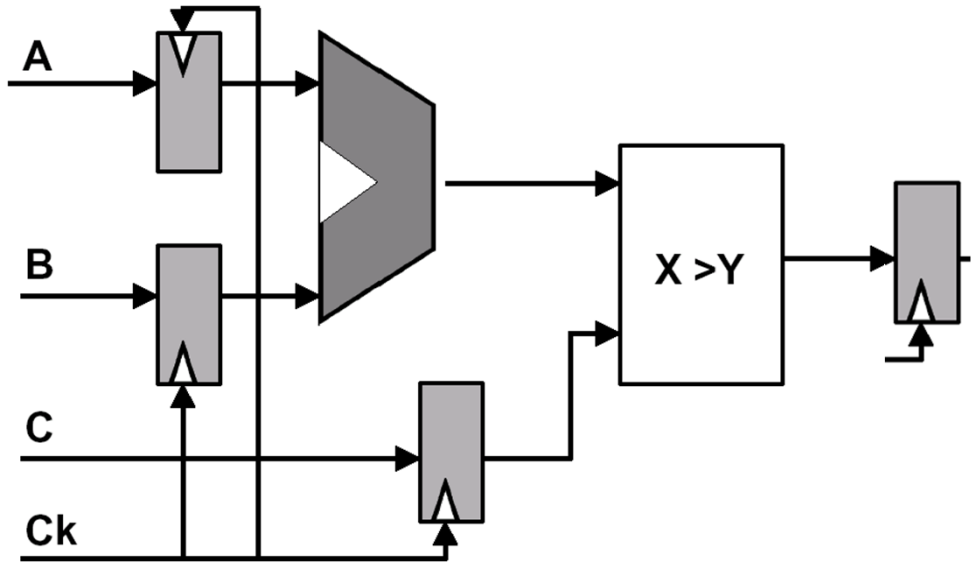
Improve the system architecture using pipelining or parallel processing

Some CMOS technologies provide dual voltage cells

Power gating: switch off Vdd or to a low voltage Level, high enough to keep the state of FFs

Architectural optimizations

- Reference circuit

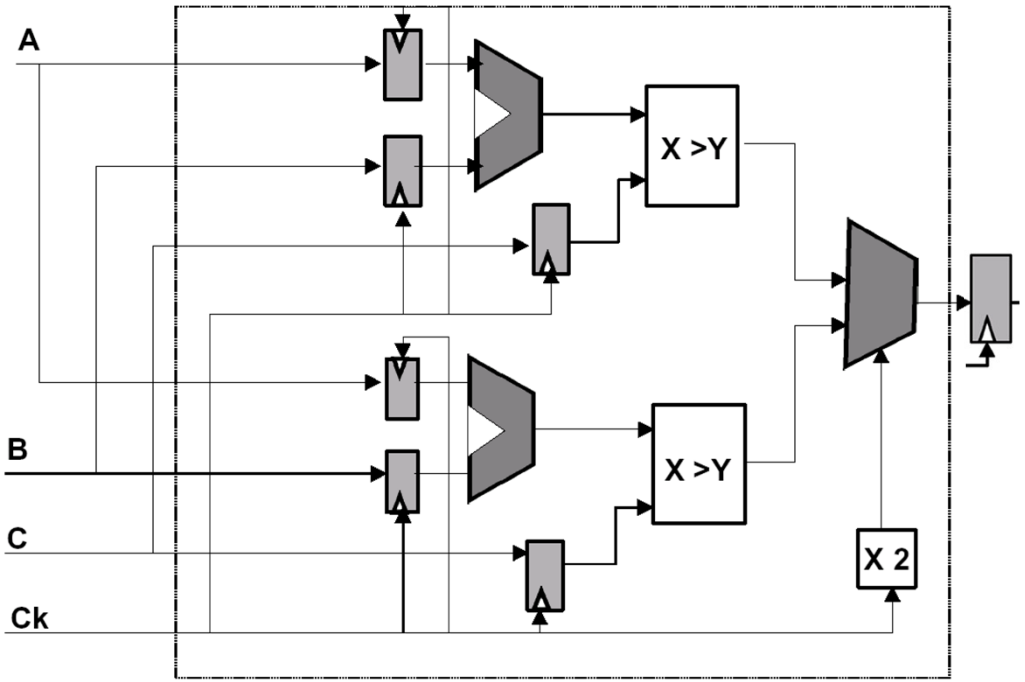


Supply voltage: Vdd_{ref}=5V
Maximum propagation delay: 25ns (adder and comparator)
Minimum clock period: 25ns (f_{ref}=1/25ns = 40 MHz)
Total effective capacitance: C_{ref}
Switching power: P_{ref} = ½ $C_{ref}.Vdd_{ref}^2.f_{ref}$

If the circuit operates at 40 MHz Vdd cannot be reduced as this would increase the delay

Based on Chandrakasan, A. P., Sheng, S., & Brodersen, R. W. (1992). Low-power CMOS digital design. IEICE Transactions on Electronics, 75(4), 371-382.

Parallel circuit (DDR operation)



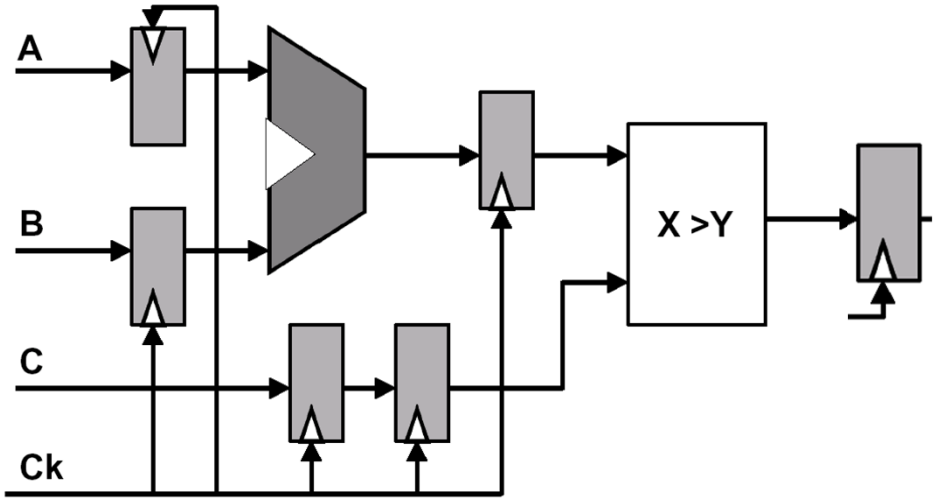
The same troughput with a propagation delay of 50ns (Fclock = 20 MHz)

To increase the propagation delay **Vdd can be reduced to Vdd_{par}= 0.58 Vdd_{ref} (2.9V)**

$C_{par} = 2.15C_{ref}$ (above 2X due to the increased logic)

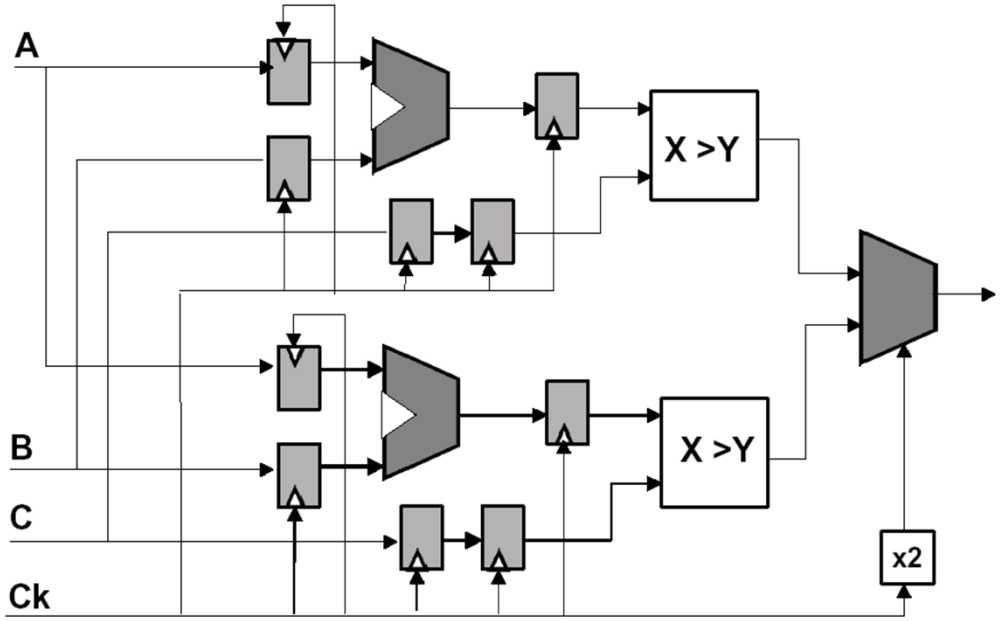
$P_{par} = ½ . (2.15C_{ref}).(0.58Vdd_{ref})^2 f_{ref}/2 \approx 0.36P_{ref}$

Pipelined circuit



With a perfectly ballanced pipeline the clock frequency can increase to 80 MHz
Vdd can decrease to increase the propagation delay to the original Fclock
The same throughput using the same clock frequency ($f_{pipe}=f_{ref}$)
$Vdd_{pipe}= 0.58Vdd_{ref}$ (2.9V)
$C_{pipe} = 1.15C_{ref}$
$P_{pipe} = ½ . (1.15C_{ref}).(0.58Vdd_{ref})^2 f_{ref} \approx 0.39P_{ref}$

Combining parallel and pipelined



The adder and comparator **may have now only ¼ of the original propagation delay**

$f_{pp} = f_{ref} / 2$

$Vdd_{pipepar} = 0.4Vdd_{ref}$ (2.0V)

$C_{pipepar} = 2.5C_{ref}$

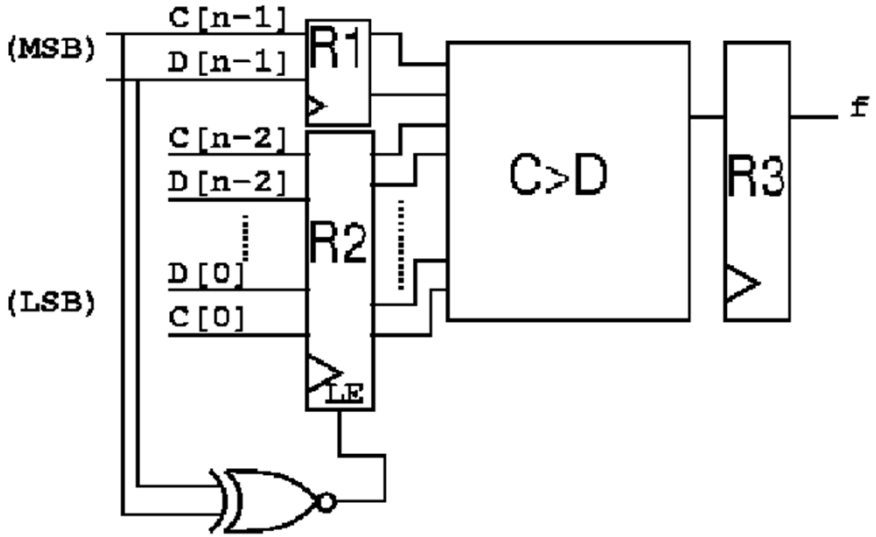
$\mathbf{P_{pipepar} = ½ \cdot (2.5C_{ref}).(0.4Vdd_{ref})^2 f_{ref} / 2 \approx 0.2P_{ref}}$

Summary

Architecture	V_{dd}	C_{Eff}	Area	Power
Reference	5V	1	1	1
Parallel	2.9V	2.15	3.4	0.36
Pipelined	2.9V	1.15	1.3	0.39
Parpipe	2.0V	2.50	4.0	0.2

Pre-computation

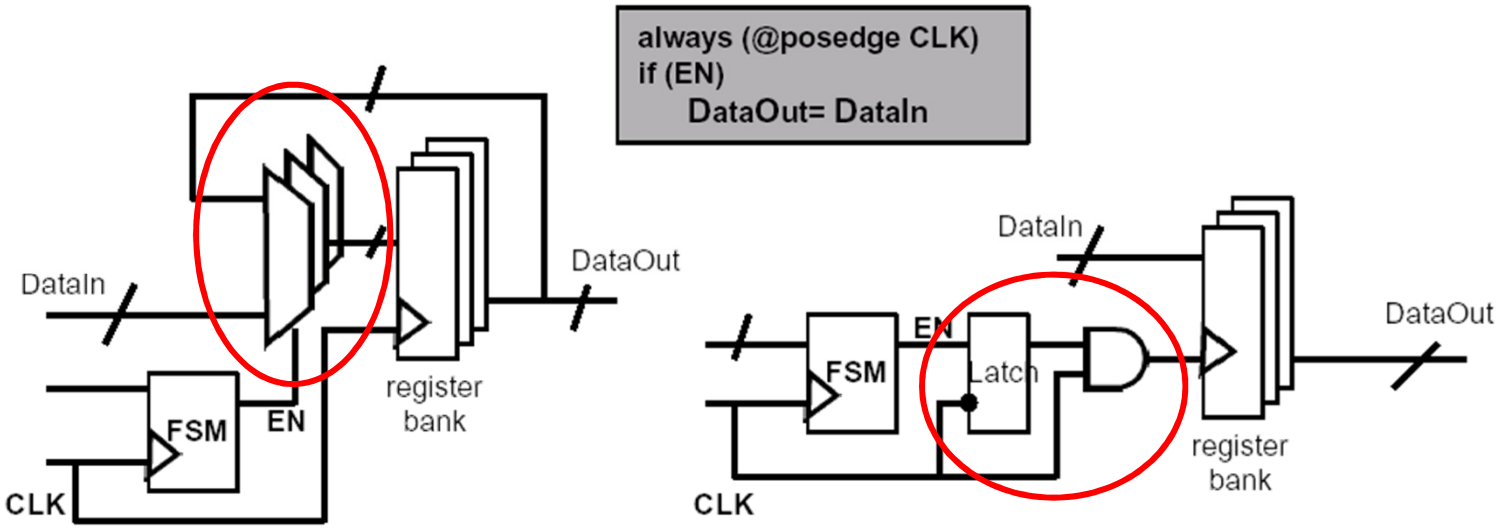
- Compute a result using only part of the operands
- Do not propagate the part of the operands not needed
- Example: magnitude comparator of unsigned numbers
  - The results can be decided for 50% of the possible inputs using only the MSBits: if C_{n-1} and D_{n-1} are diferente, R2 does not need to be reloaded, avoiding the propagation of switching activity



Clock gating

- Switch off the clock signal when not needed
  - Registers usually do not update their state in every clock transition
  - The clock signal is continuously switching and feeding all the flip-flops
    - Power is wasted in the clock line when registers are not being updated
    - A synchronous clock enable can disable the register load but keeps the clock running
      - `always @(posedge clk) if (en) Q <= D`
  - Switching on/off the clock cannot be done with an AND gate in the clock path
    - Need to avoid glitching and guarantee the synchronous operation
  - Synthesis tools and libraries provide resources for building clock gating
- Examples
  - In FSMs, if the keep-in-state condition is true the clock may be switched off
  - Banks of configuration registers are only updated for setting up parameters
  - Sequential functional units only require the clock when performing a calculation

Clock gating



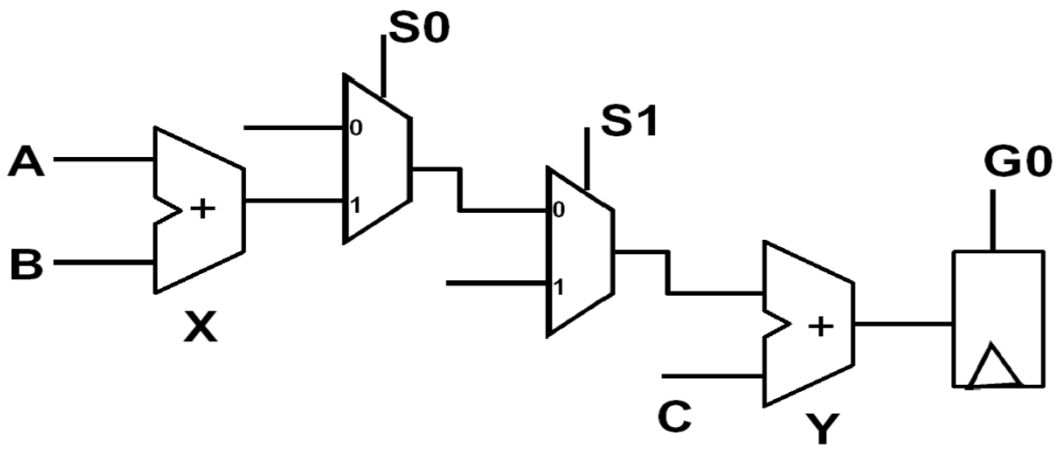
Synchronous clock enable:
A 2÷1 mux is required for every FF

Clock gating:
A single clock gating control creates
a gated clock that may feed a set of FFs

Clock gating may also save silicon area

Operand isolation

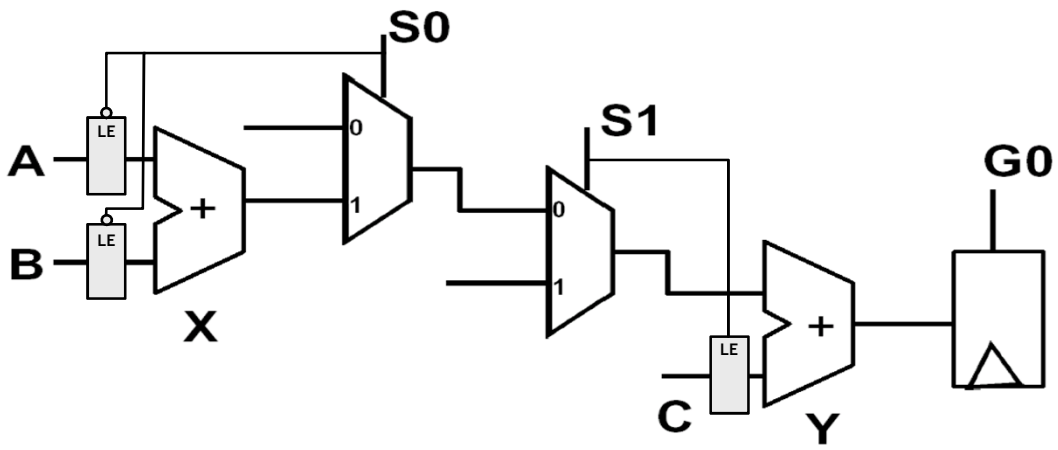
- transparent latches can block propagation of unnecessary data



If S0=0 or S1 = 1 the adder inputs A and B may be blocked
If G0 is disabled, A, B and C may be blocked.

Operand isolation

- transparent latches can block propagation of unnecessary data



If S0=0 or S1 = 1 the adder inputs A and B may be blocked
If G0 is disabled, A, B and C may be blocked.

Power gating

- Switch Vdd to zero when a block is not needed
- Control Vdd for different *power islands*
  - *ON state, high performance (high Vdd)*
  - *ON state, low performance (lower Vdd)*
  - *Retention state (low Vdd, keep state of registers)*
  - *OFF state (zero Vdd, state is lost)*
- Power management unit
  - Always on, controls the power state of each power island
- Interface between ON and OFF domains
  - Isolation cells
  - Retention cells