MŰEGYETEM 1782

Budapest University of Technology and Economics
Department of Electron Devices

Technology of IT Devices

Lecture 4

Logic gates

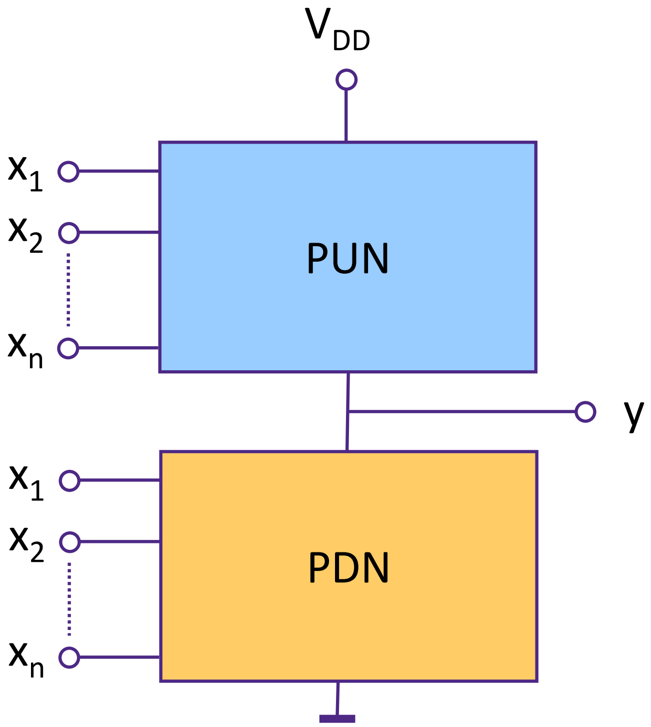
Registers

High-speed CMOS logic

CMOS logic gates

- Every CMOS logic gates is comprised of a pull-up network (PUN) consisting of pMOS transistors and a pull-down network (PDN) consisting of nMOS transistors.
- The number of transistors in the PUN and PDN is equal to the number of inputs
- For the input combinations that yield a logic 0 output, the PDN shorts the output node to the ground while the PUN is an open circuit.
- For logic 1 inputs the PUN connects the output to VDD and the PDN is an open circuit.

Summarized…



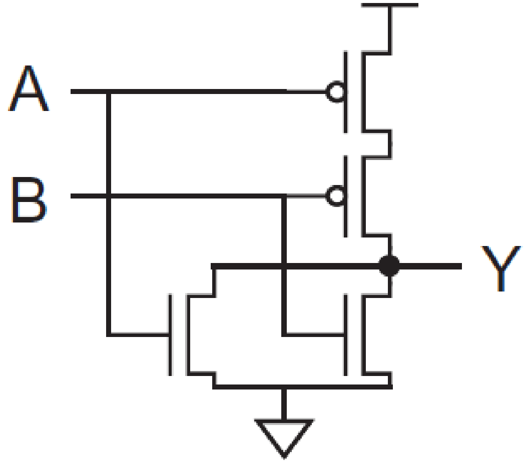
$$\boxed{y = f(\mathbf{X})}$$

- The PUN and the PDN are dual networks. This means that when two transistors are connected in series in one of the networks, their counterparts will be connected in parallel in the other network, and vice versa.
- pull up network
  - pMOS transistors
  - Short circuit, if f(**X**)=1
  - Open circuit, if f(**X**)=0
- pull down network
  - nMOS transistors
  - Short circuit, if f(**X**)=0
  - Open circuit, if f(**X**)=1

CMOS NOR gate

- The pull down network consists of two nMOS connected in parallel
- If both inputs are low, both nMOS transistors are closed and both pMOS transistors are open, so the output is high.
- When any of the inputs is logic 1, at least one of the nMOS transistors is open, and at least one of the pMOS transistors is closed: the output is connected to the ground.

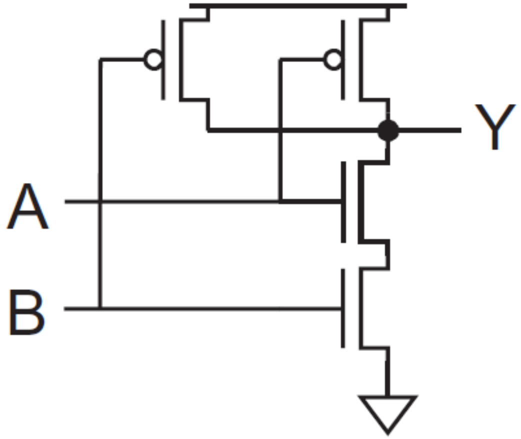
$$\boxed{Y = \overline{A + B} = \bar{A} \cdot \bar{B}}$$



CMOS NAND gate

- When both inputs are high, both nMOS transistors are open so they connect the output to the ground, while both pMOS transistors are closed.
- When any of the inputs is low, one of the nMOS transistors are closed, so there is an open circuit between the output and the ground, but at least one of the pMOS transistors is open so the output is connected to VDD

$$Y = \overline{AB} = \bar{A} + \bar{B}$$

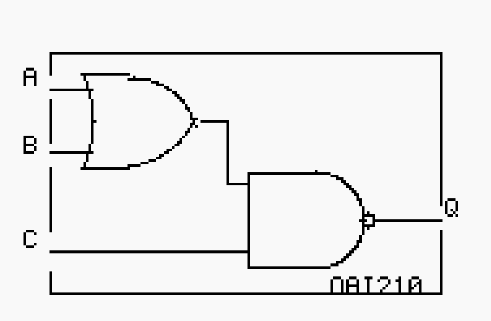


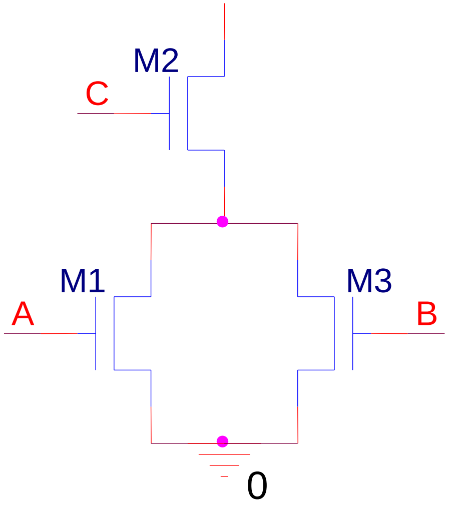
Complex logic gates

- Complex gates can be realized at the transistor level.
- Usually the number of inputs is limited to 4 (the number of transistors in series between the ground and supply is limited)
- For example:
  - OAI21
    - $Y = \overline{(A+B)C}$
  - AOI22
    - $Y = \overline{AB + CD}$
- Usually n-input AND/OR combinations can be realized by *2n* transistors.
- Because there is only one stage, the propagation delay is less than in the case of multi-stage realization

Example

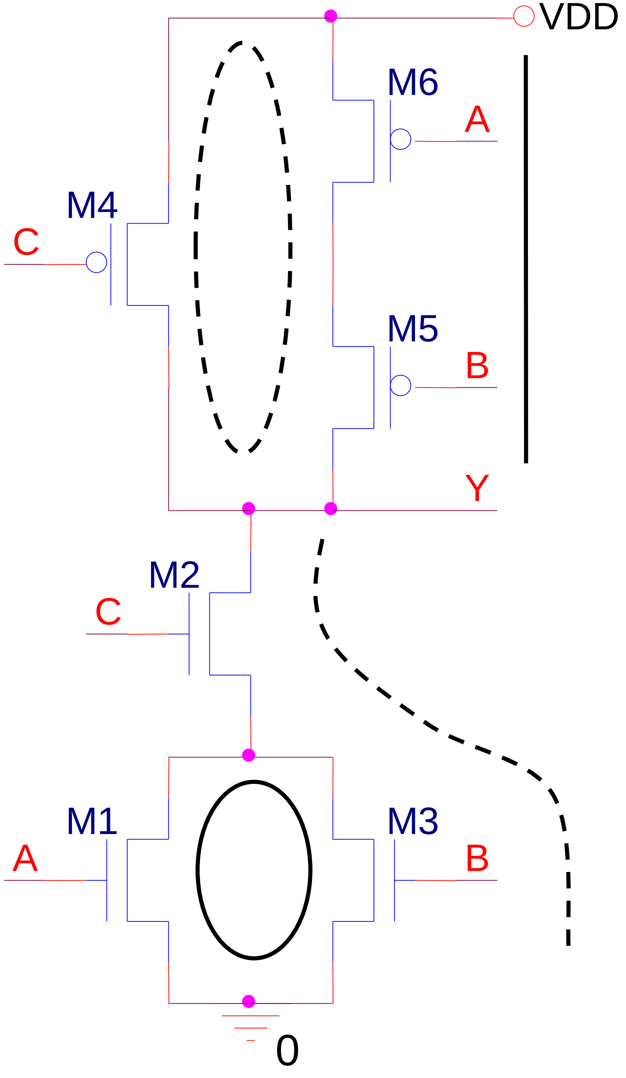
- Design the complex gate realizing the logic function $Y = \overline{(A+B)C}$
- First the pull-down network (PDN) is created
  - The OR function is realized by two n-type FETs connected in parallel.
  - The AND function is realized by two n-type FETs connected in series.
  - The schematic so far:





Example, cont.

- Next the pull-up network (PUN) is designed with p-type transistors
  - The PUN has to create a current path between the supply rail and the output for every logic 1 of the logic function.
  - This can be done by creating the dual network of the PDN.
  - In the dual network every series connection is turned into a parallel connection and vice versa.

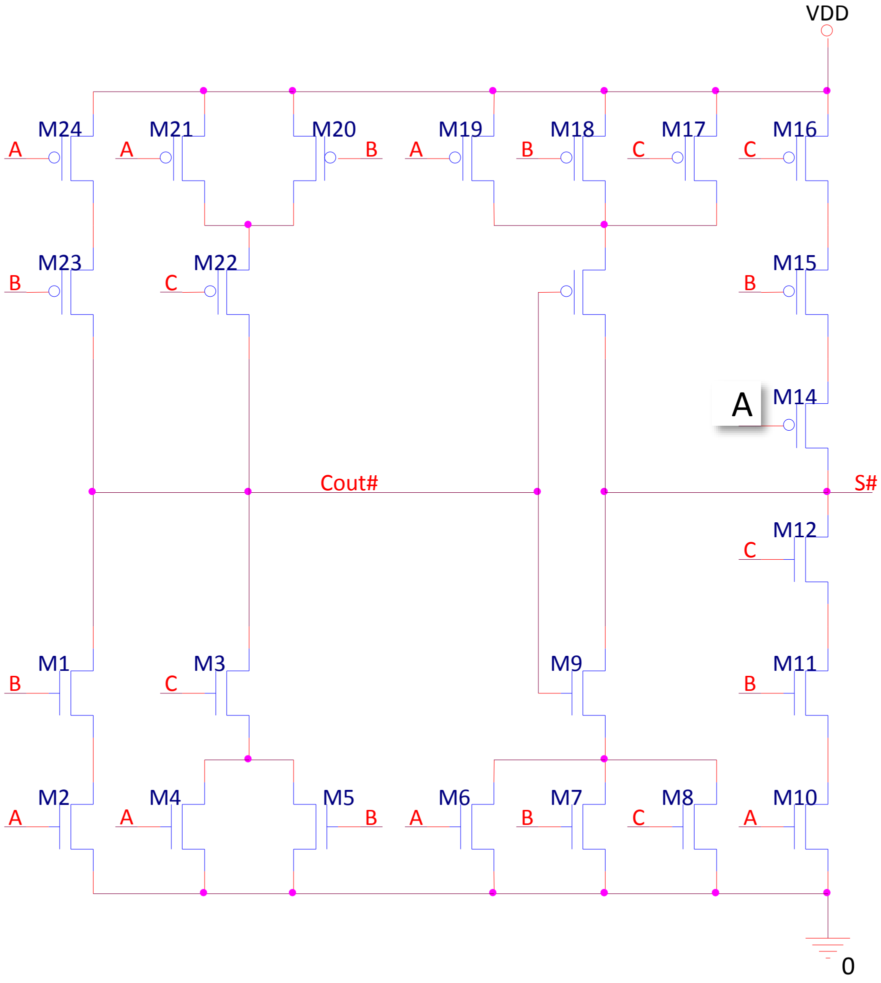


Cont.

- As the p-type transistors conduct when the input is logic 0, the function has to be inverted using the De-Morgan laws.
- In this case: $Y = \overline{C(A+B)} = \bar{C} + \overline{A+B} = \bar{C} + \bar{A}\,\bar{B}$
- As shown above, the two methods yield the same results.

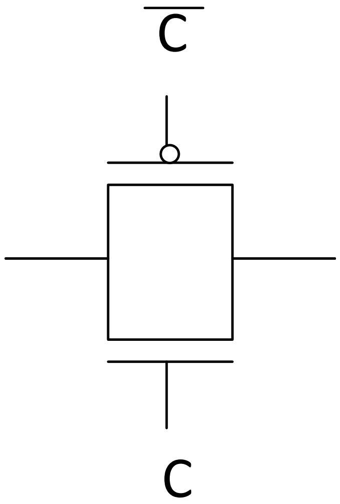
Full adder

- Basis of all arithmetic
  - Adder, counter, comparator.
  - Carry generation results in a critical path (in case of multiple-bit numbers)
- We can use the carry to calculate the sum:
  - $C_{OUT} = AB + C(A + B)$
  - $S = ABC + (A + B + C)\overline{C_{OUT}}$
  - 24 transistors + 4 transistors for two inverters.
- The topology is fully symmetrical
- The carry is generated before the sum



CMOS transmission gate

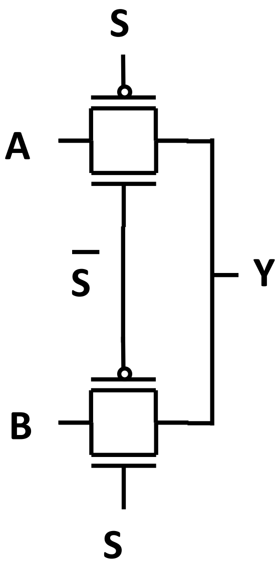
- A transmission gate is an electronically controlled switch in a signal path.
- It consists an nMOS and a pMOS transistor
- The control signal of the nMOS is fed to the pMOS through an inverter.
- Logic gates can be simplified using transmission gates.



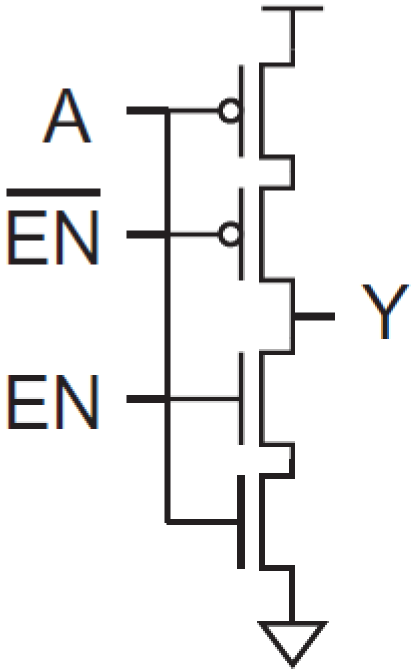
C=0 -> open circuit
C=1 -> short circuit

Application of transmission gates in logic gate design

- Some gates can be designed with far fewer transistors using transmission gates.
- Selective functions are typical examples.
  - In programmable logic circuits the selection is done in this way (e.g. in FPGAs)
- This schematic depicts a simple 2-input multiplexer circuit
  - $Y = A\bar{S} + BS$
- This multiplexer would require 8 transistors in traditional CMOS design. With transmission gates only 4 FETs are needed.



Clocked CMOS (C^2MOS)



- Tri-state gate
- EN=0 – the output floats (high-impedance)
- EN=1 – the input is inverted

CMOS latches and flipflops

Latches

Flip-flops

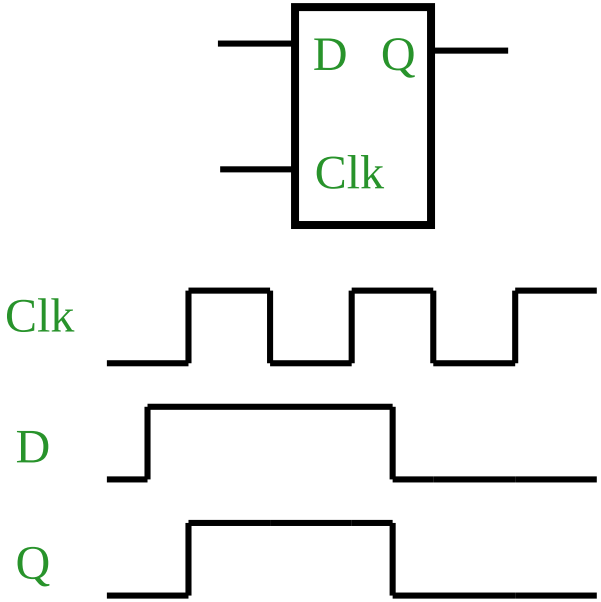
Registers

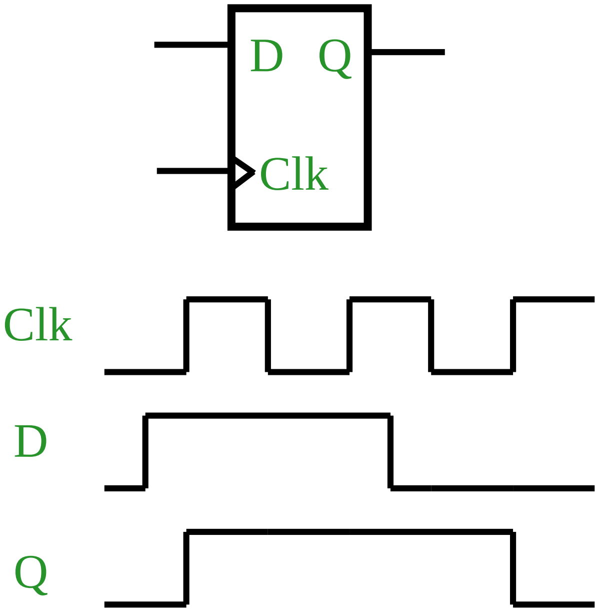
Elements with two states

- Monoflops:
  - have a stable and an unstable states
  - when set to their unstable state, they return to the stable state after a certain time
  - create impulses of a given width
- Bistable flops:
  - latches: have two transparent states – a change at the input is instantly seen at the output
  - flipflops: the output only changes at the rising or falling edge of the clock

Latches and flipflops

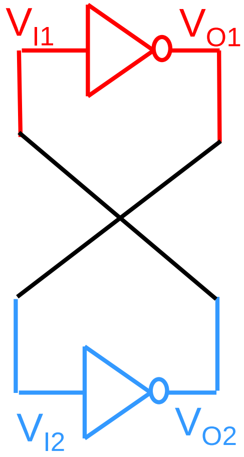
- Latches are transparent while clock level = 1
- Flipflops change their outputs only at the edges of the clock

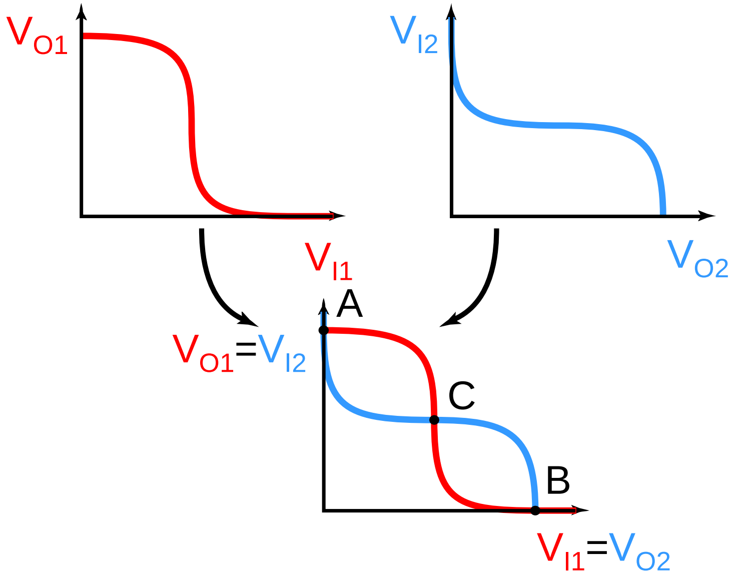




Fundamentals of data storage

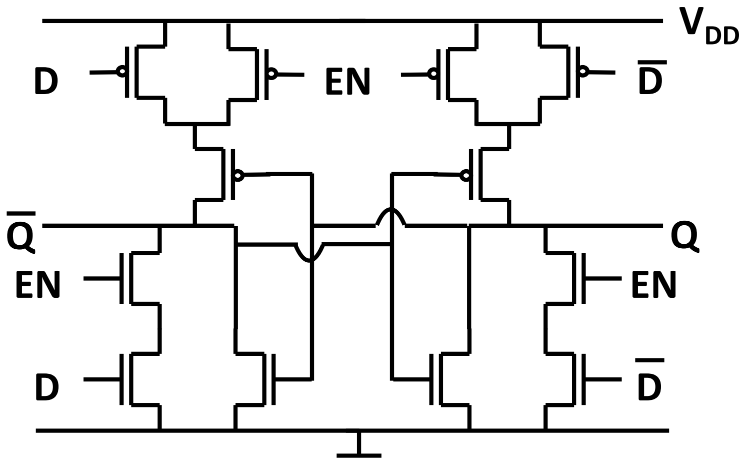
- Bistable circuits: cross connected inverters
  - This circuit has two stable operating points (A, B)
    - It also has a third, metastable OP (C): every voltage is V_C





Topology of storage elements

- It is possible to form storage resources from logic gates (however, it is not favorable from the viewpoint of number of transistors)
  - E.g.: SR latch may be created using 4-input NAND gates
- D-lathes can be made of two AOI gates (and-or-invert gate)

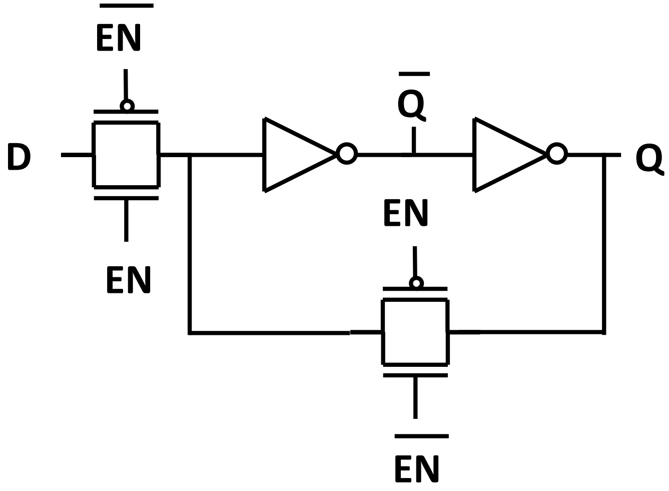


$$Q = \overline{EN \cdot \bar{D} + \bar{Q}}$$

$$\bar{Q} = \overline{EN \cdot D + Q}$$

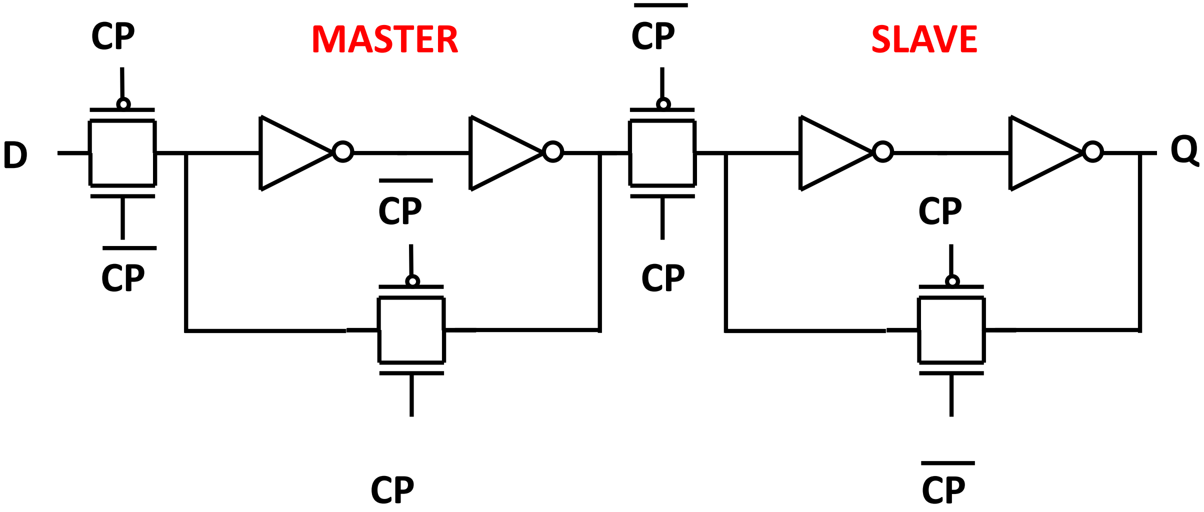
D-latch with transmission gates (TG)

- It consists of two transmission gates and two inverters
  - The transmission gates are controlled by complementary signals.
  - EN = 1: the first TG is transparent (short-circuit), the feed-back TG is opaque (open-circuit) → Q = D
  - EN = 0: the feed-back TG copies the output to the input, the TG on the input is opaque.
  - Only 8 transistors are required.



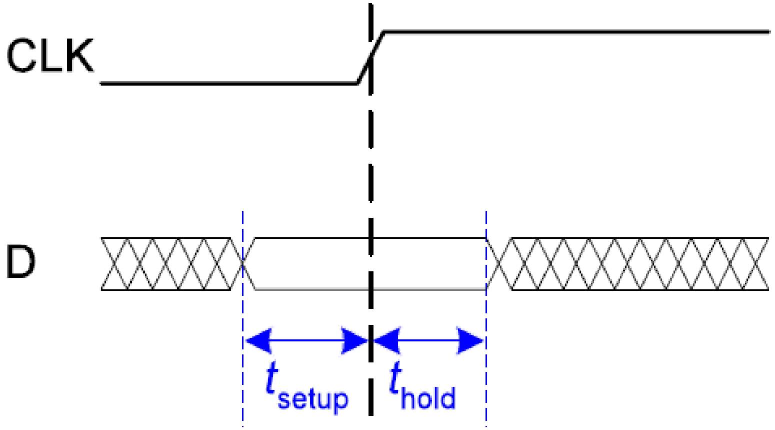
D-flipflop

- Master-slave flipflops comprise two latches connected in series
  - At CP = 0, the first latch is transparent.
  - At CP = 1, the state of the first latch is written into the second one, while the input D is disconnected.



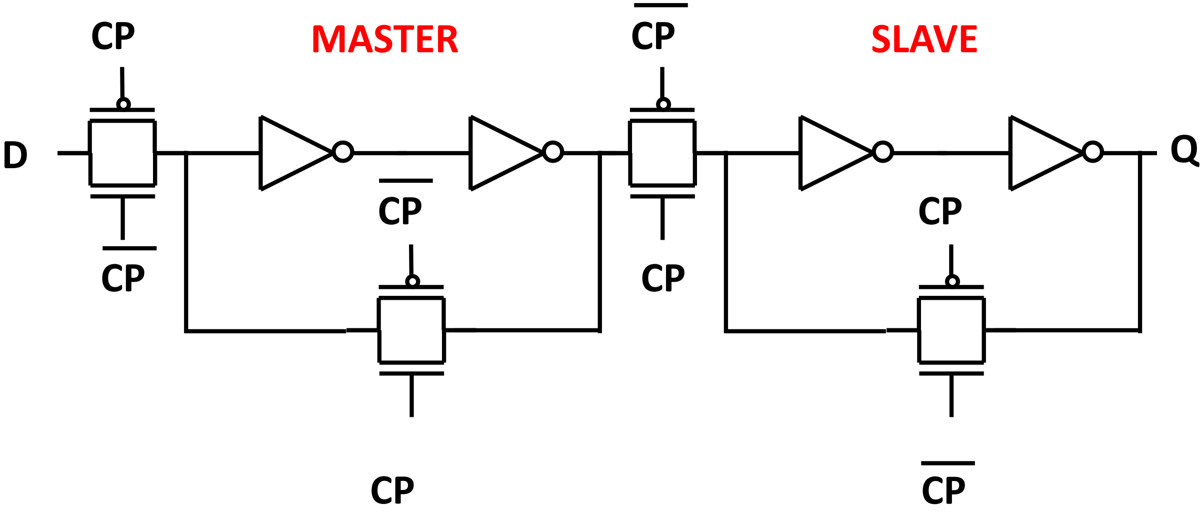
Timing of flipflops

- Setup time: the input (D) has to be stable before the rising edge of clock signal.
- Hold time: the input (D) shall be unchanged after the rising edge of clock signal.



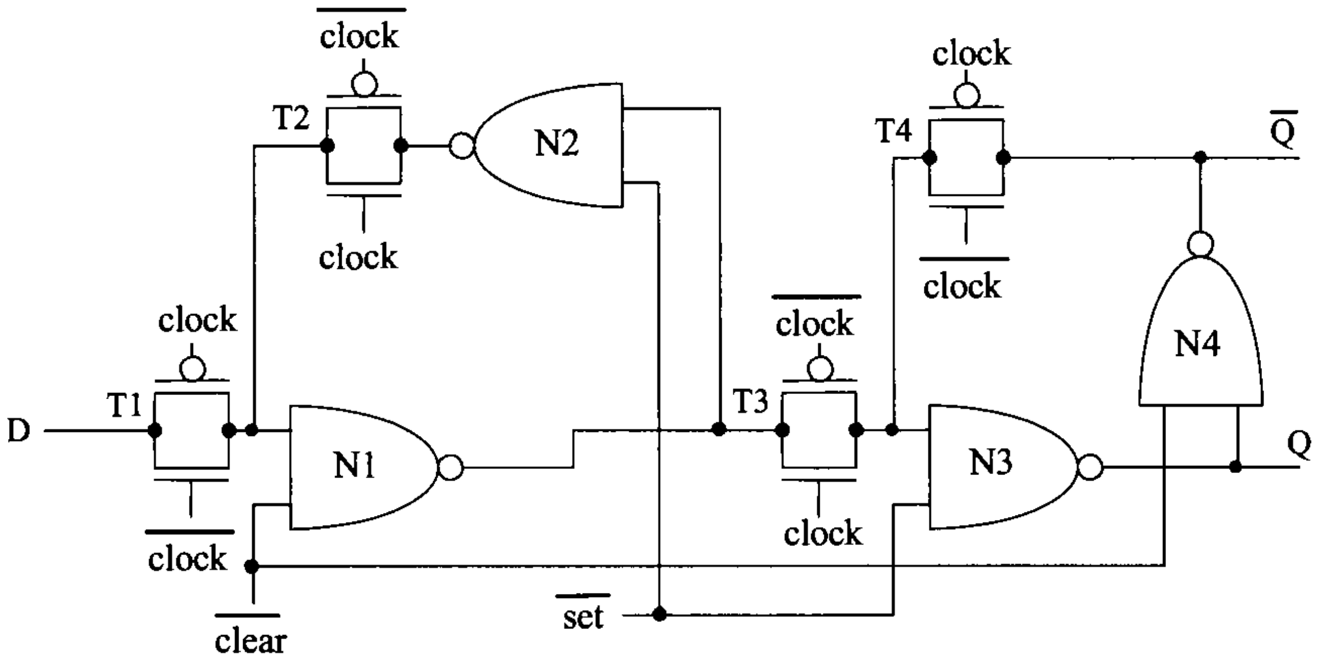
Timing of flipflops

- $t_{SETUP} > t_{PDTG} + 2t_{PDINV}$
  - Sampling: The signal goes through a transmission gate and two inverters.
  - the input (D) has to be stable before the rising edge of the clock signal.
- $t_{HOLD} > t_{PDTG}$
  - The input shall be stable while the first TG is switching



D flip-flop with asynchronous clear and set

- We can use two-input gates instead of inverters (while keeping the inverting behavior)
- Example:


$\overline{\text{clock}}$
T2
N2
clock
clock
T4
$\overline{Q}$
$\overline{\text{clock}}$
clock
T1
D
N1
N4
$\overline{\text{clock}}$
T3
N3
Q
clock
clock
$\overline{\text{clear}}$
$\overline{\text{set}}$

Budapest University of Technology and Economics

Department of Electron Devices

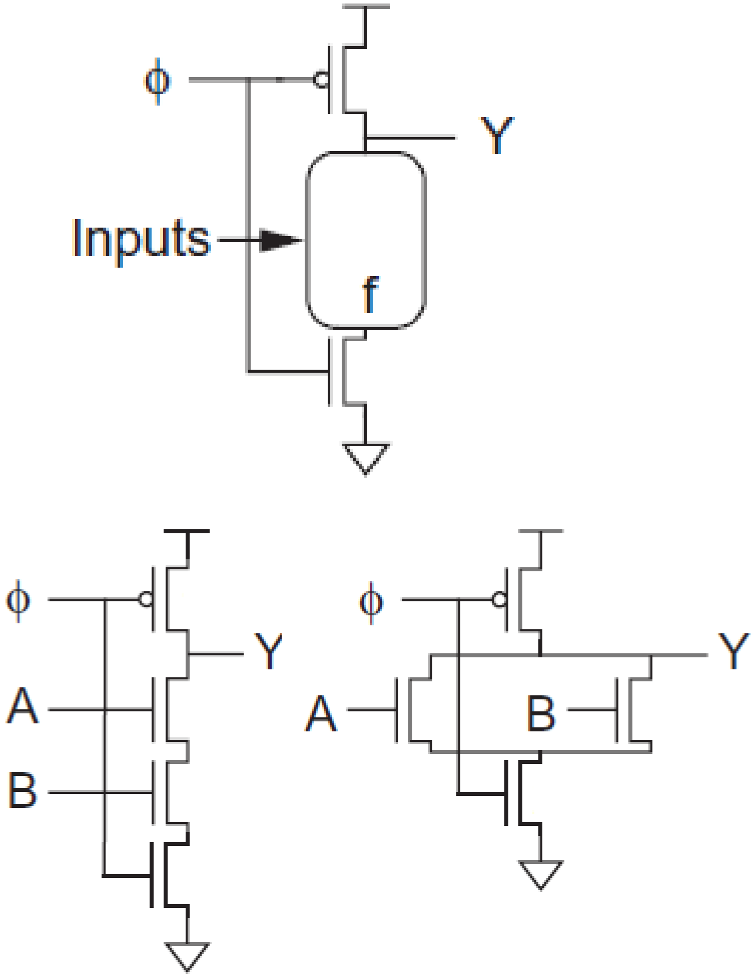
High-speed CMOS logic

Circuit-level techniques for decreasing the propagation delays

High-speed CMOS logic

- Propagation delay of static CMOS logic
  - $t_{pd} \sim \frac{CV}{I}$
- Technology-level techniques
  - Decreasing load capacitances
  - Increasing currents (while keeping or decreasing the sizes)
- Circuit-level techniques
  - Decreasing logic swing: differential signaling
    - The logic value is represented by the sign of a voltage difference
    - SCL: Source-Coupled Logic
  - The parasitic capacitances are exploited (intentionally for temporarily storing logic values)
    - Area saving → increasing speed
    - CMOS domino logic, dynamic storage elements

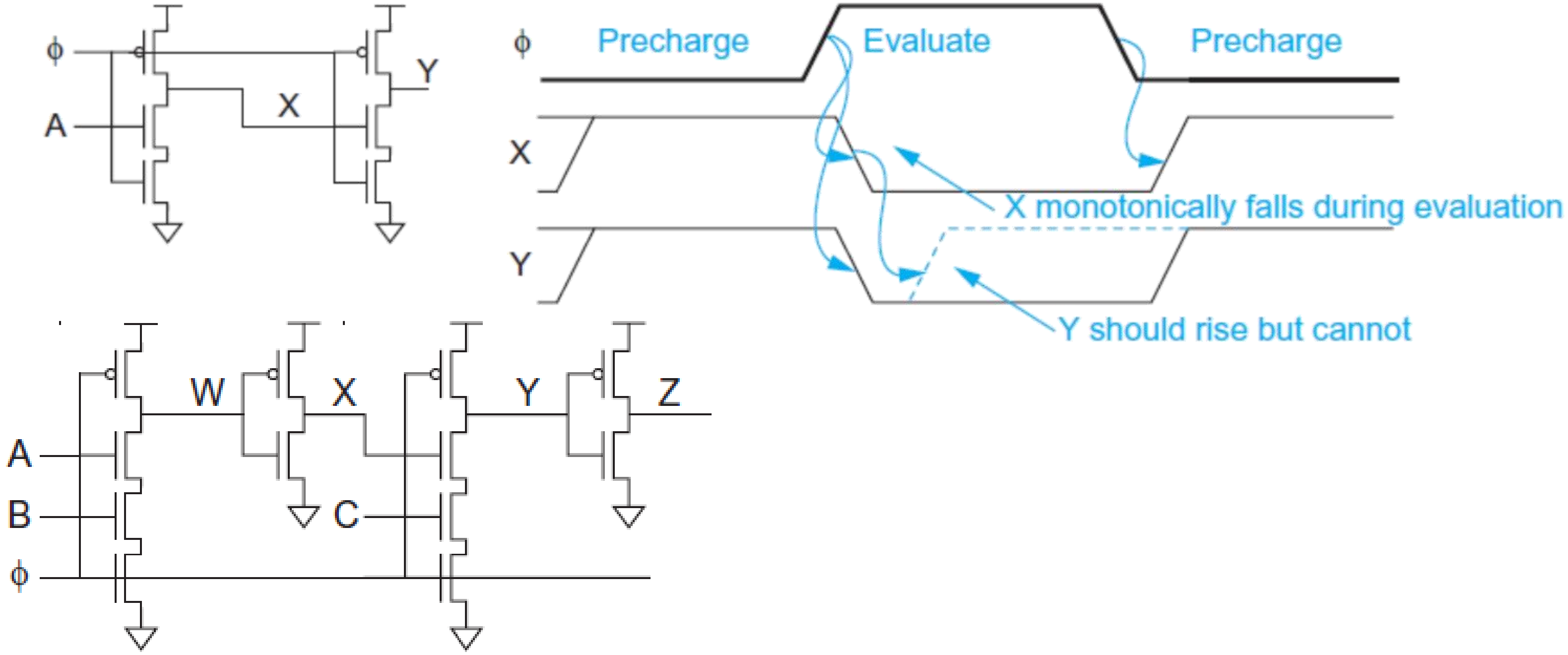
CMOS domino logic



- Parasitic output capacitances are precharged
- Φ=0: precharge to V_{DD}.
  - pMOS conducts, nMOS is open-circuit, output: logic 1
- Φ=1: evaluation
  - According to the logic inputs, the pull-down network discharges the output capacitance (or not)
  - Pros:
    - N+2 transistors needed for N-input logic gates
    - It is faster, because of the reduced load capacitances.

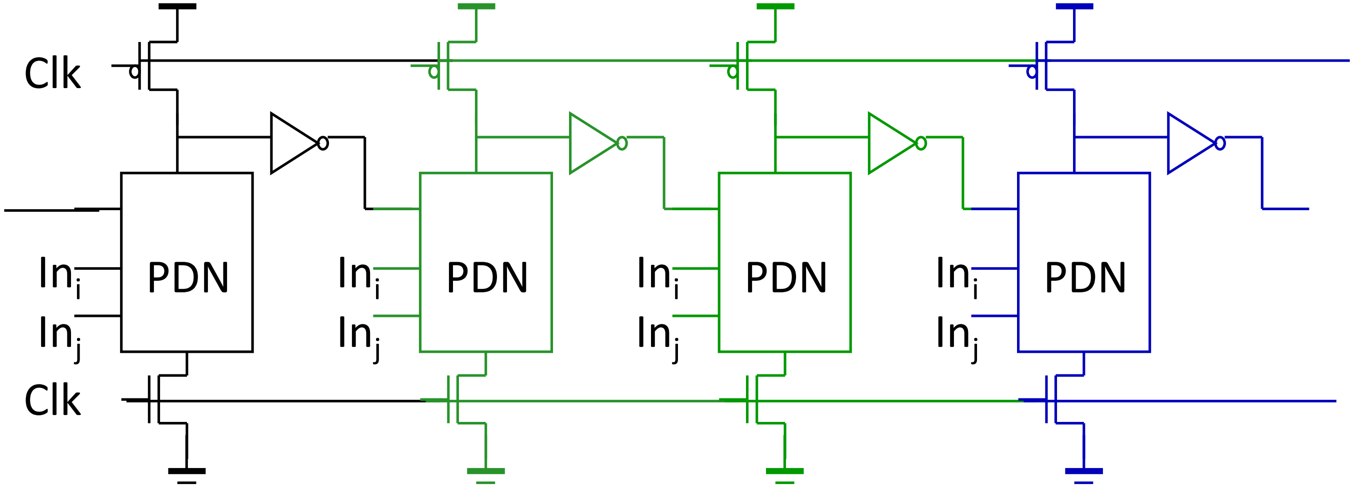
CMOS domino logic

- Direct cascading is problematic
  - At the beginning of the evaluation, if a logic 0 is needed at the output of the nth stage, a false logic 1 appears on the input of the (n+1)th stage.
  - The false logic 1 may start to discharge the output of stage n+1, the logic value decreases unintentionally.
  - The stages are cascaded through static inverters.



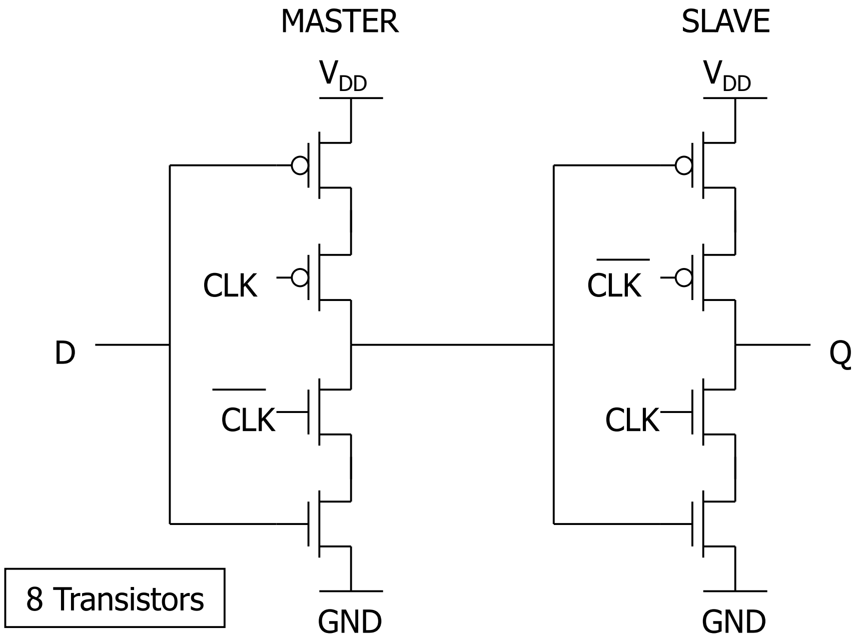
CMOS domino logic

- It is ~1.5× faster than static CMOS logic



Dynamic D flipflop

- We connect two clocked CMOS inverters in series, which is controlled by complementary signals
- The information is stored in the parasitic capacitances between the stages

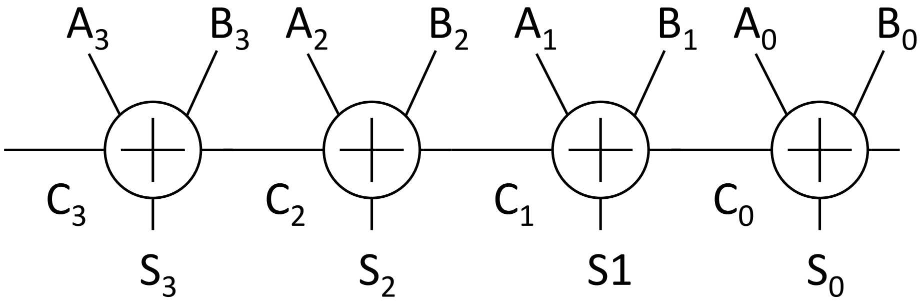


Elements of datapaths

- Ripple carry adder
- Combinational multiplier

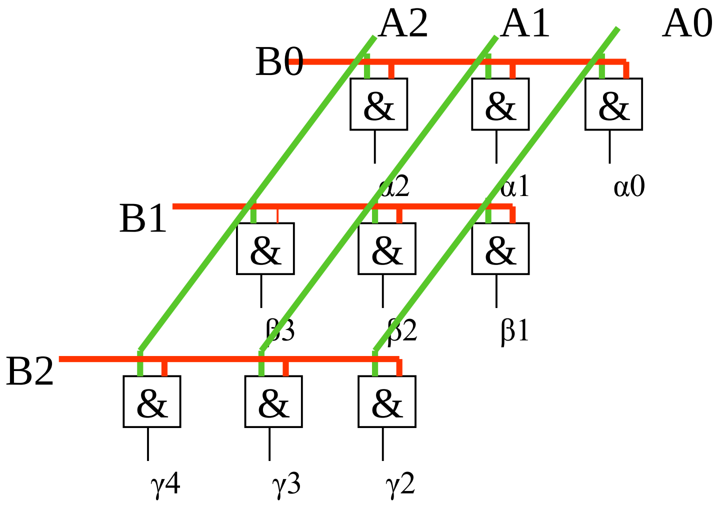
Ripple-carry adder

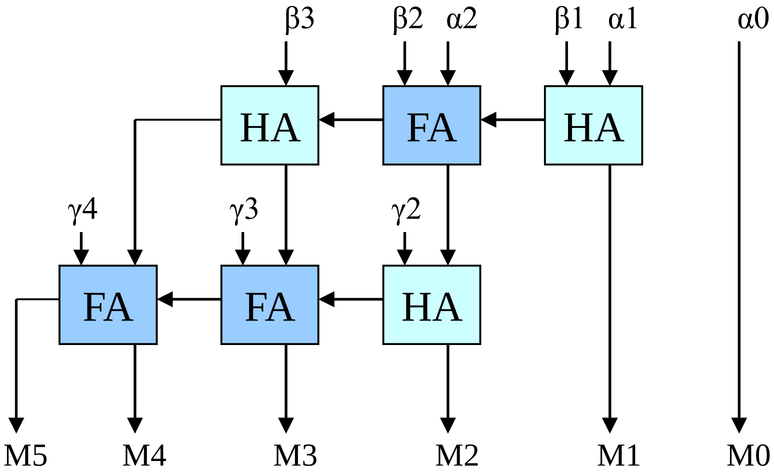
- Multiple 1-bit adders are cascaded to form multi-bit adders
- Required computation time: $n \cdot t_{pdcarry}$
- The carry propagation line forms the critical path.
  - It is implemented using alternating CMOS logic (inverters are not needed)
  - The delay of the critical path significantly rises as the number of bits increases



The structure of a combinational multiplier

- Generates the sum of the partial products with respect to the decimal offset.


A2
A1
A0
B0
&
&
&
α2
α1
α0
B1
&
&
&
β3
β2
β1
B2
&
&
&
γ4
γ3
γ2


β3
β2
α2
β1
α1
α0
HA
FA
HA
γ4
γ3
γ2
FA
FA
HA
M5
M4
M3
M2
M1
M0