Gates in CMOS-Technology

6. Standard/complex CMOS static logics gates

7. Special gates and the applications
    - 7.1 Transmission gates
    - 7.2 Clocked CMOS (Tri-state CMOS gates)
    - 7.3 Bus-driver
    - 7.4 Multiplexer

8. CMOS-circuit techniques (logic gates)
    - 8.1 Pseudo-NMOS
    - 8.2 Cascode-logics

9. Dynamic logics (Precharge-Evaluate)

10. Summary:
    - CMOS-circuit types for combinatorial circuits
        - 10.1 Advantages and disadvantages of static logics
        - 10.2 Advantages and disadvantages of dynamic logics

Dynamic Logics

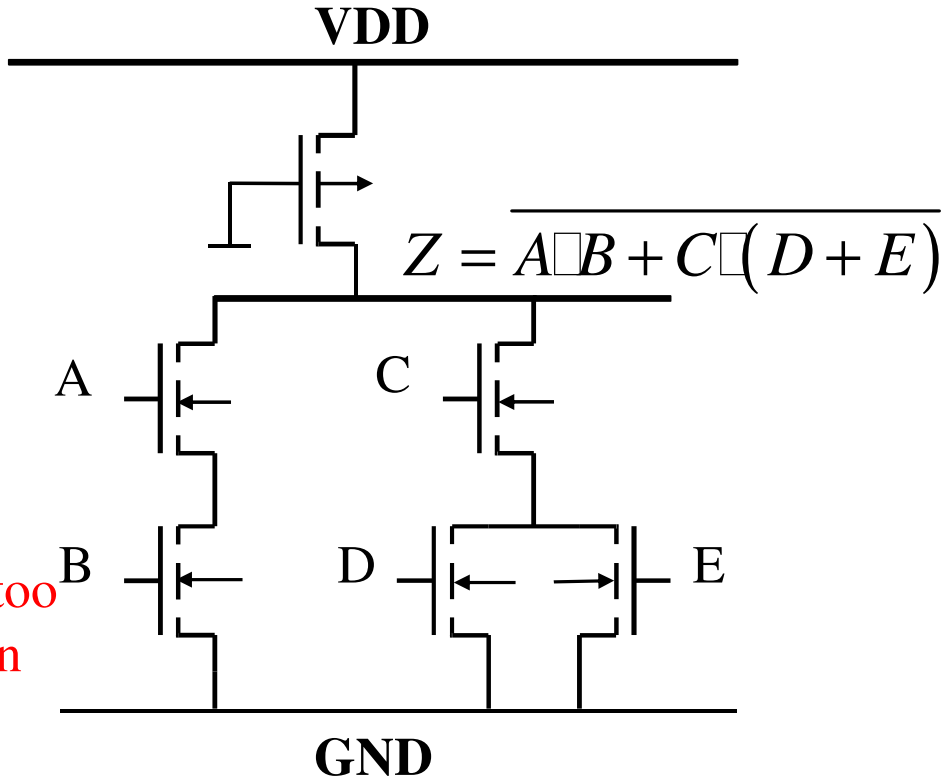
Up to now: Static Logics

⇒ Logic level at the output is changing with changing inputs

⇒ Pseudo NMOS ideal for

chip size, speed &
dynamic power dissipation

but due to switched on p-channel transistor there is too much static power dissipation at low-output

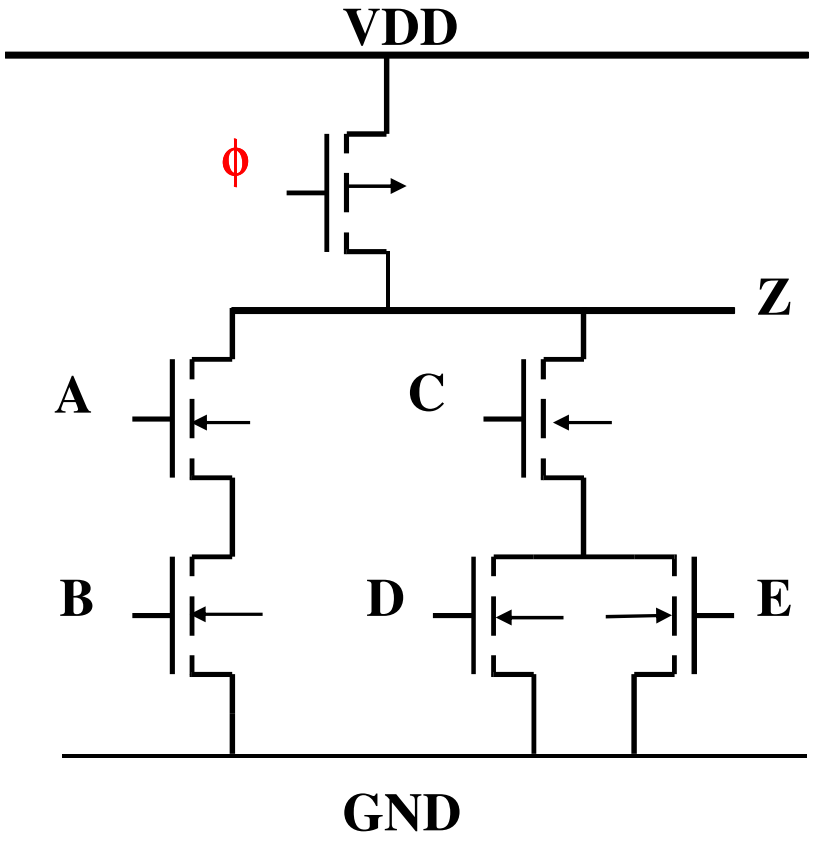


Dynamic Logics

⇒ Switch the p-channel „smartly“ (within one clock) on and off to save the static power dissipation

≠ Register, Flip-Flop
but combinatorial functions

Precharge / Evaluate (P/E) Logics

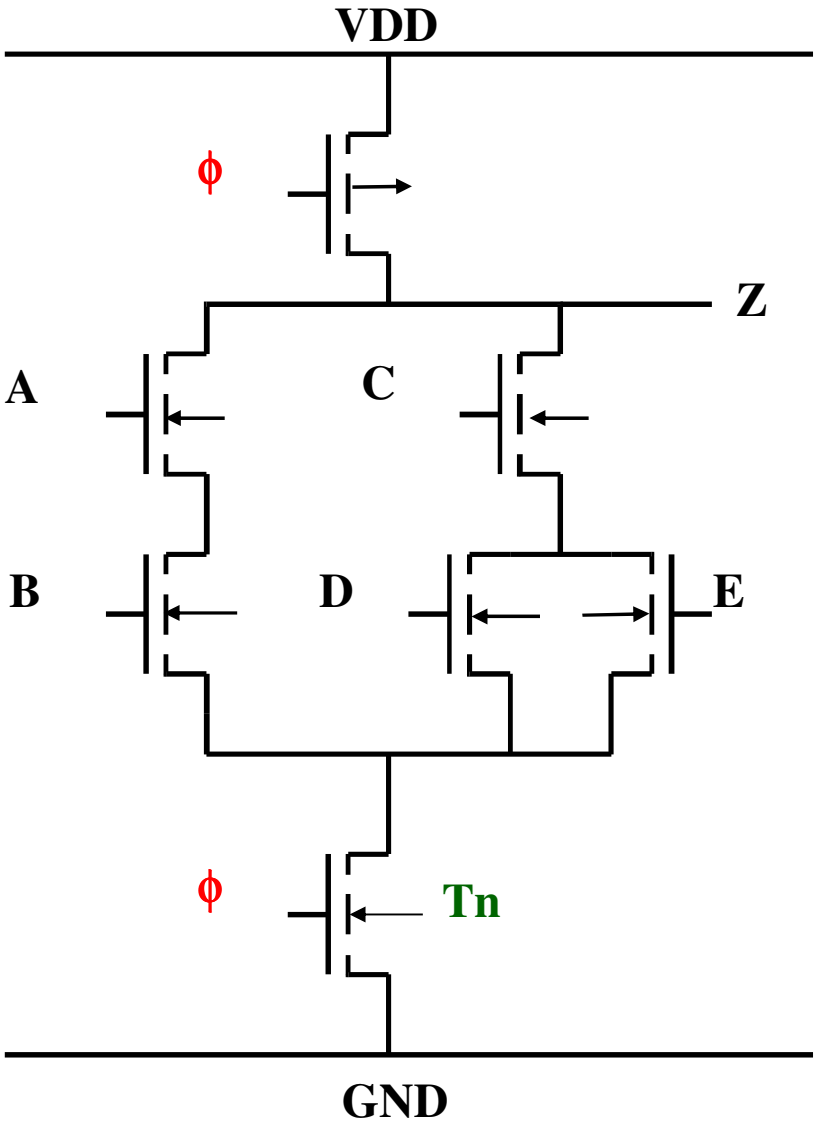


Dynamic Logics Circuit

Clock : ϕ	Precharge	Evaluate

	Z=1
	charged
	
	So that no current flows
	⇒
	Inputs all =0!
	
	⇒ But Outputs all =1!
		Z=f(A,B,...)
		⇒ Inputs can be active
		
		if path to GND exists:
		⇒ Z=0
		
		if not:
		⇒ Z=1
		
		capacitivly stored (charged)
		
		➢Leakage current

Improved Dynamic Logic Circuit



V_{DD}-Path	GND-Path
Z=1 charged	Z=f(A,B,...) evaluated
Tn blocks:	Tn conducts:
Current does not flow ⇒ Inputs are arbitrary!	if path to GND exists ⇒ Z=0 if not: ⇒ Z=1
	capacitivly stored (charged)
	➢Leakage current

Advantages of Dynamic Circuit Technology

1. Low area requirements
   (1 Transistor per input)

2. Low capacitive load
   (1 Transistor per input)
   (n+2 instead 2n Transistors)

3. No static power loss

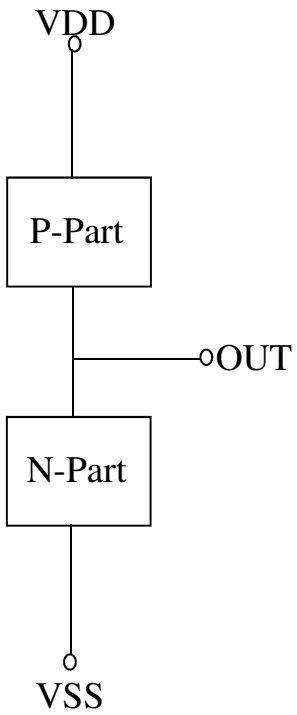
Disadvantages of Dynamic Circuit Technology

1. Clock for each gate

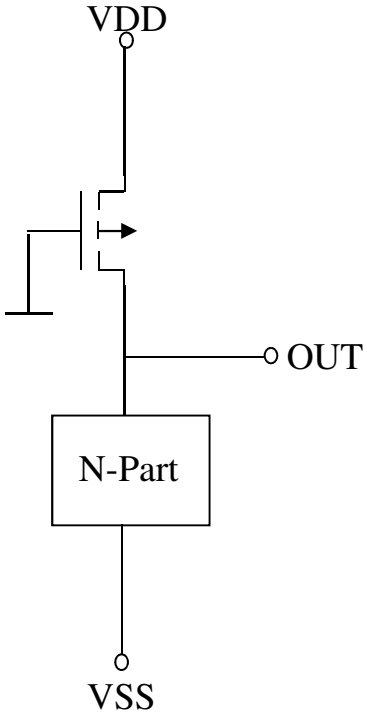
2. High-level only stored capacitivly

Overview of Combinatorial Logics

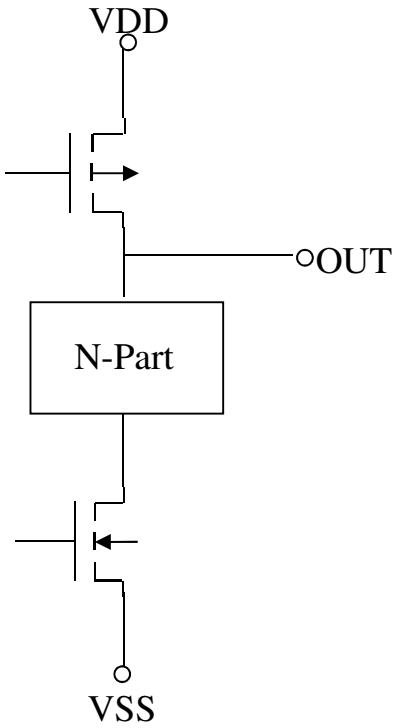
Standard Logic



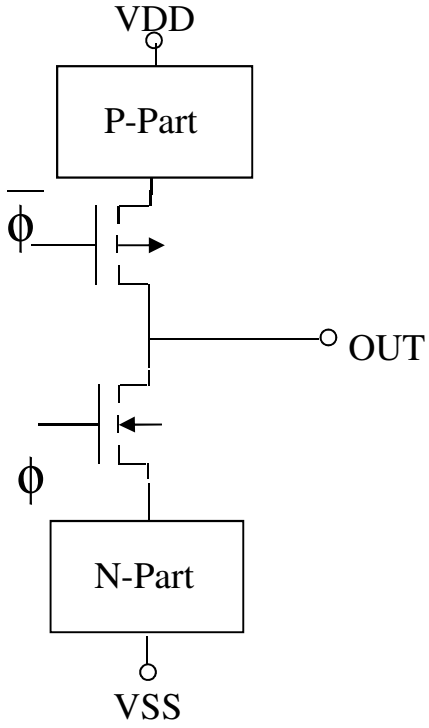
Pseudo NMOS



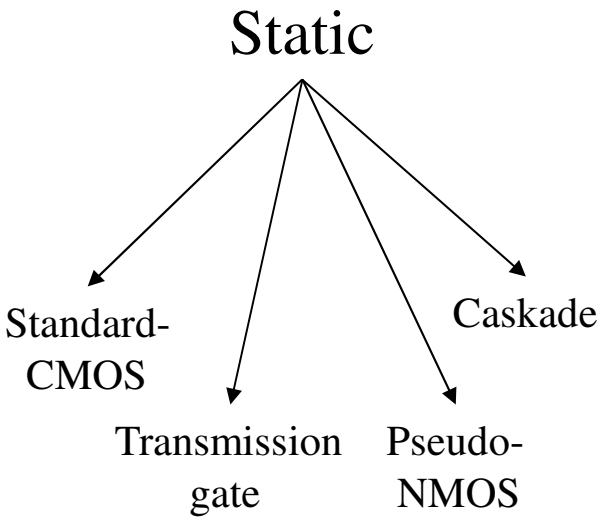
Dynamic Logic precharge/evaluate

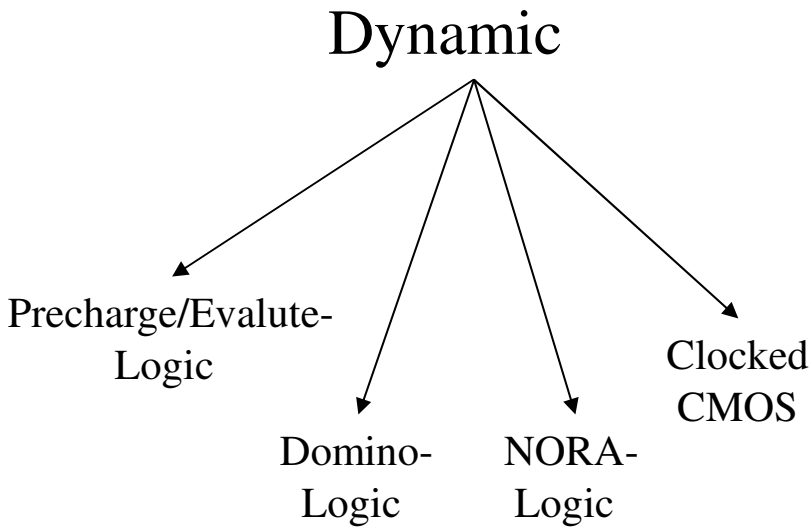


Clocked CMOS Logic



Comparison static/dynamic logic





Static

- 2n Transistors
- Large area
- High load capacity
- **High power loss**
- **Long delay**

+ good noise margin

+ power loss depends on frequency up to and including 0 Hz

Dynamic

- Layout and dimensioning is more difficult

- Clock supply is needed

- A "1" is only stored capacitivly, clock must always be refreshed

+ low load capacity
- **small delay**
- **low power loss**

11. Sequential Circuits

1. **Realization of Registers**
   (Latches, Flip-Flops)
   1. **Dynamic Registers**
   2. **Static Registers**
2. **Pipeline**
3. **Clock Distribution – Clock Networks**
4. **Clock Generation**

Limits of combinatorial Logics

Up to now we only considered the design of pure combinatorial logics.

The outputs of a gate network are only depending on the inputs applied at this time. Effects in time are only considered with regard to signal propagation from input to output (delay). Previous input assignments have no effect on circuit output (no storage; no memory)

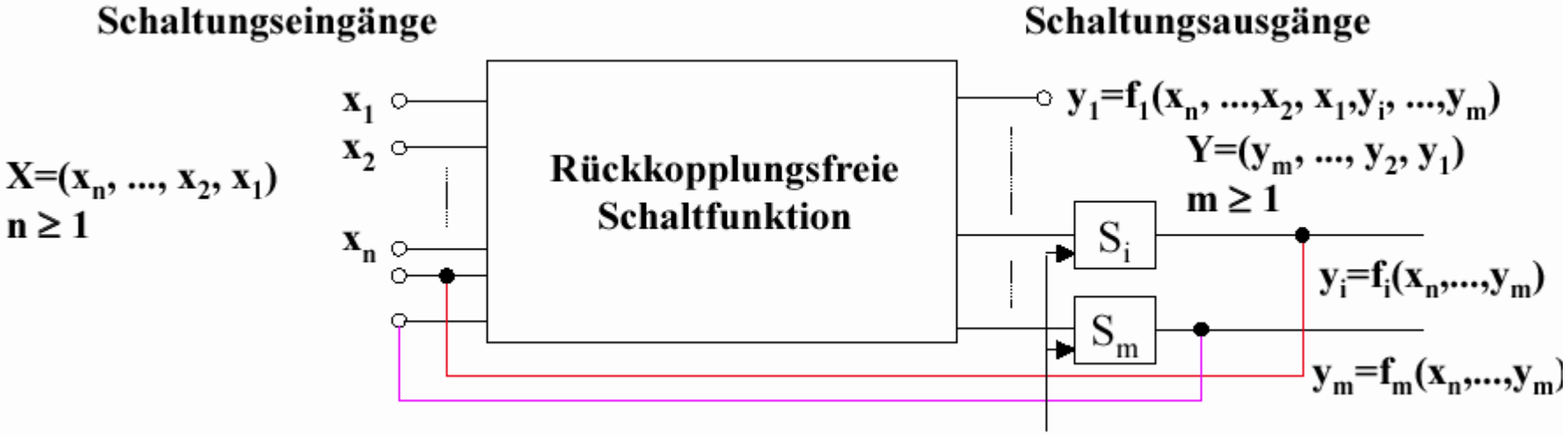
Advantage: Simple structure, easy to understand and to design
Disadvantage: All input information has to be given simultaneously!

Sequential Circuits

A safer and more calculable design of a feedback structure by using of storage elements in the feedback loop as memory

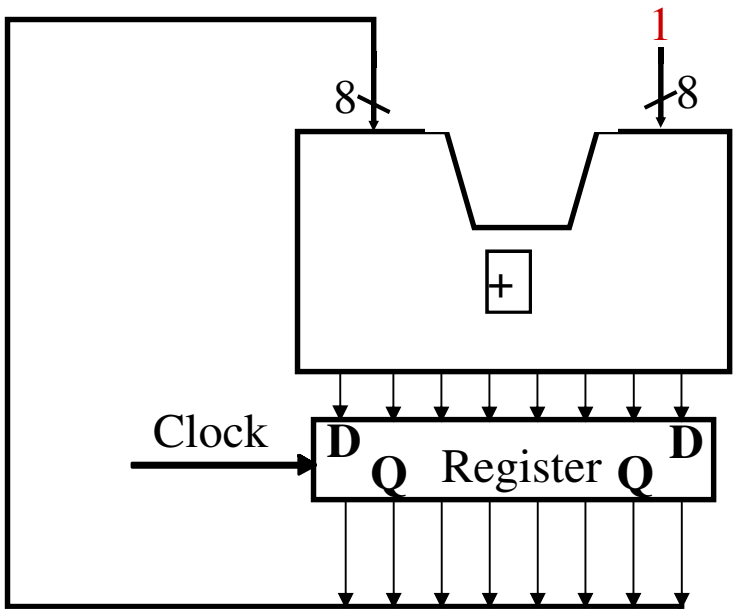
Control of the storage elements by external system clock

Synchronous switching: Synchronous sequential circuit
(temporally decoupled)



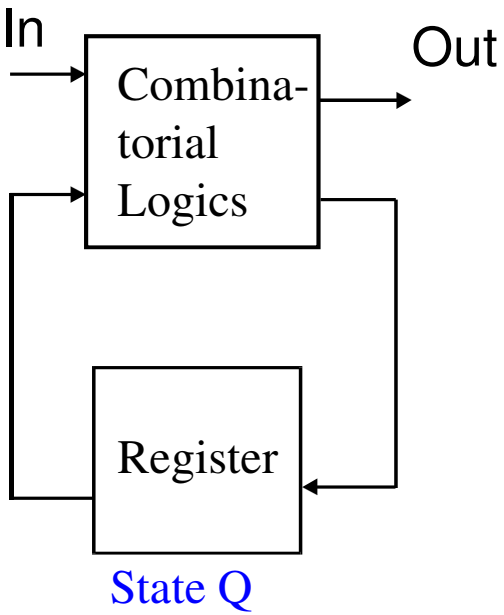
Synchronization prevents Race and Oscillations.

8-bit Adder



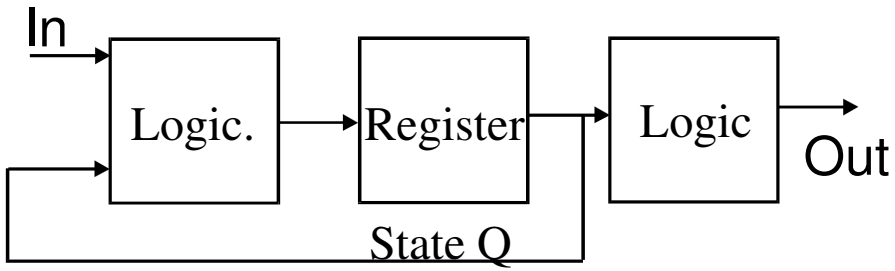
Accumulator / Incrementer

Finite State Machine, FSM



Mealy-Machine

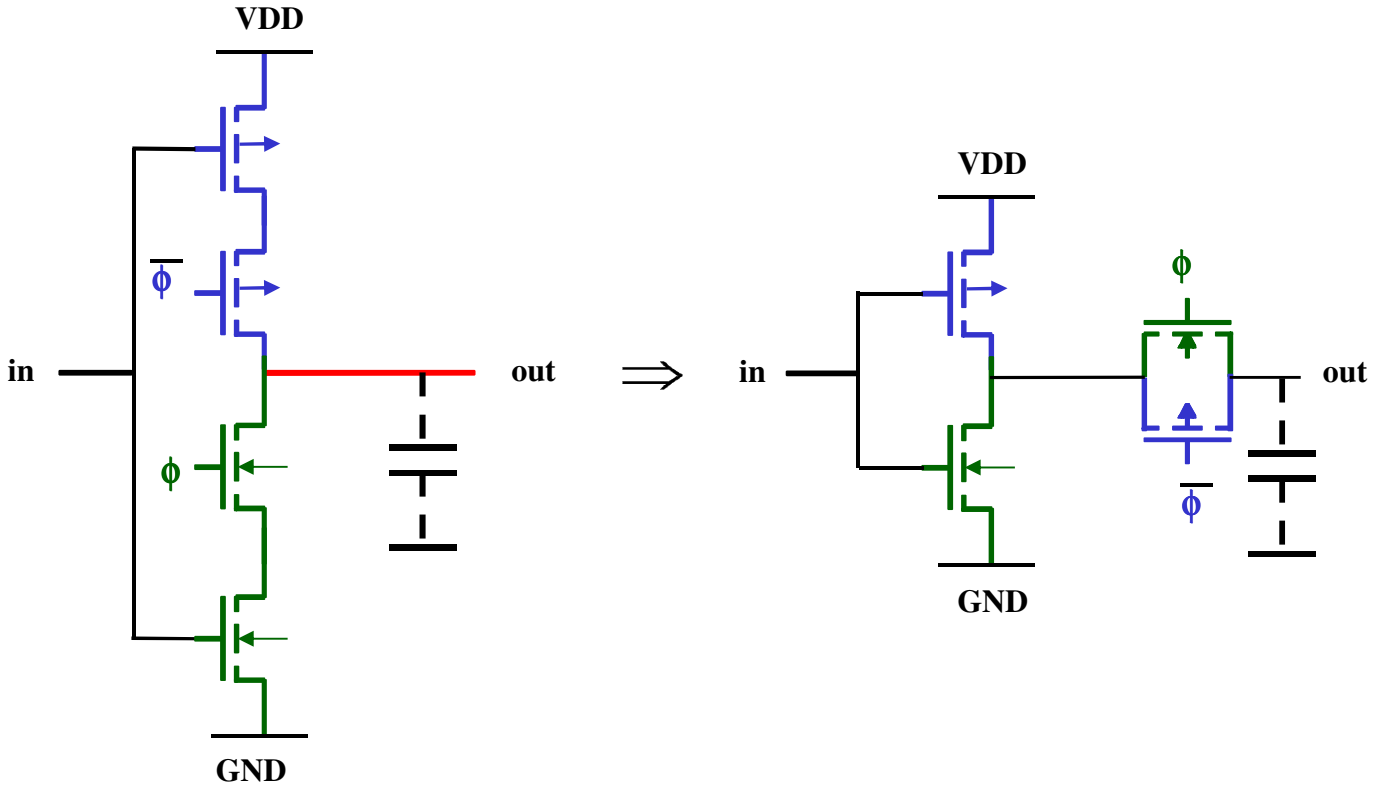
out=f(in,Q)



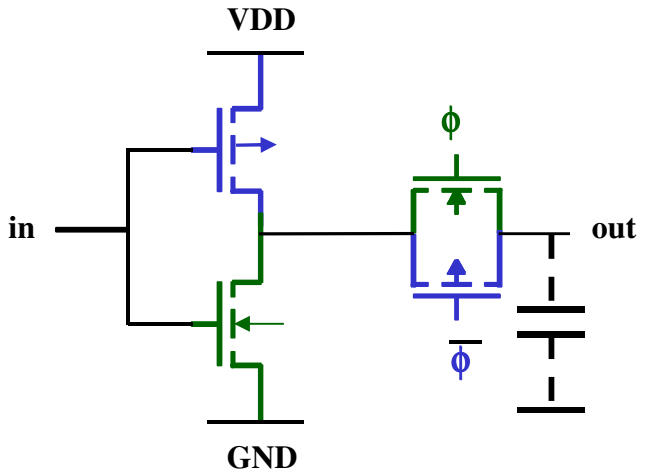
Moore-Machine

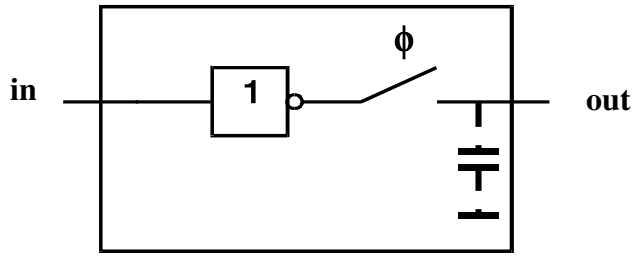
out=f(Q)

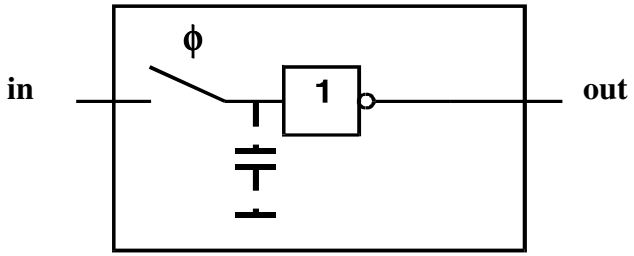
CMOS Inverter for simple Latch



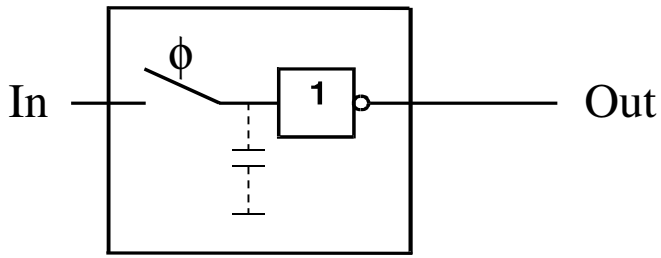
Inverting Latch



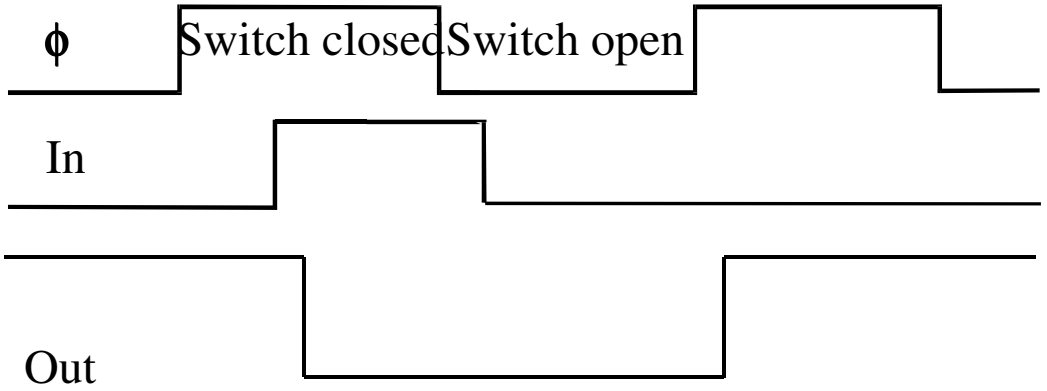




Function of Latch



Dynamic Latch

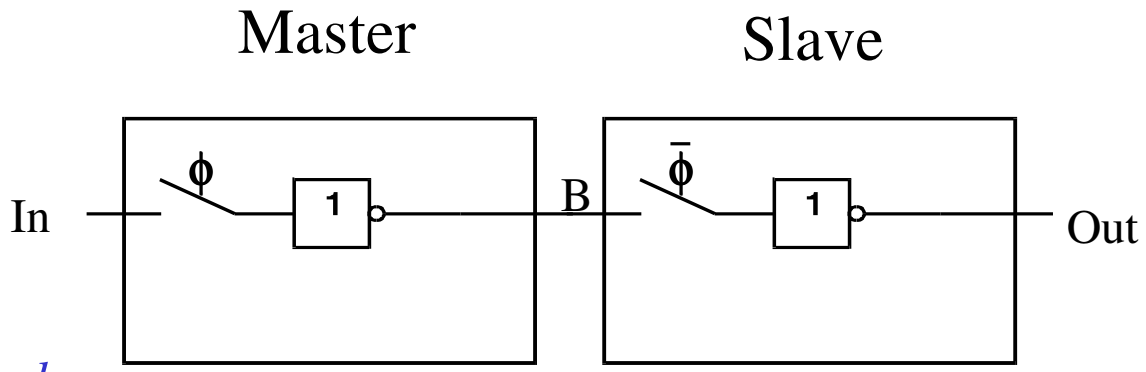


Timing of the Signals

ϕ = 1: transparent (load), In is mapped directly on Out

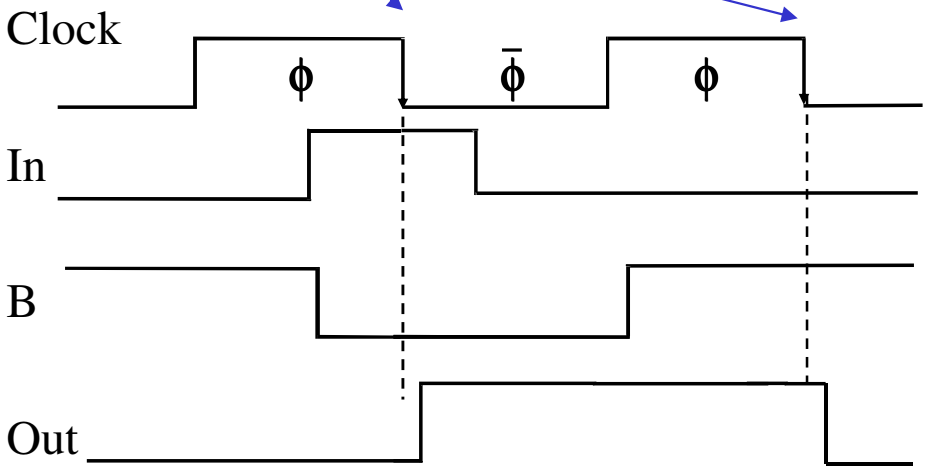
ϕ = 0: hold (storage), last state of Out is frozen

Dynamic Flipflop

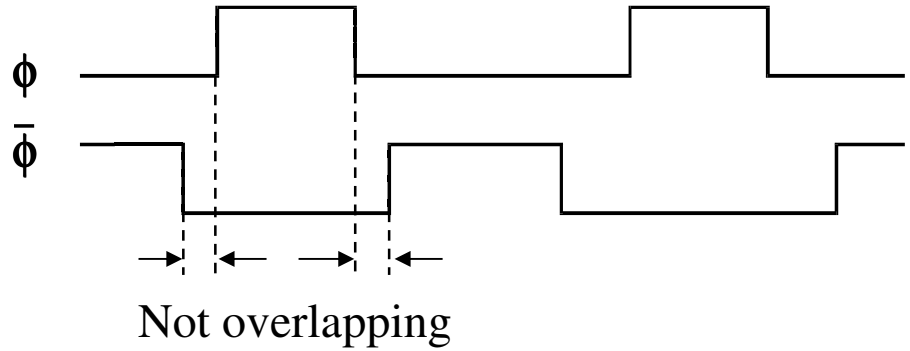


Flipflop with two Latches

edge controlled

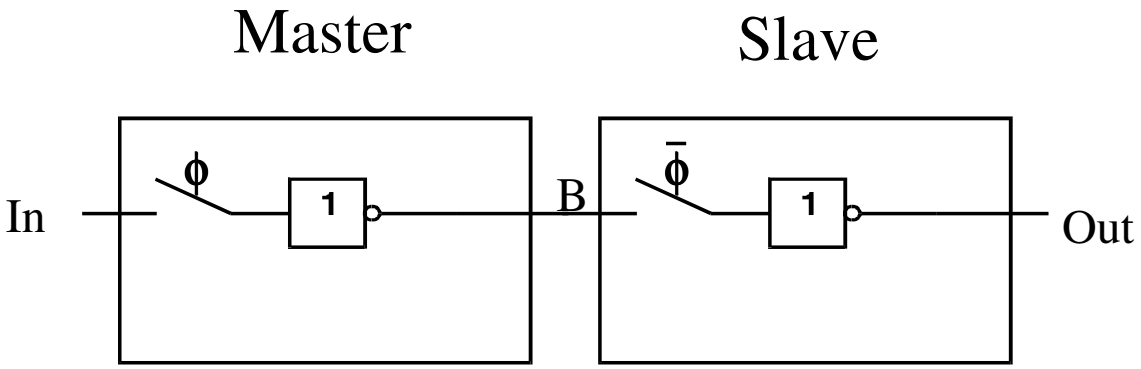


Timing of the Signals

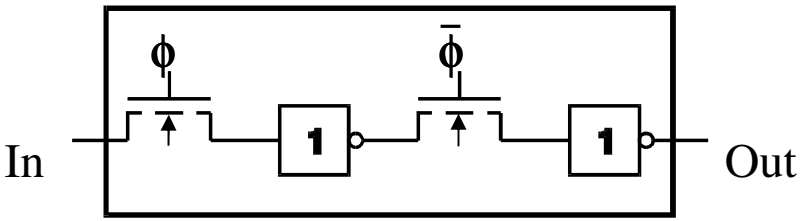


Two-phase clock

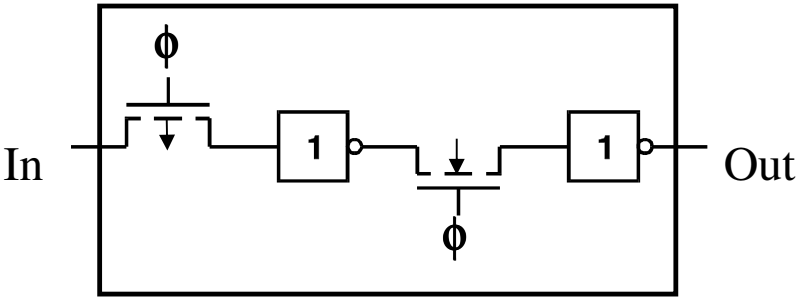
Implementation of dynamic Flipflop



Register with tow Latches

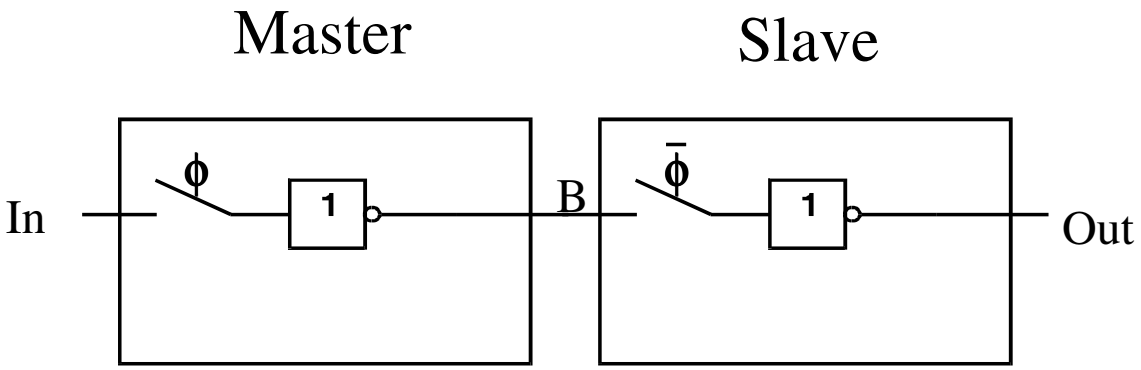


Register with two-phase clock
and N-Transistor as Switch

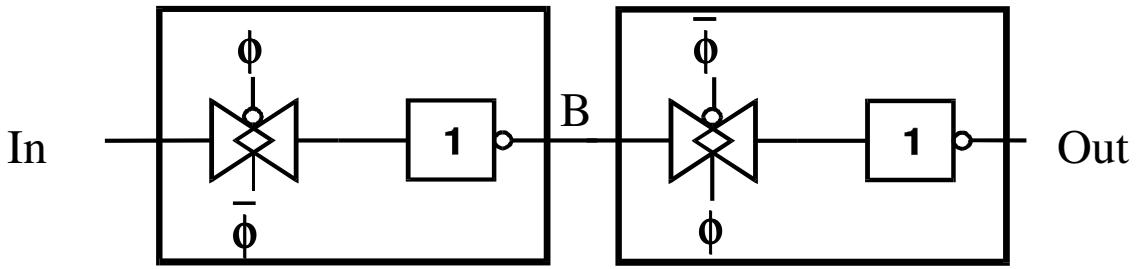


Register with one-phase clock
and N- & P-Transistor as Switch

Transmission-Gate dynamic Flipflop



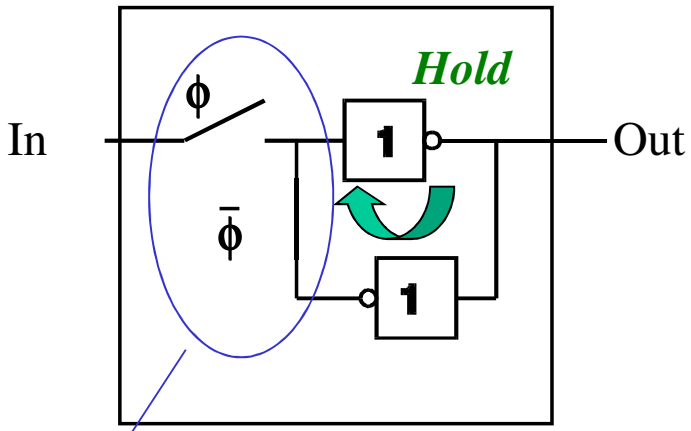
Register with two Latches



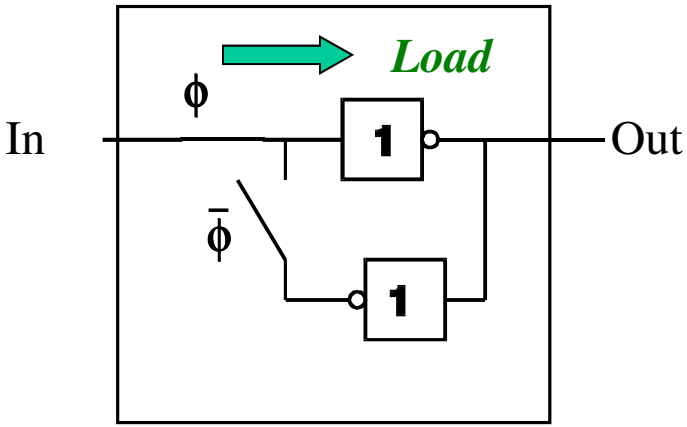
Register with two Latches and Transmission-Gate as Switch

8 Transistors

Static Latch (with Transmission Gate)



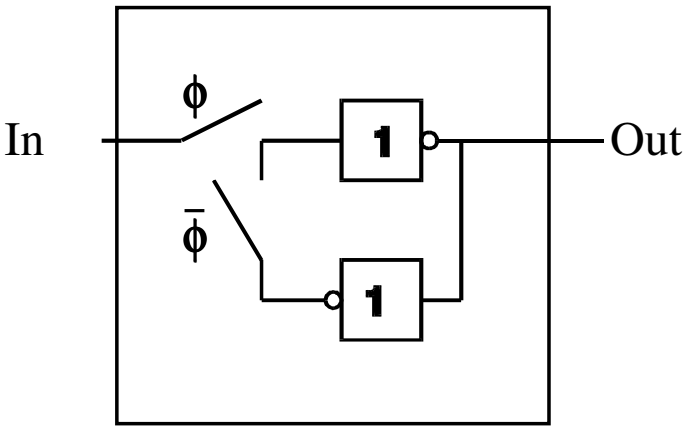
Static Latch
$\Phi = 0$



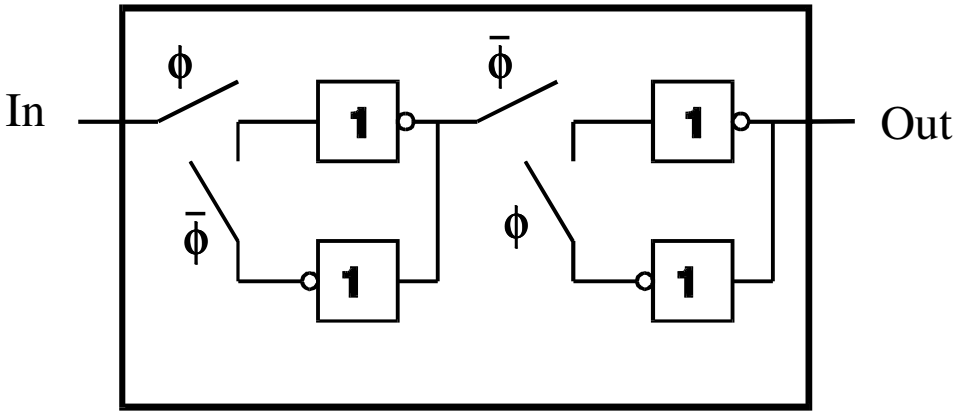
Static Latch
$\Phi = 1$

Multiplexer

Static Master-Slave Flip-Flop (with Transmission Gate)



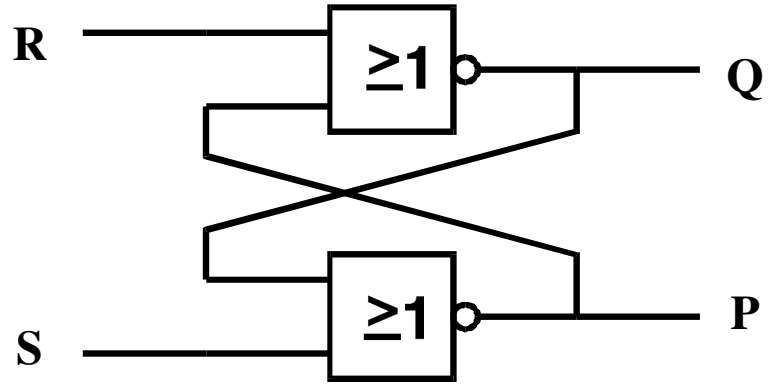
Static Latch

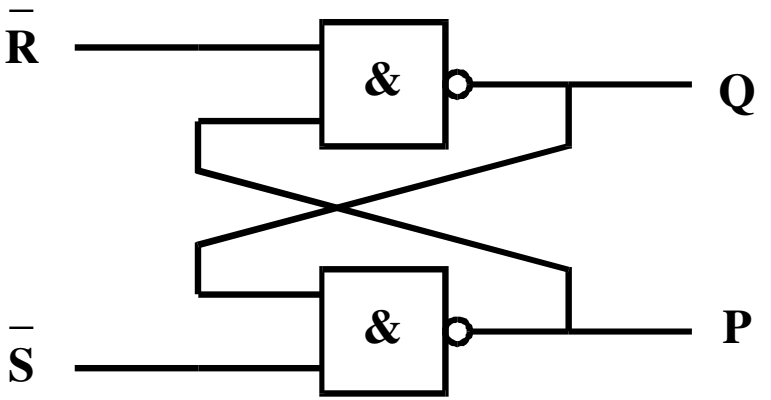


Static Master-Slave Flip-Flop

16 Transistors

RS-Latch





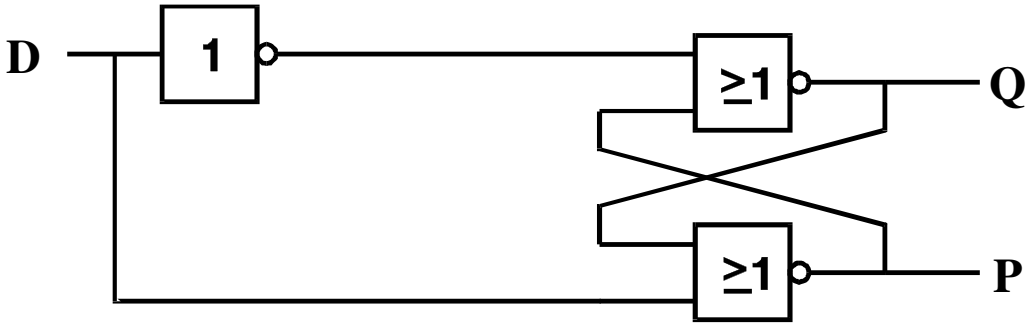
same function

Truth table of RS-Latch

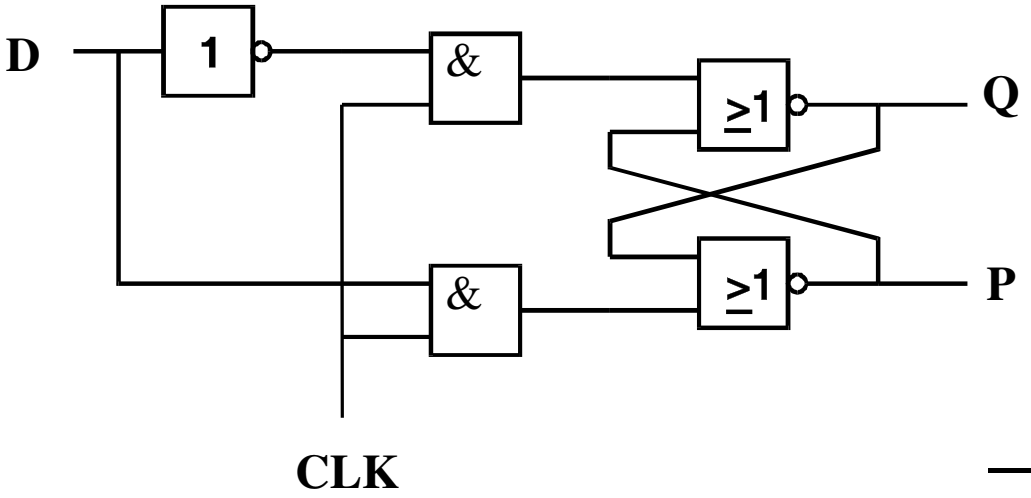
R	S	Q	P
0	1	1	0
1	0	0	1
1	1	0	0
0	0	Q(t-1)	P(t-1)

Storage

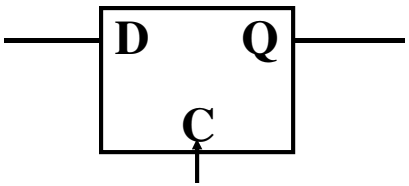
D-Latch



D	Q	P
1	1	0
0	0	1

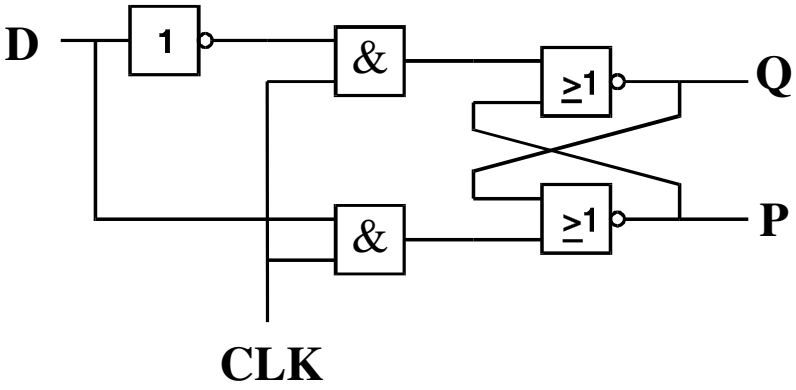


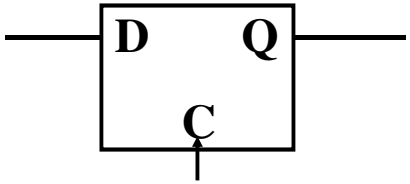
C	D	Q	P
1	1	1	0
1	0	0	1
0	X	Q(t-1)	P(t-1)



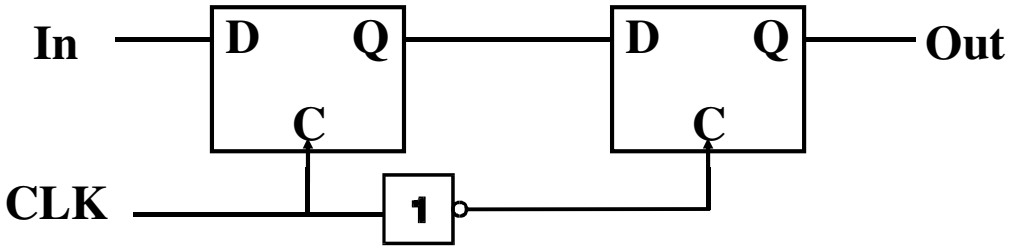
Clocked D-Latch

D-Flip-Flop



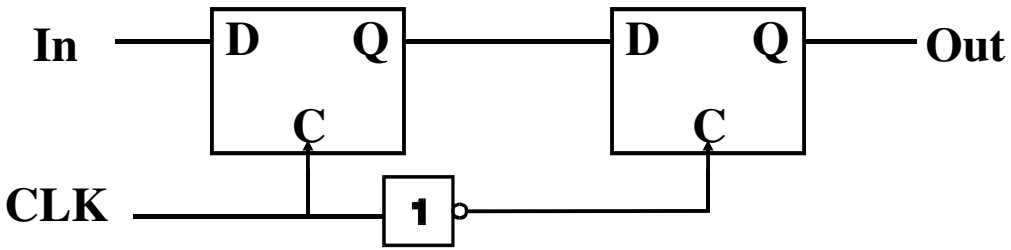


Clocked D-Latch



Master-Slave-Flip-Flop with two clocked D-Latches

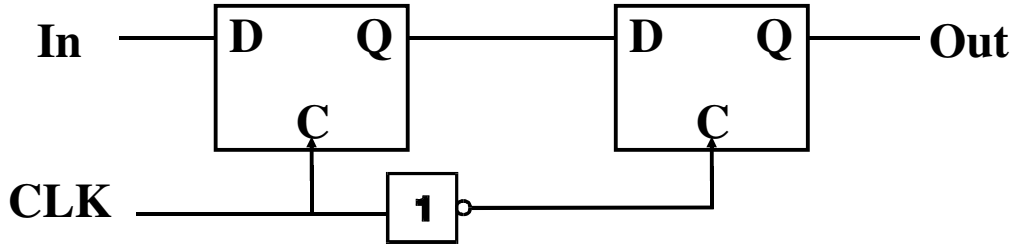
Truth table of D-Flipflop

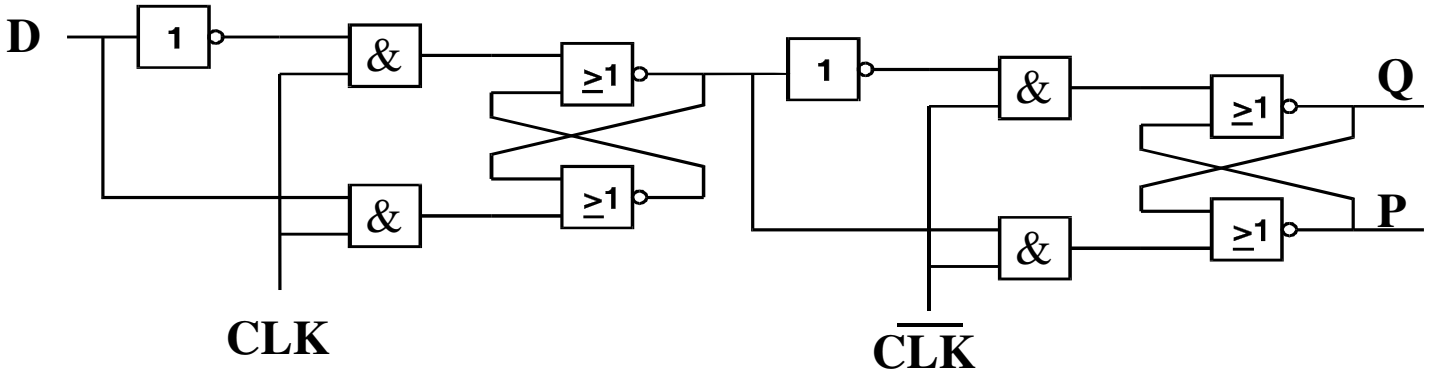


IN	CLK	OUT
X	0	OUT(t-1)
X	1	OUT(t-1)
0	↓	0
1	↓	1

Input data is only passed with (falling) edges.

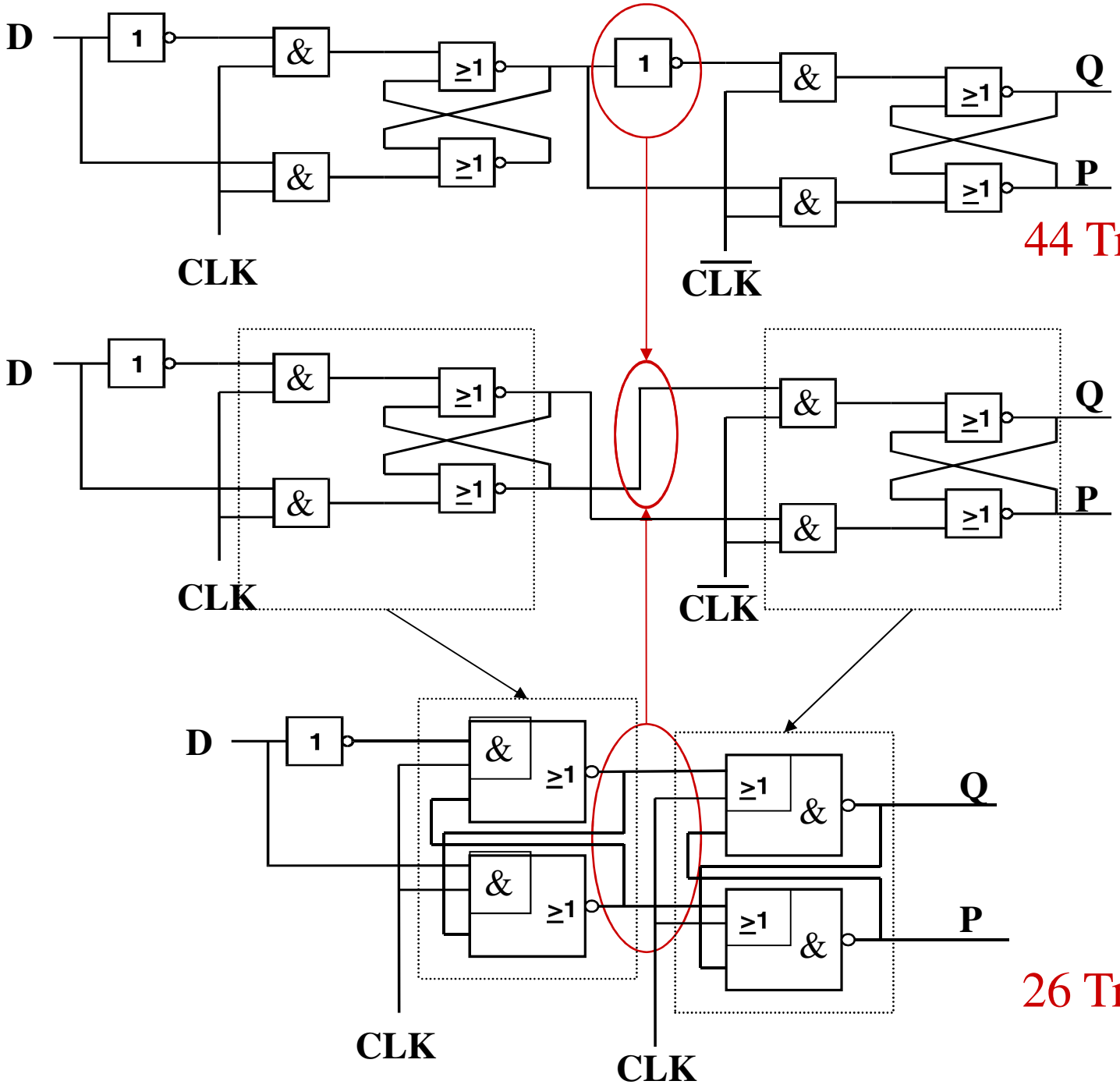
Gate-Implementation of Master-Slave-Flip-Flop





Master-Slave-Flip-Flop of two clocked D-Latches

44 Transistors


D
1
&
≥1
CLK
Q
P
44 Transistors
26 Transistors

Summary of terms

Flipflop: bistable flipflop, basis of all sequential circuits

Level controlled: **Latch**
Edge controlled (transition of level): Flipflop in a narrow sense

⇨ 1-Bit-memory

Register: an Array of FlipFlops.

A 16- bit Register is also an Array of 16 D-Flip-Flops

with common clock
with common Clear, Enable, etc.
In- and outputs are combined to Busses (Data, Addresses)

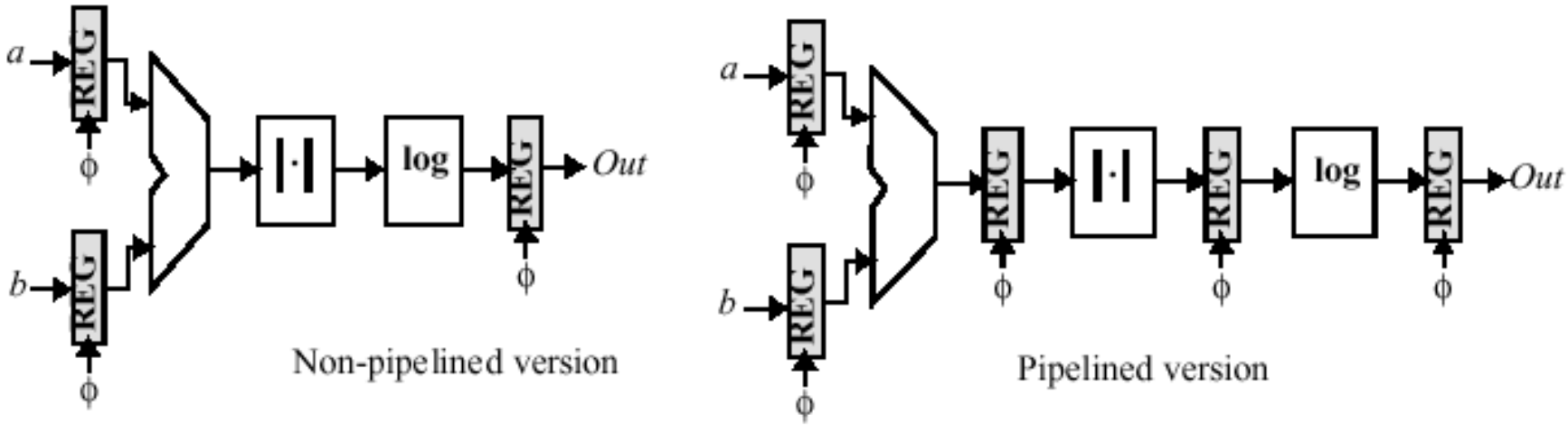
11. Sequential Circuits

1. Realization of Registers
   (Latches, Flip-Flops)
   1. Dynamic Registers
   2. Static Registers
2. **Pipeline**
3. **Clock Distribution – Clock Networks**
4. **Clock Generation**

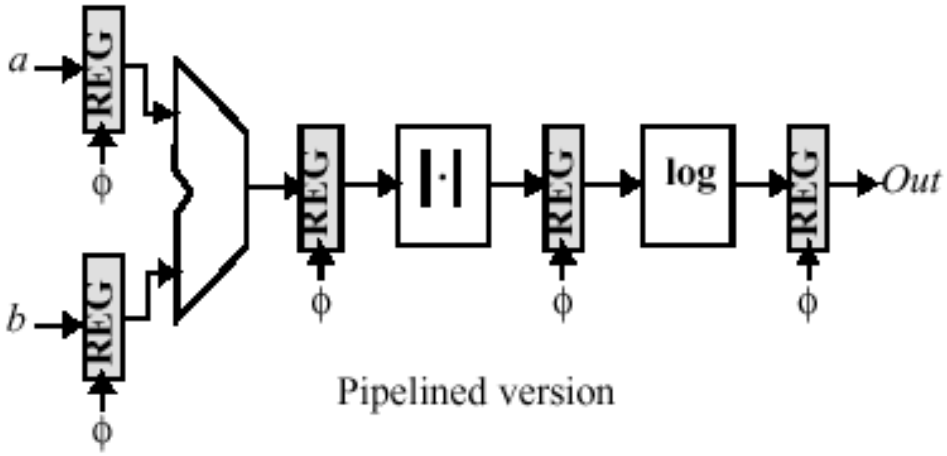
Pipeline

- Pipelining = Consecutive processing of a task.
  Single steps are finished in one clock each
- Registers are needed for several steps/stages with logics inbetween
- The result is only available after a long **Latency** (counted in clocks).

$$\text{Processing}: Out = \log|a+b|$$



$Pipeline \quad for \quad Out = \log|a+b|$

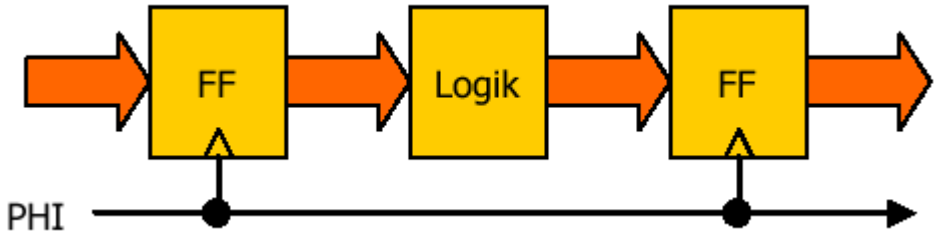


Pipelined version

Clock Period	Adder	Absolute Value	Logarithm
1	a1+b1		
2	a2+b2	|a1+b1|	
3	a3+b3	|a2+b2|	log(|a1+b1|)
4	a4+b4	|a3+b3|	log(|a2+b2|)
5	a5+b5	|a4+b4|	log(|a3+b3|)

Pipeline-Implementation

1) Flipflops, one phase



2) Latches, two phases
(not overlapping)