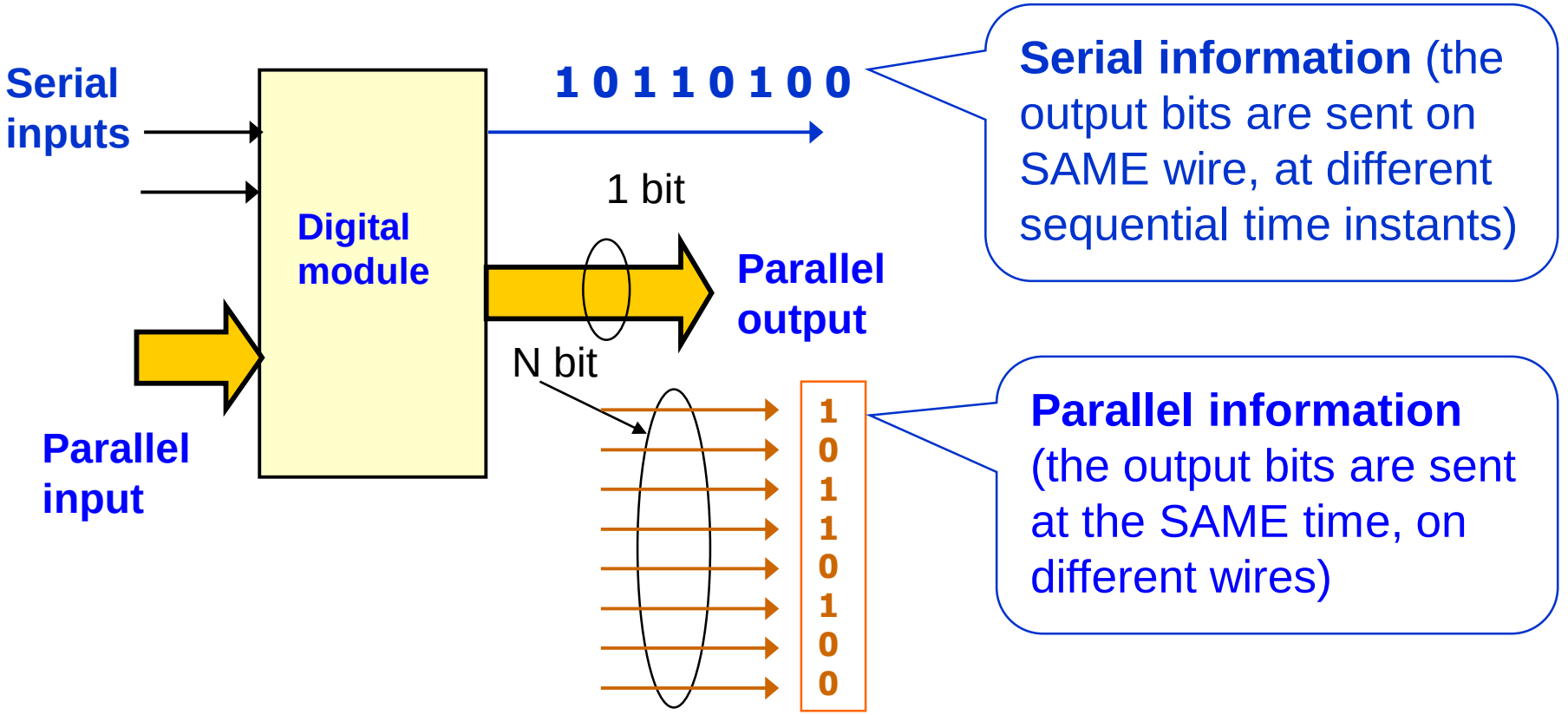
Applied Electronics

Sequential Circuits

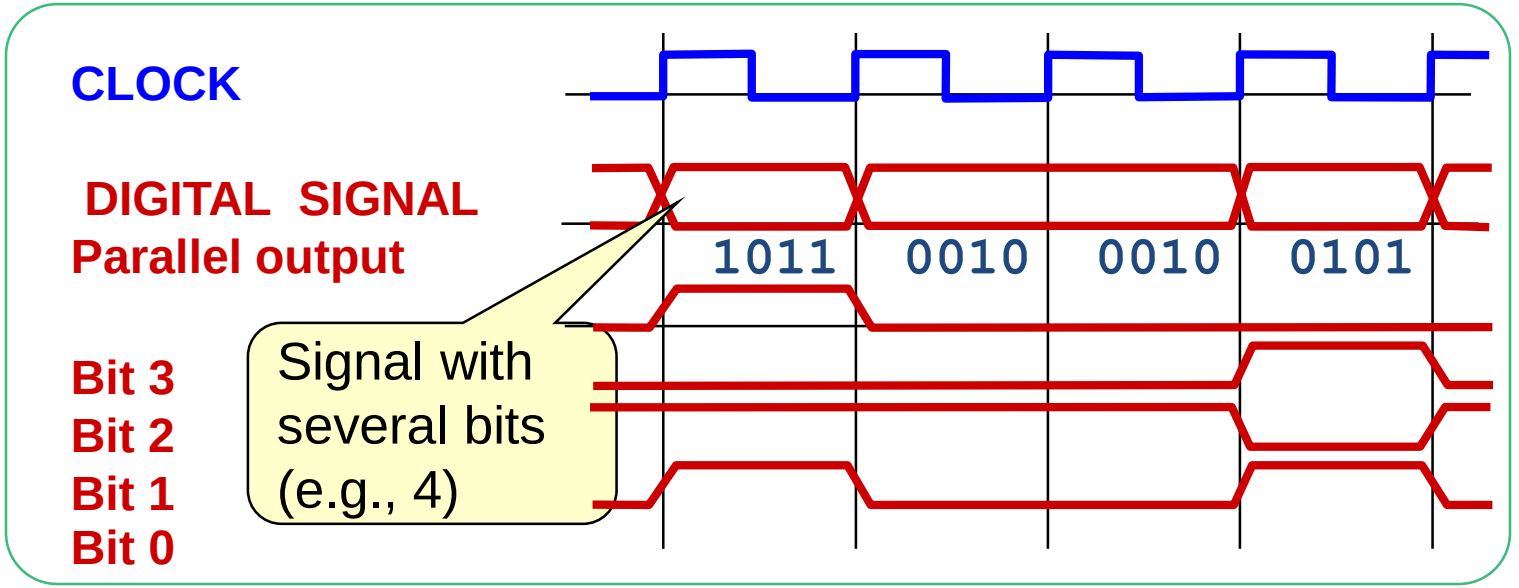
Serial and Parallel Signals

- Digital signals can be represented and transferred in serial or parallel mode



Parallel Data Transfer

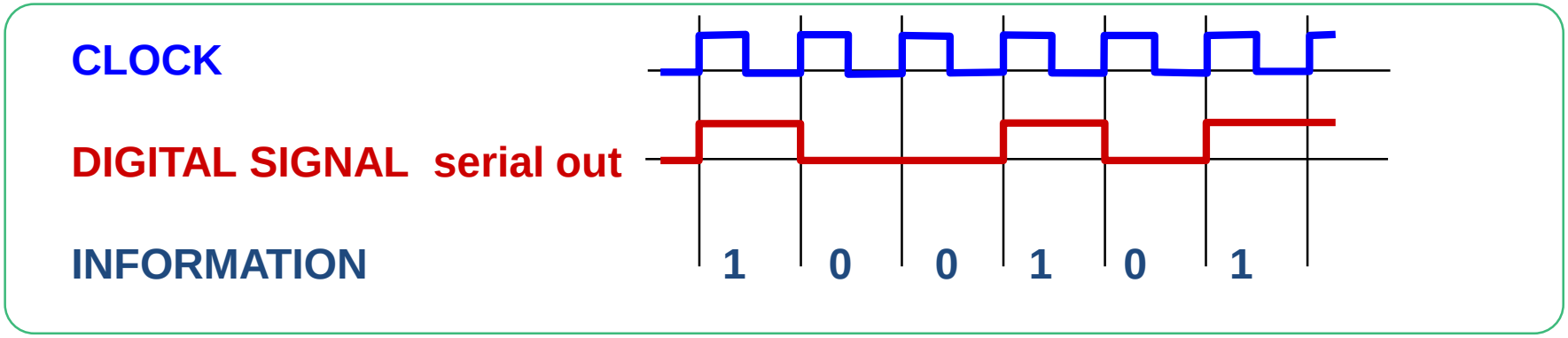
- The clock determines the signal rate and ensures synchronization



- Parallel transfer: N bits in one clock cycle
  - N bits at the same time => total time T_{ck}

Serial Data Transfer

- The clock determines the signal rate and ensures synchronization



- Serial transfer rate: N bits need N clock cycles
  - 1 bit per CK cycle
    => total time for data transfer: $N\ T_{ck}$

Benefits and Drawbacks

- Parallel connection
  - Fast, sends several bits at the same time
  - Many lines: higher power consumption and cost
  - Used mainly for short distances
    - Inside ICs, on PCBs among ICs
    - Buses: PCI, ATA, …
- Serial connection
  - Slow, one bit at a time (but “time” can be very short)
  - A single line: reduced power consumption and cost
  - Used mainly for long distances and “channels”
    - Ethernet, SATA, USB
    - Radio channels

Review of FF Synchronization

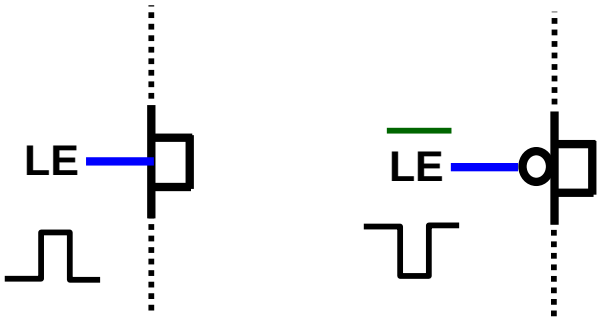
- Level-sensitive (LE command)
  - Latch registers
  - Transparent when enabled (LE = 1)
  - Memory when blocked (LE = 0)
- Edge-sensitive (CK command)
  - Master-slave registers
  - Store input state on the active transition of the CK
  - No transparent mode
- LE and CK
  - Active Low / High
  - H→L / L→H

FF and Latch Synchronization Symbols

Four basic synchronization types:

LATCH

POSITIVE **NEGATIVE**

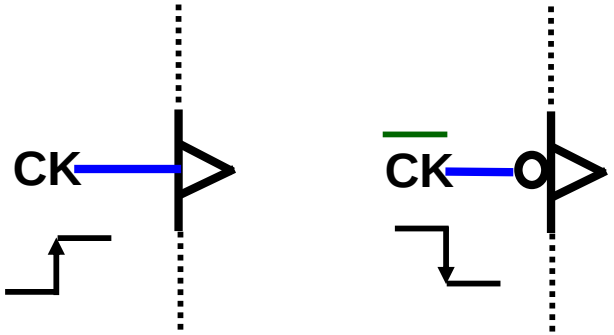


LATCH
(LEVEL sensitive)

Output can change at any time in the transparent phase (LE = 1)

EDGE-TRIGGERED

POSITIVE **NEGATIVE**

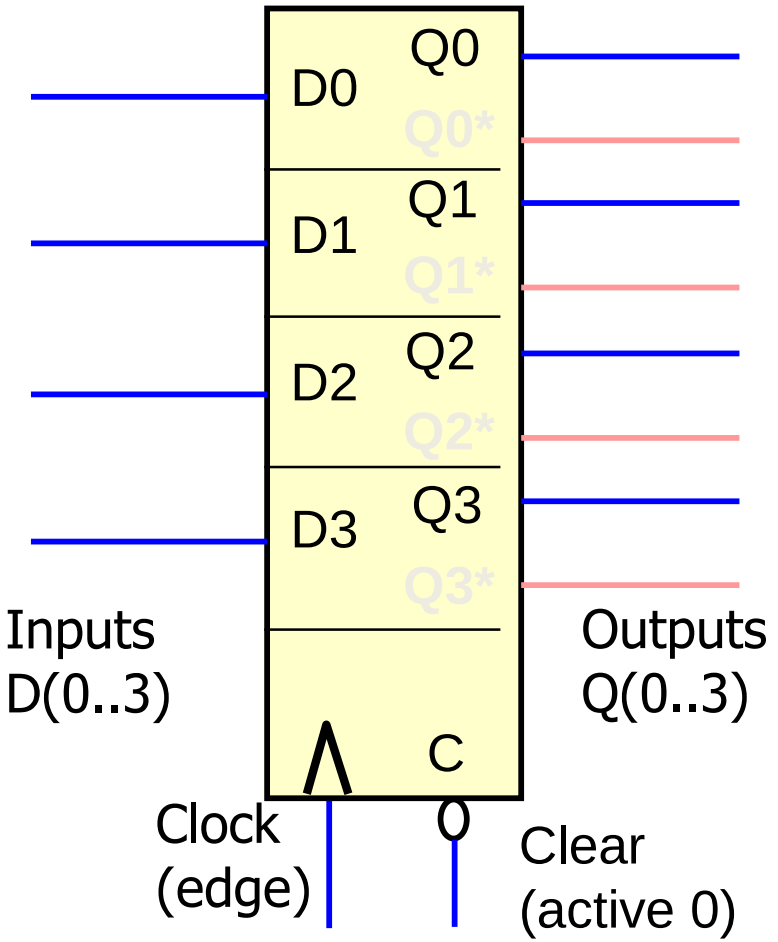


EDGE-TRIGGRED
(EDGE sensitive)

Output can change **only** on the (active) clock edge

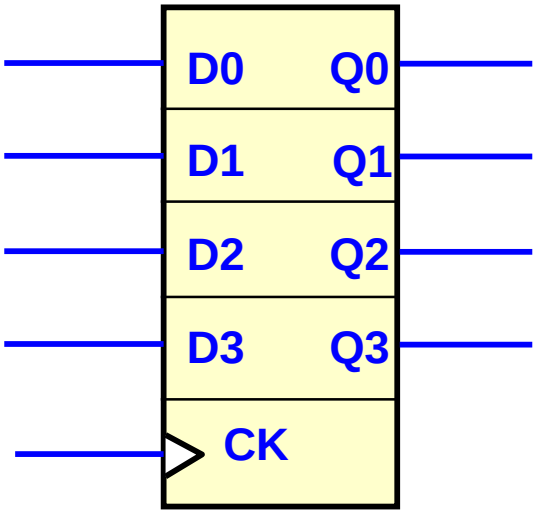
Registers

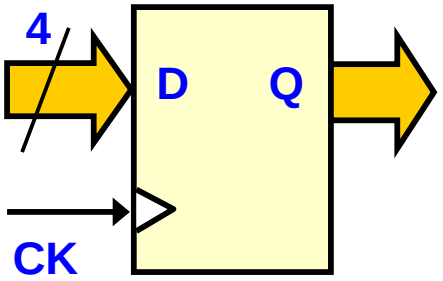
- A group of FF with shared commands
  - Clock
  - Reset/Clear
  - …
- Register type given by the FF clocking
  - Latch
  - Edge-Triggered



Parallel Register

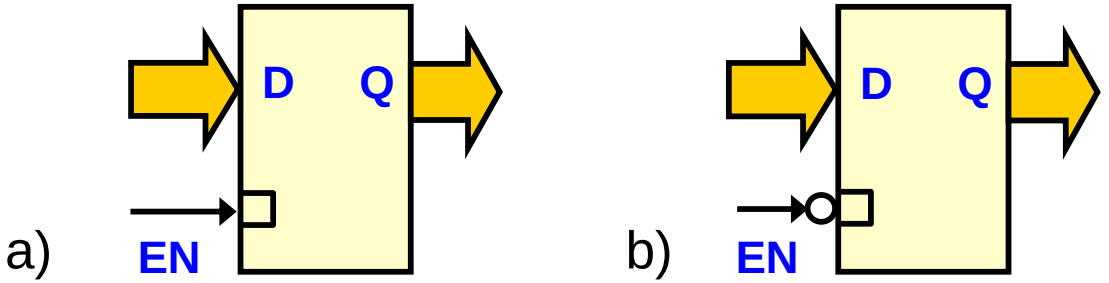
- Parallel input
- Parallel output
- PIPO Register
  - Parallel In/Parallel Out
- **Parameters:**
  - Bit number *N*
  - Sync (clock) type
    - Level (latch)
    - Edge (D-FF, master-slave)
  - Other commands
    - Clear, OE, …



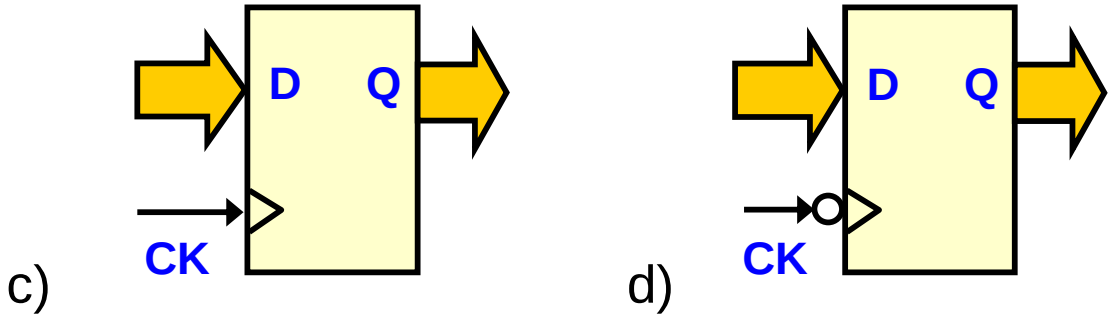


Synchronization Signals

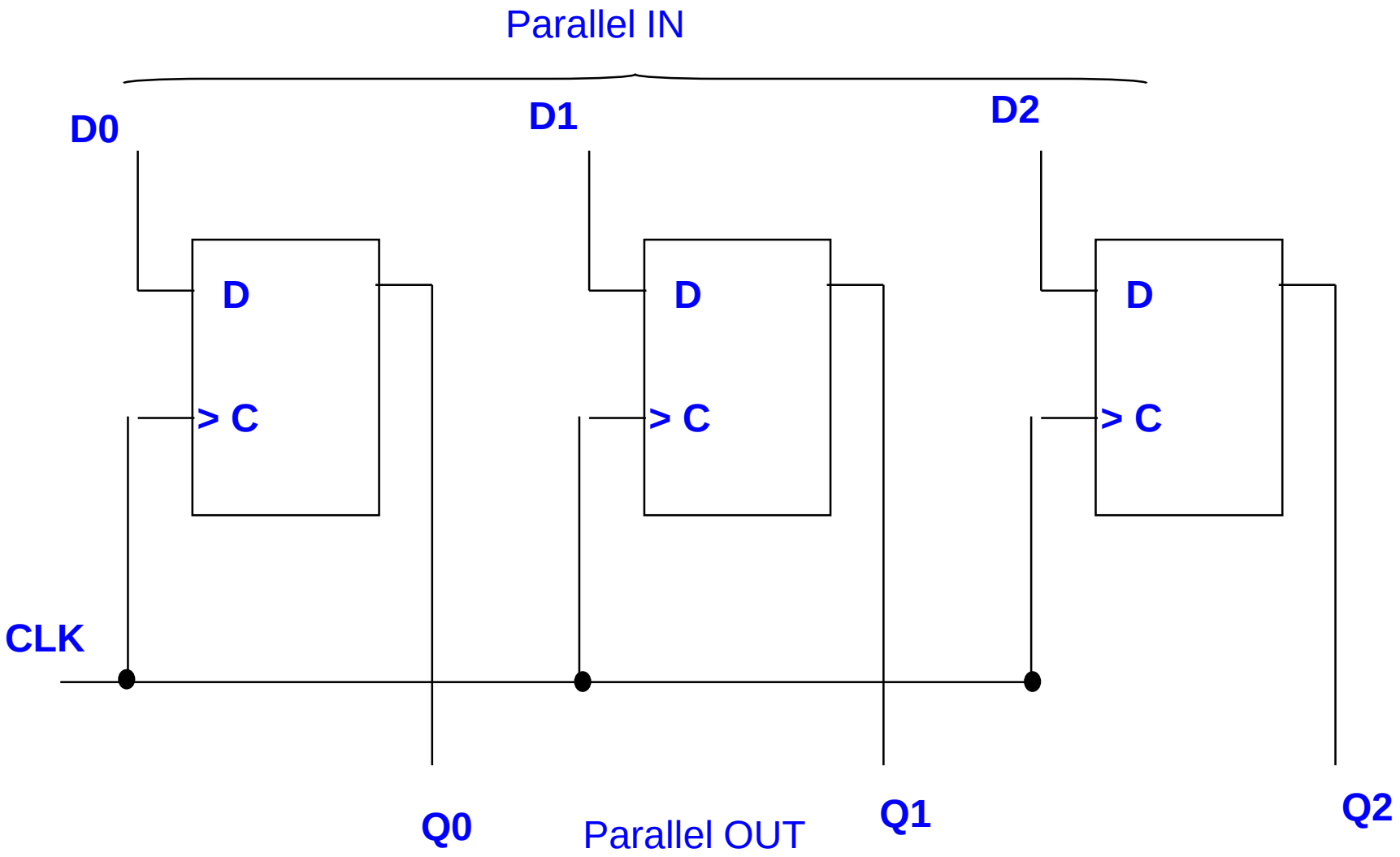
- Level enabled



- Edge triggered

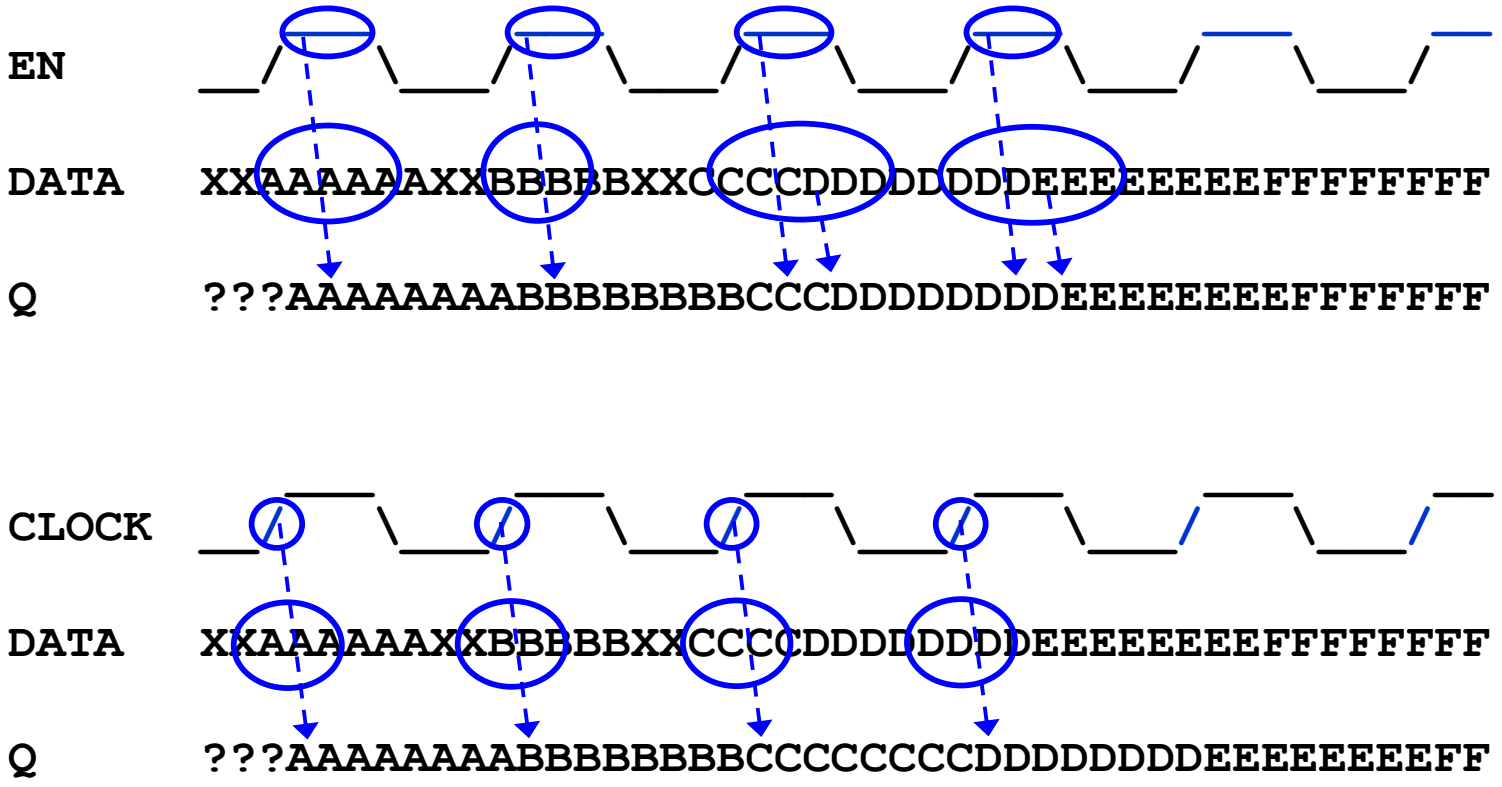


Example: PIPO With D-FF



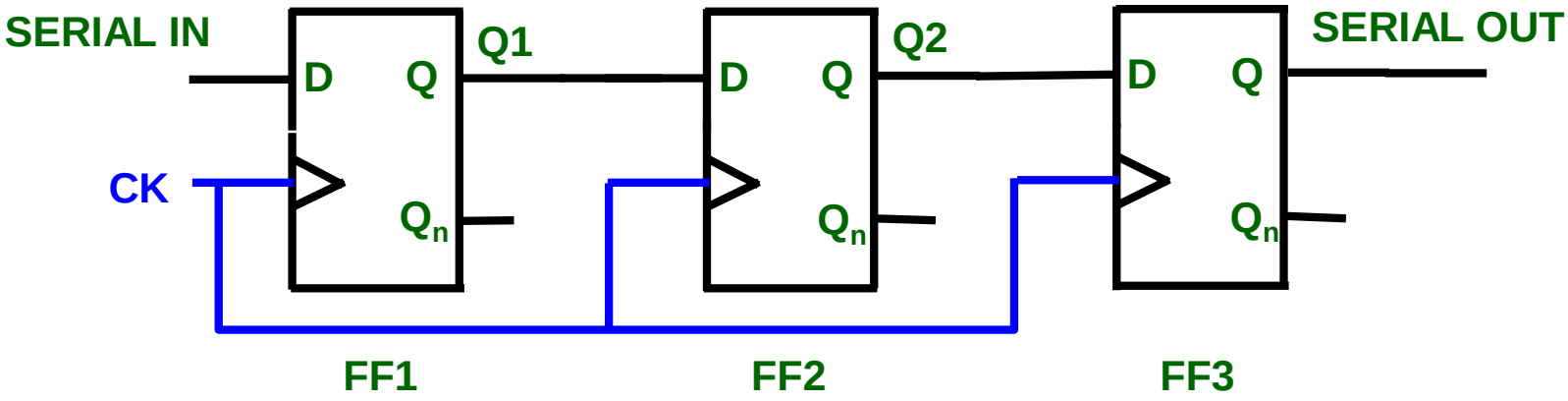
PIPO Register Timing

- Latch and master-slave with the same inputs

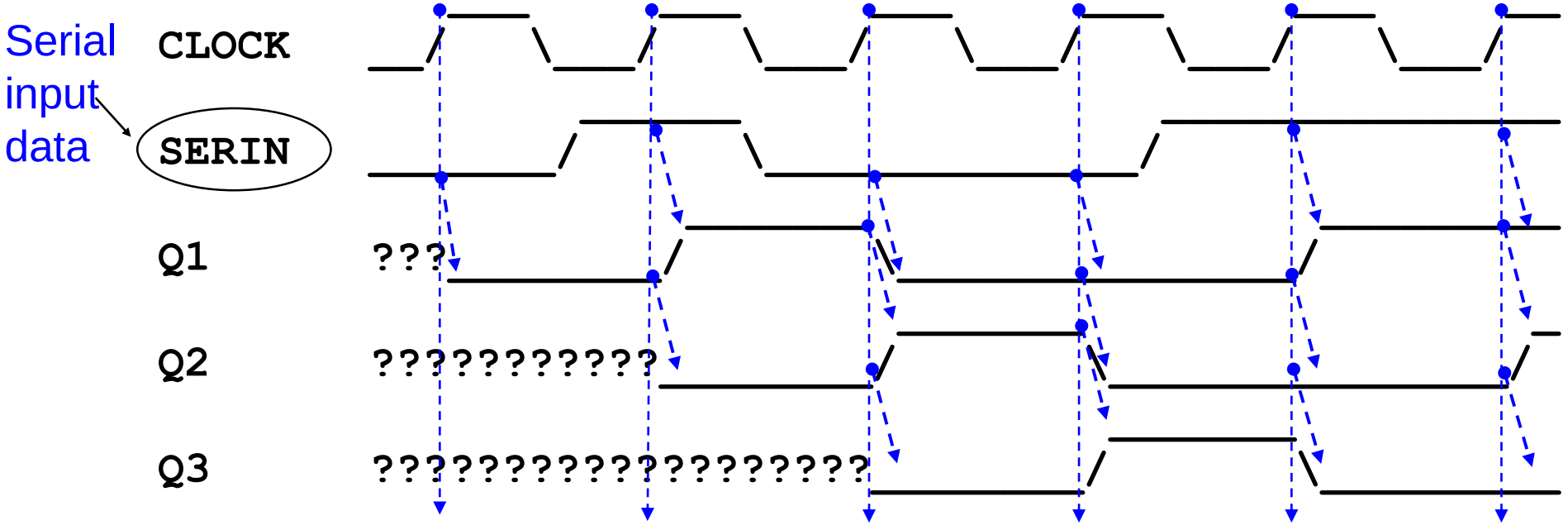


SISO Shift Register

- A set of D-FFs with chain connection ($Q_n \rightarrow Q_{n+1}$)
  - Common Clock CK (common Reset, if any)
- The input Data is moved through the chain
  - Serial In → Q1 → Q2 → Serial Out
- Shift Register: Serial In – Serial Out: SISO



SISO Register Timing



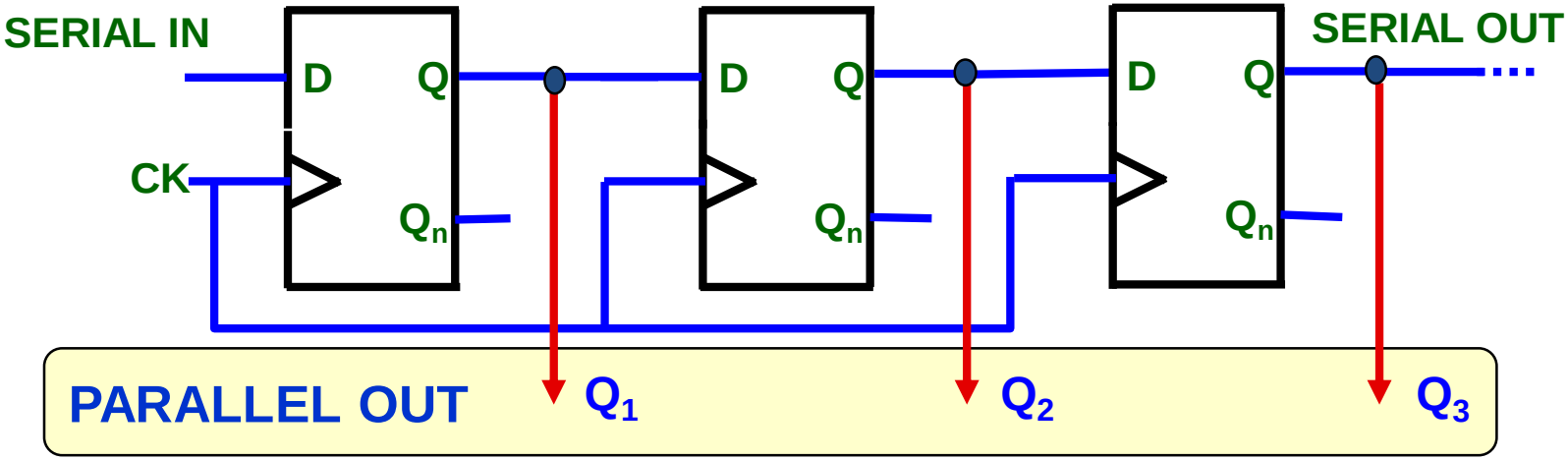
Only the Q_3 output is available outside the device.
(or Q_n for a N-stage register).
The SERIN is available in Q3 after 3 clock cycles

Shift Register SIPO

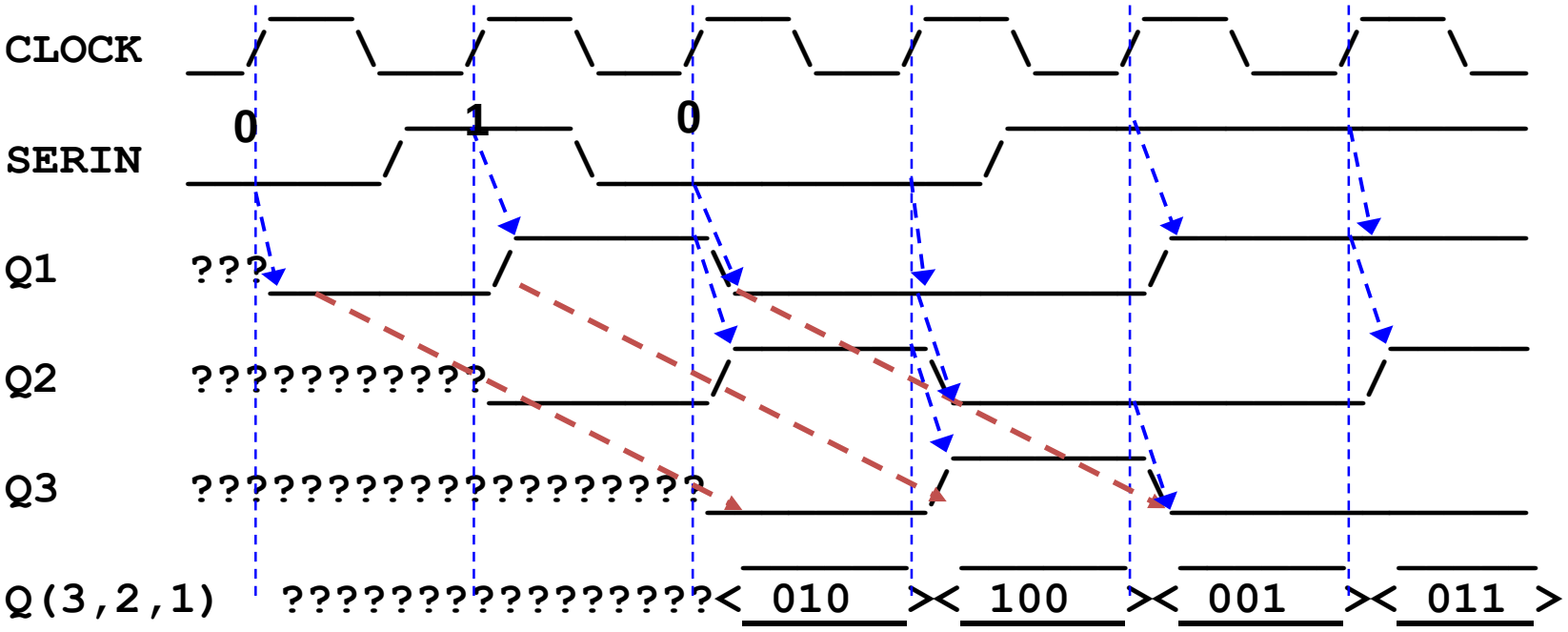
- Serial data transmission is widely used by peripherals to exchange with a computer
  - USB (universal serial bus) connects peripherals
- The computer processes data in parallel
  - We need serial-to-parallel conversion

Shift Register SIPO

- A set of D-FF with chain connection ($Q_n \rightarrow D_{n+1}$)
  - Common clock CK (common Reset, if any)
- Can convert serial data into parallel
  - SIPO register: Serial In → Parallel Out



SIPO Register Timing

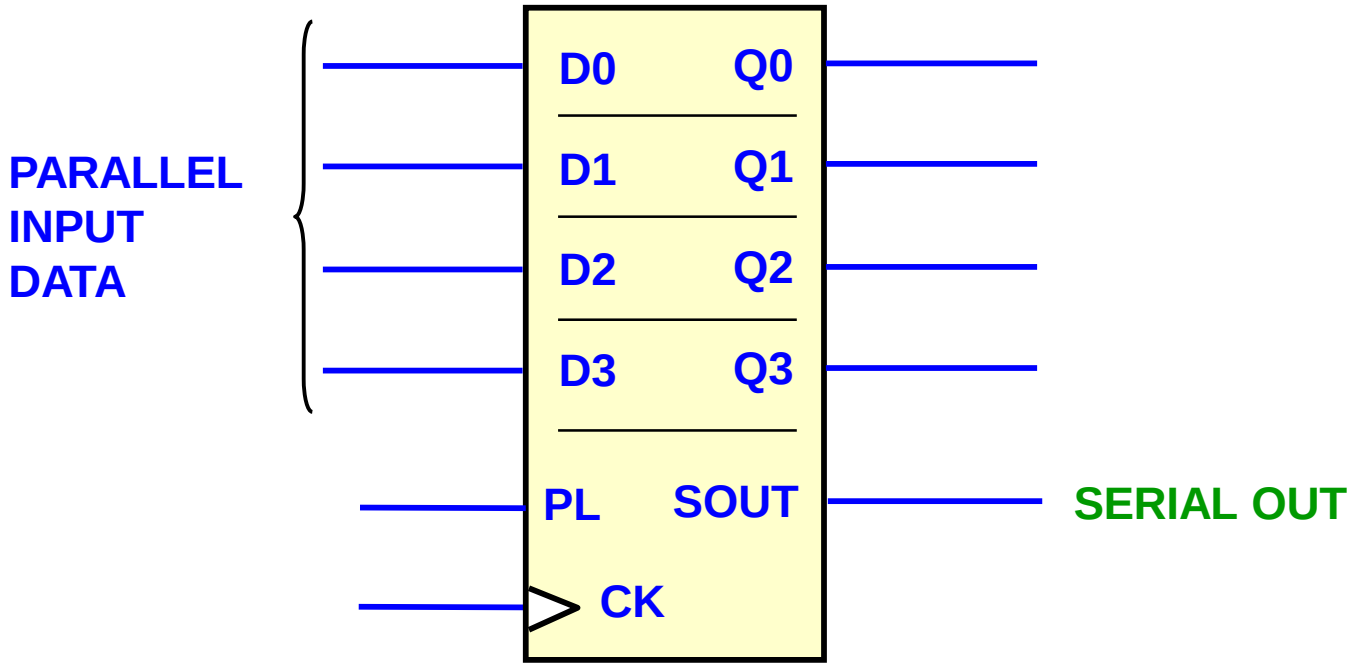


All Qi outputs are available

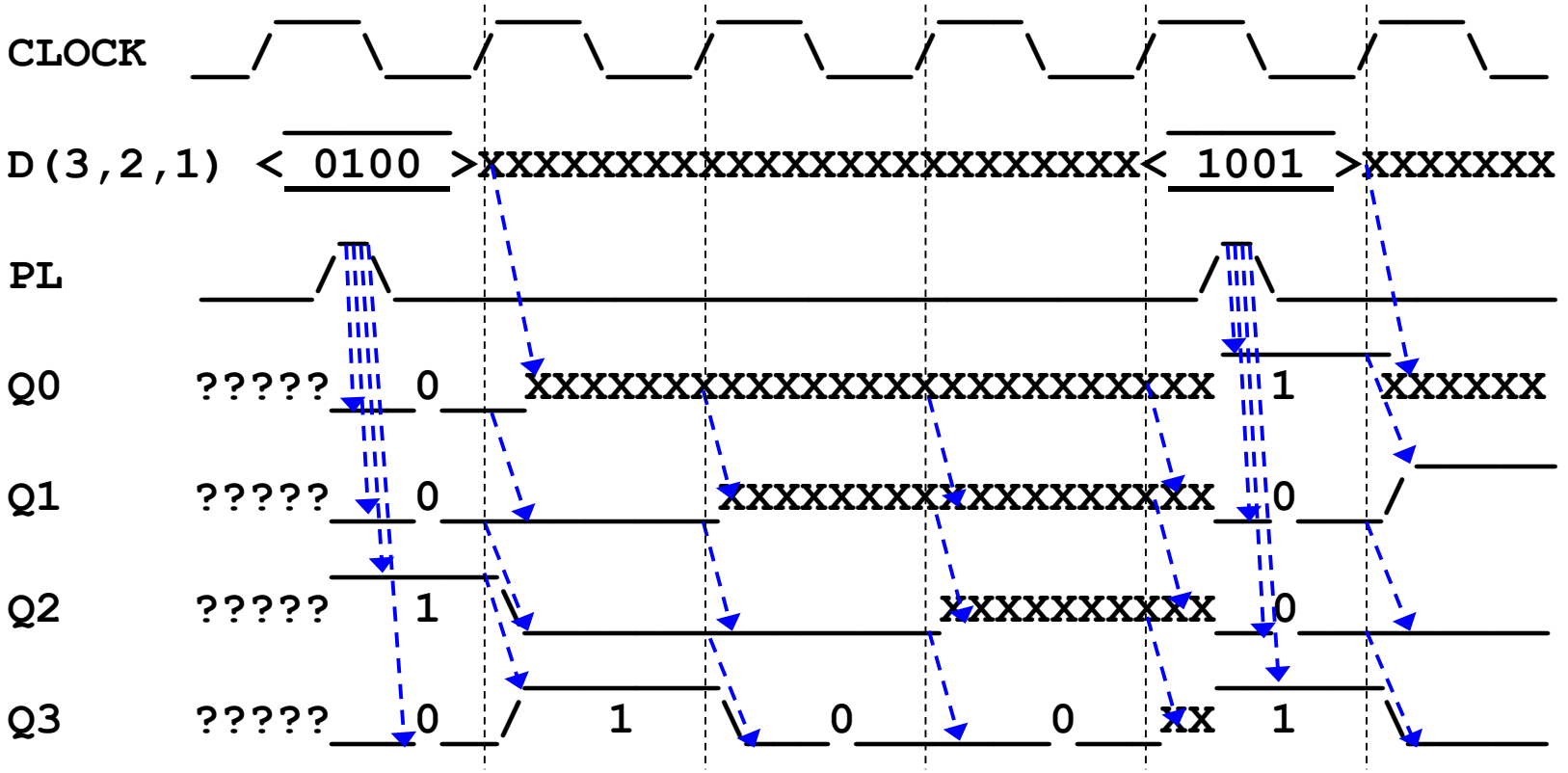
the input data (010011…) is available in parallel format (3 bits) on <Q1, Q2, Q3>

PISO Shift-Register

- It allows to load all FFs with a single operation
  - Parallel Load command (PL)
  - Both parallel and serial outputs are available



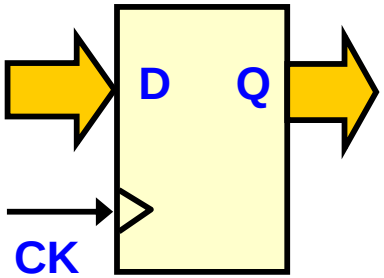
PISO Register Timing



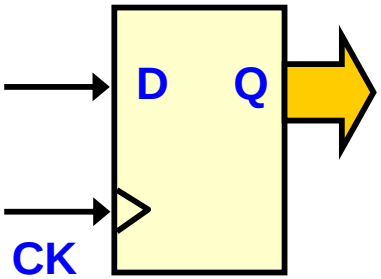
The Data D(3,2,1,0) is loaded in the FF on a Parallel Load pulse (PL)
The serial output SEROUT corresponds to Q3 (sequence 0 1 0 0)

Register Summary

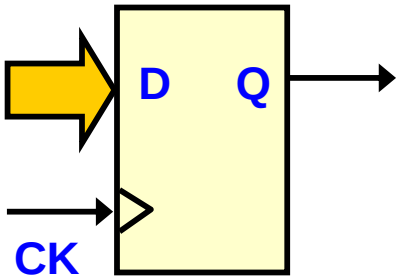
- PIPO



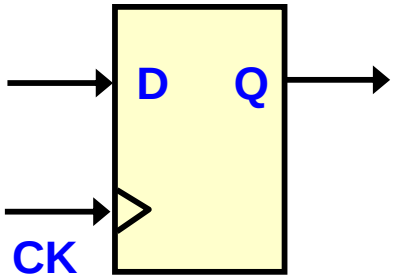
SIPO



- PISO

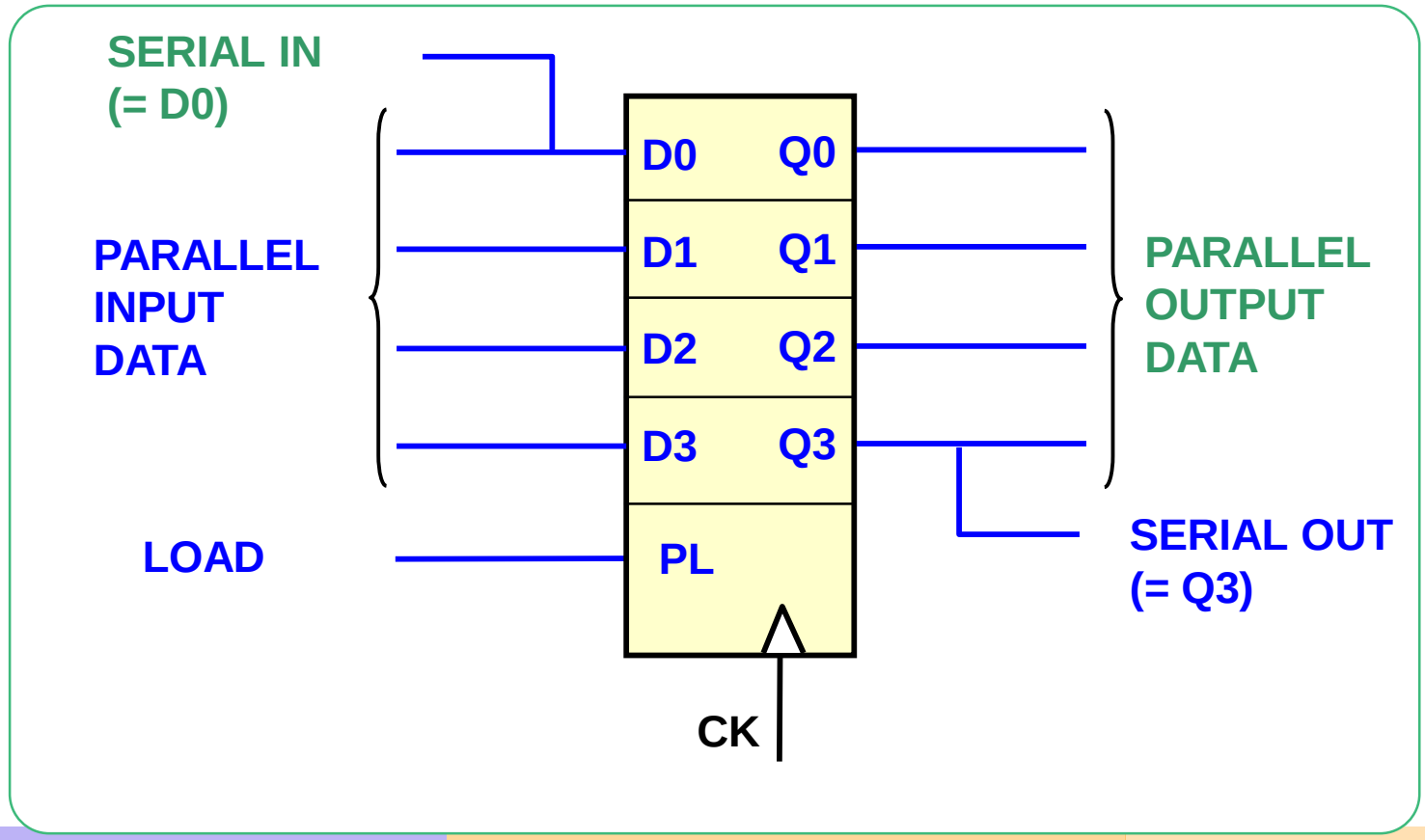


SISO



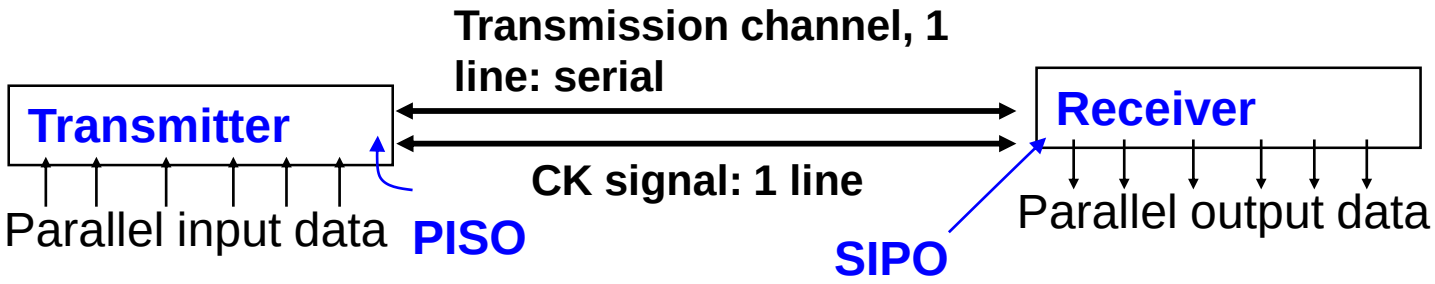
Shift-Register Complete Version

- It can manage both serial and parallel input and output

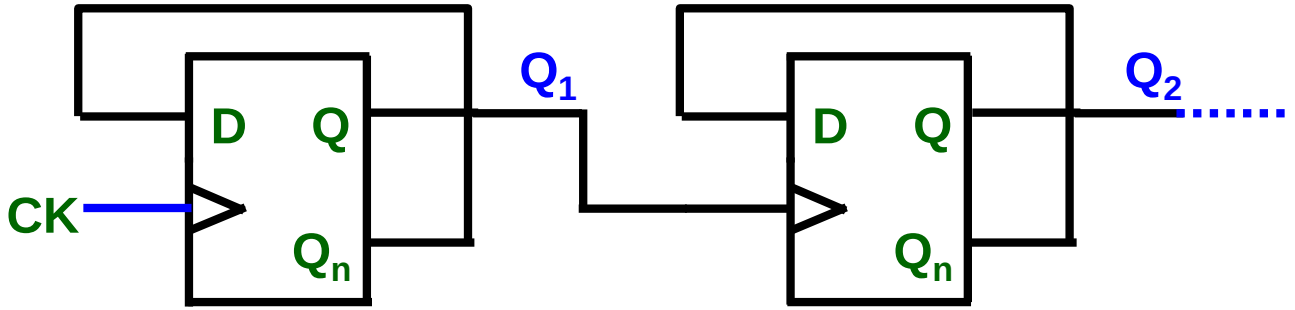


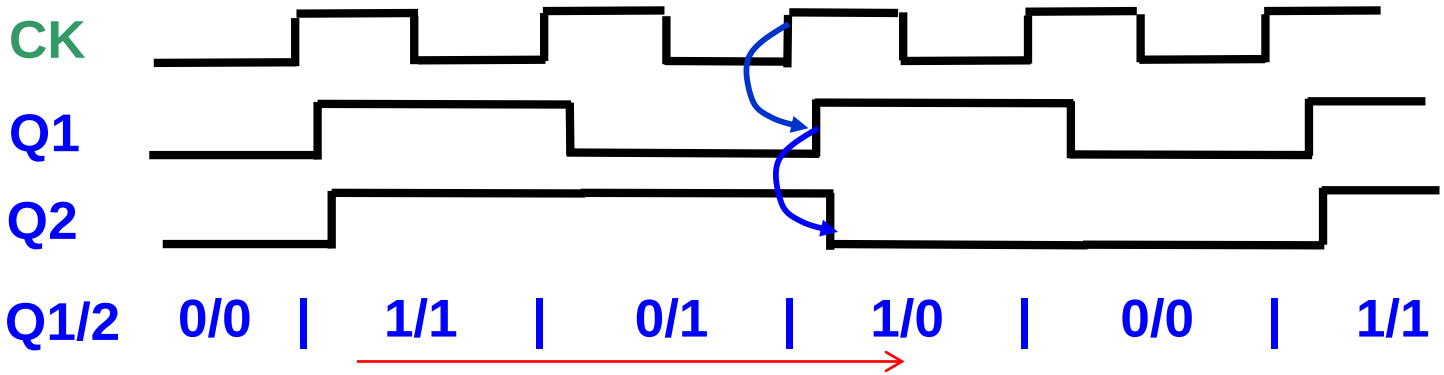
Application Example: Serial Communication

- Serial output over channel; just one line needed
- Used in long-distance communication
- Same CK for transmitter and receiver
  - CK sent over the channel
  - Or the receiver has a CK recovery circuit
    - Can recover the CK from the data sequence



Divider Modulus 2/4 (Ripple)





Q1 changes (0,1,0,1…) at each CK rising edge

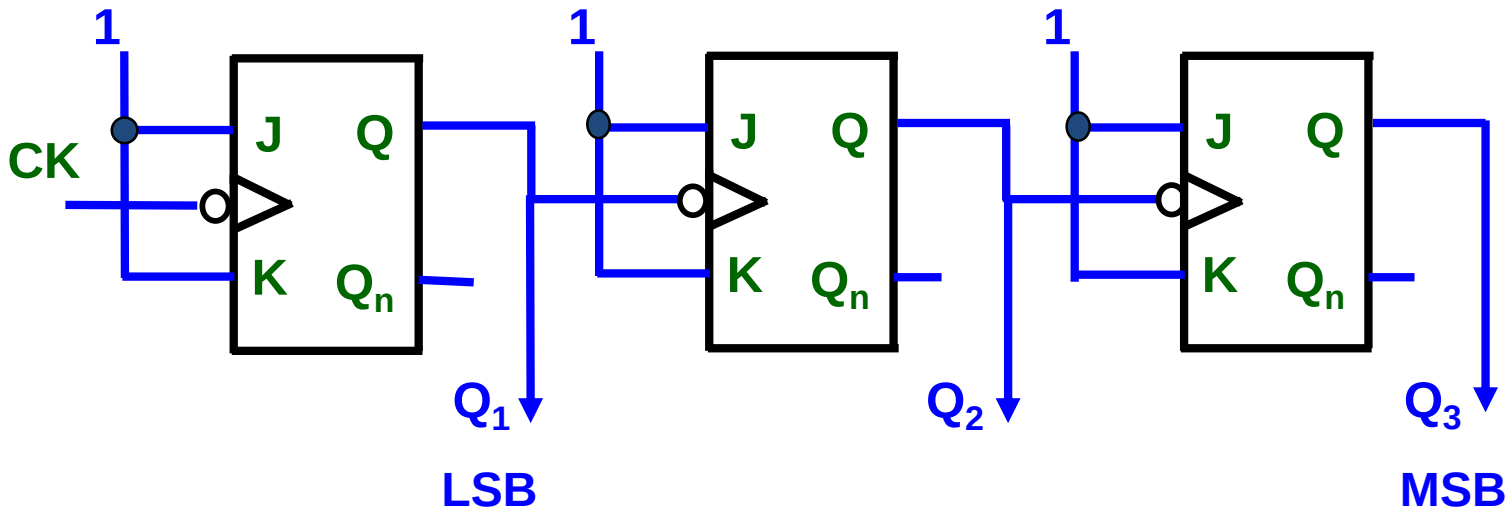
Count down: the output is 11, 10, 01, 00

Each stage divides the clock rate (CK) by 2

M stages: division modulus 2^M

Counter With JK-FF

- JK-FF toggles on each clock pulse when J = K = 1
  - JK-FF can be used to build counters
- Example: three stage counter (:8)
  - JK-FF Falling-Edge-Triggered

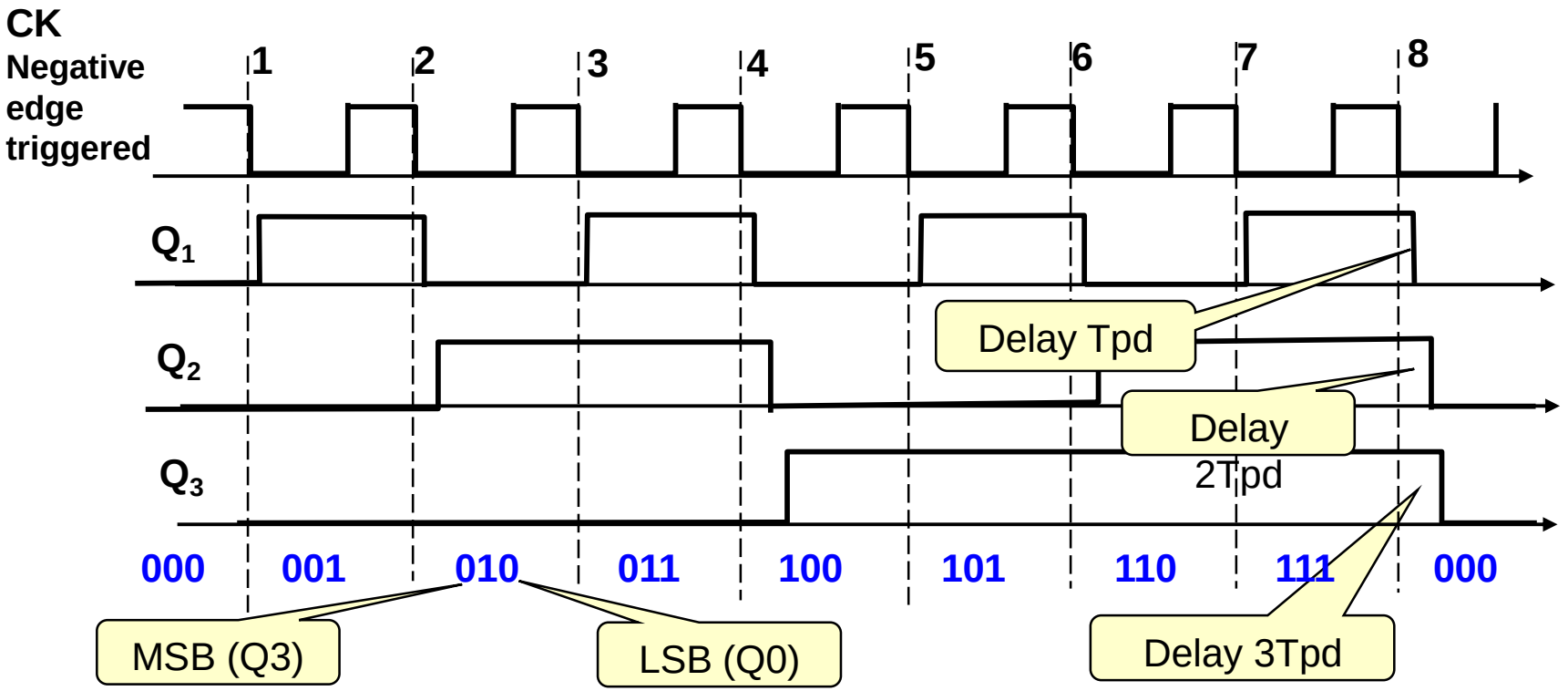


Asynchronous Counter

- Clocks connected in a chain
- Switching delays add up along the chain
- The circuit is an **asynchronous counter**
  - The outputs switch with different delays
    - Single FF: delay T_{pd} (from CK to Q)
    - Q_M output: delay $T_{pdM} = M\ T_{pd}$

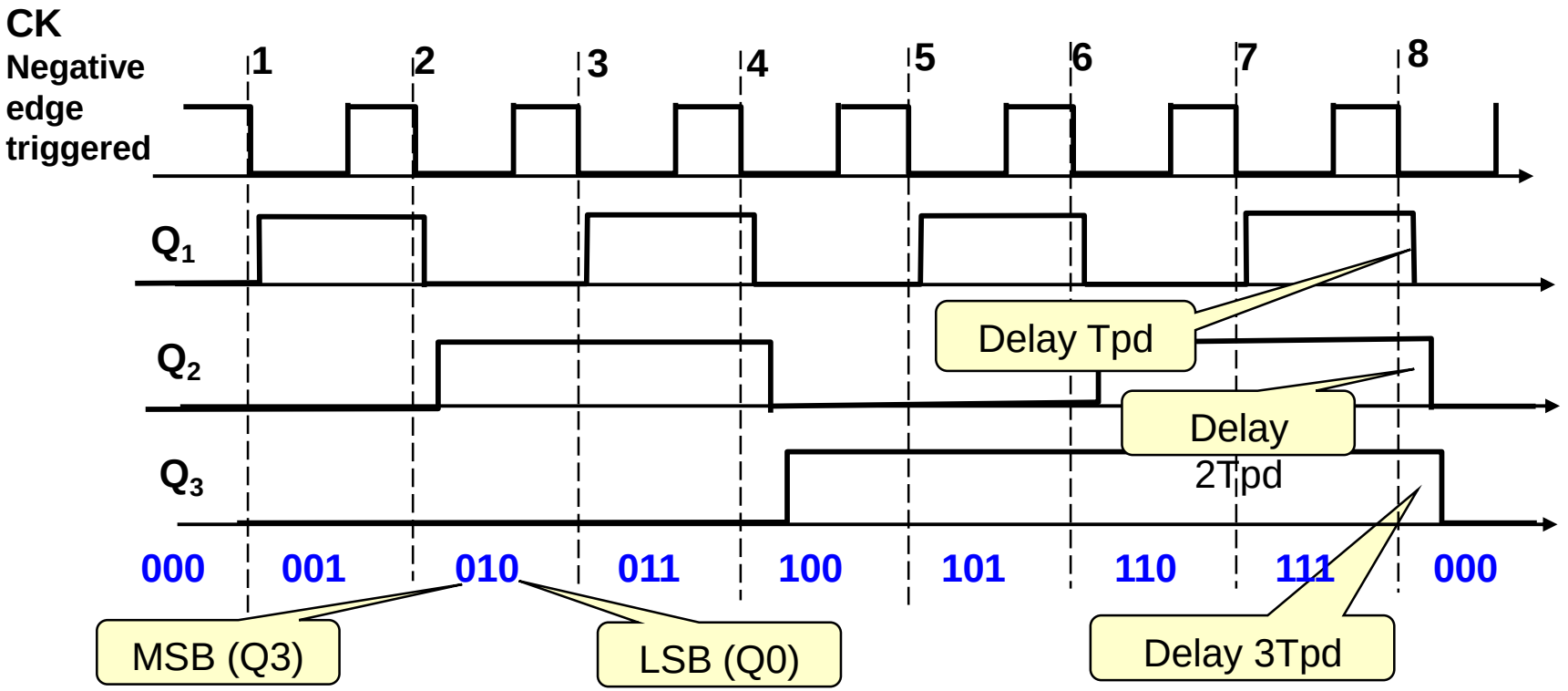
Asynchronous Counter Timing

- Each stage toggles at half the rate of the previous one
- Acts as a frequency divider by 2^n (n is the number of FFs)
- Generates the sequence of binary numbers



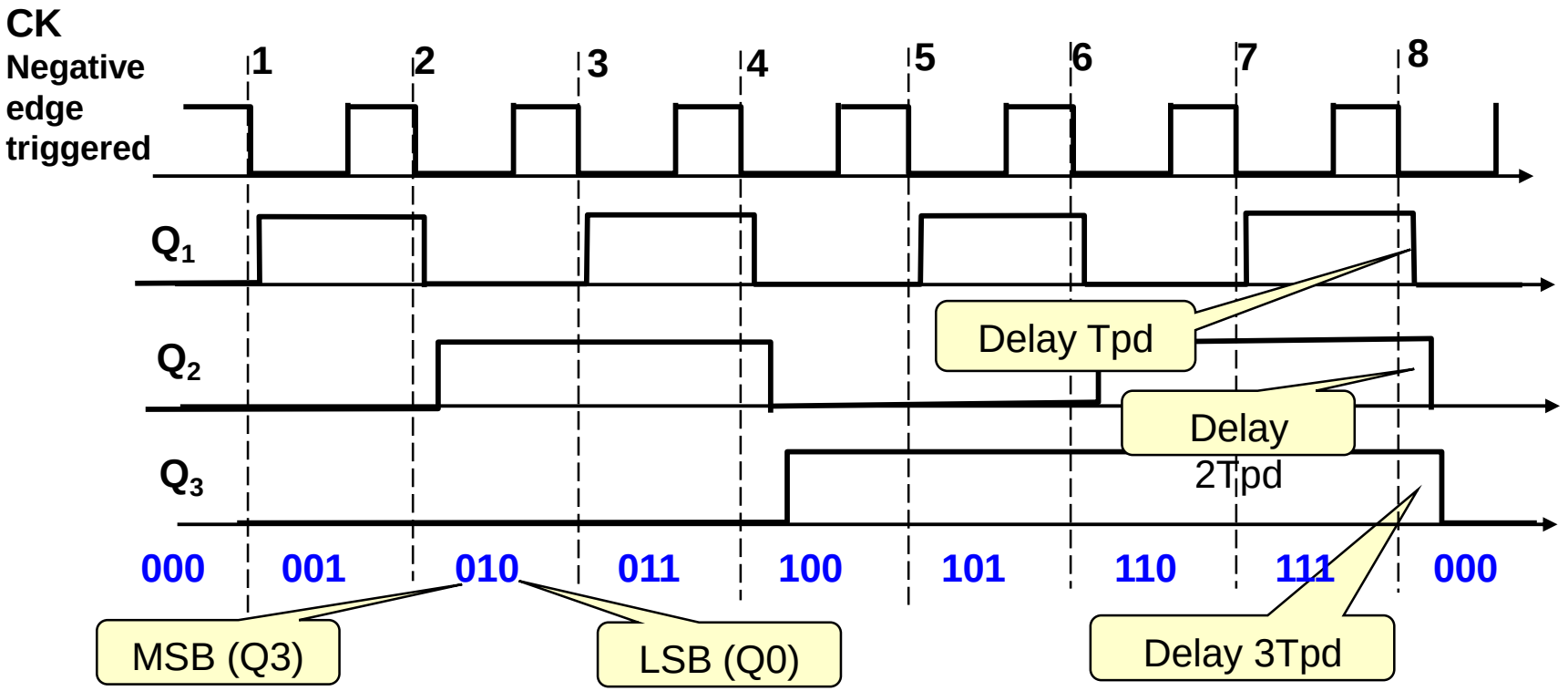
Asynchronous Counter Timing

- Each stage toggles at half the rate of the previous one
- Acts as a frequency divider by 2^n (n is the number of FFs)
- Generates the sequence of binary numbers



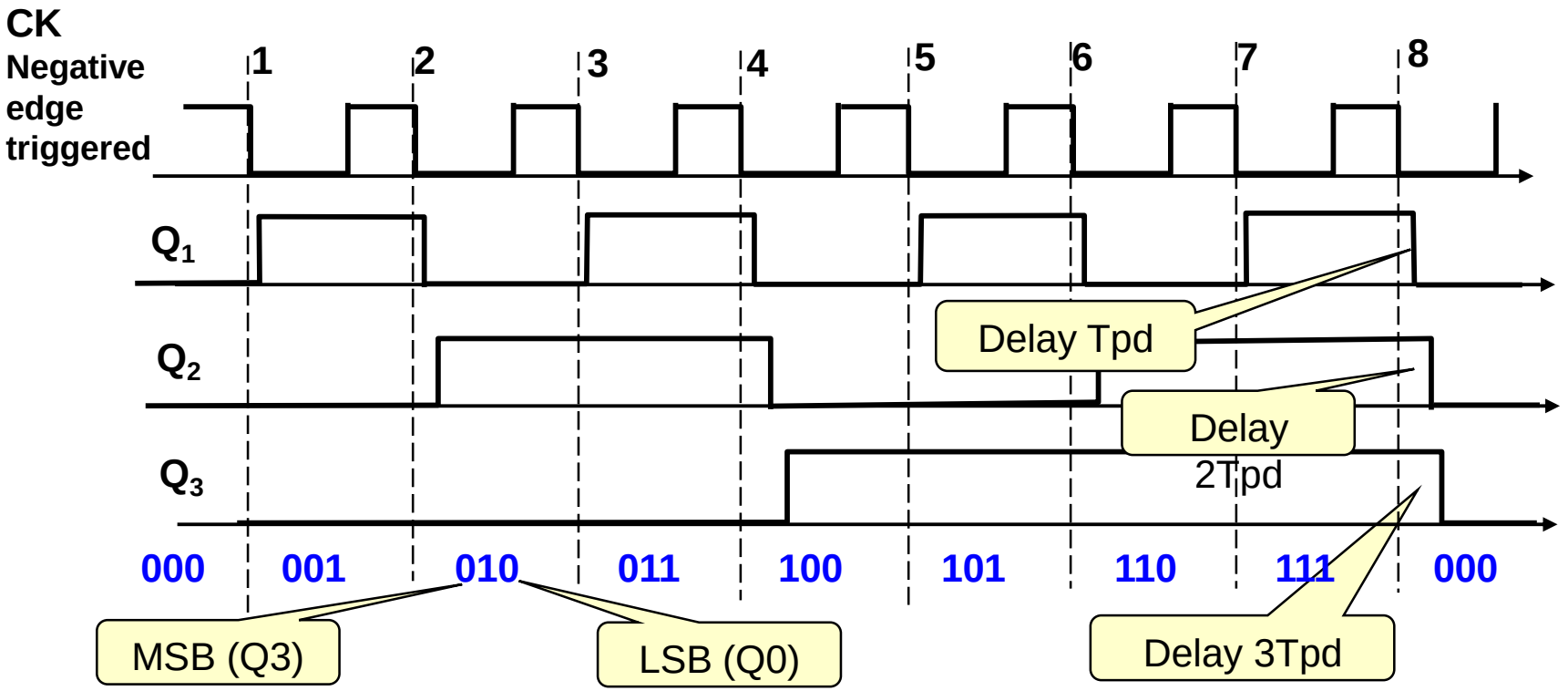
Asynchronous Counter Timing

- Each stage toggles at half the rate of the previous one
- Acts as a frequency divider by 2^n (n is the number of FFs)
- Generates the sequence of binary numbers



Asynchronous Counter Timing

- Each stage toggles at half the rate of the previous one
- Acts as a frequency divider by 2^n (n is the number of FFs)
- Generates the sequence of binary numbers



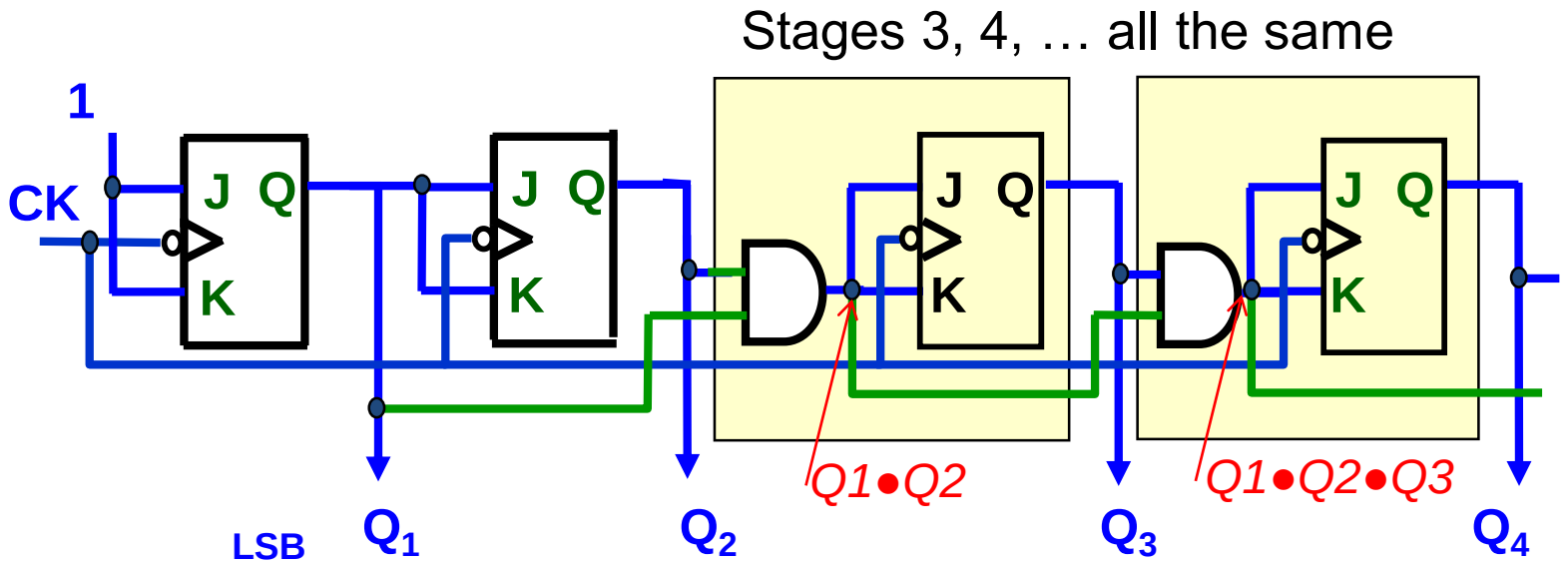
Sync and Async Counters

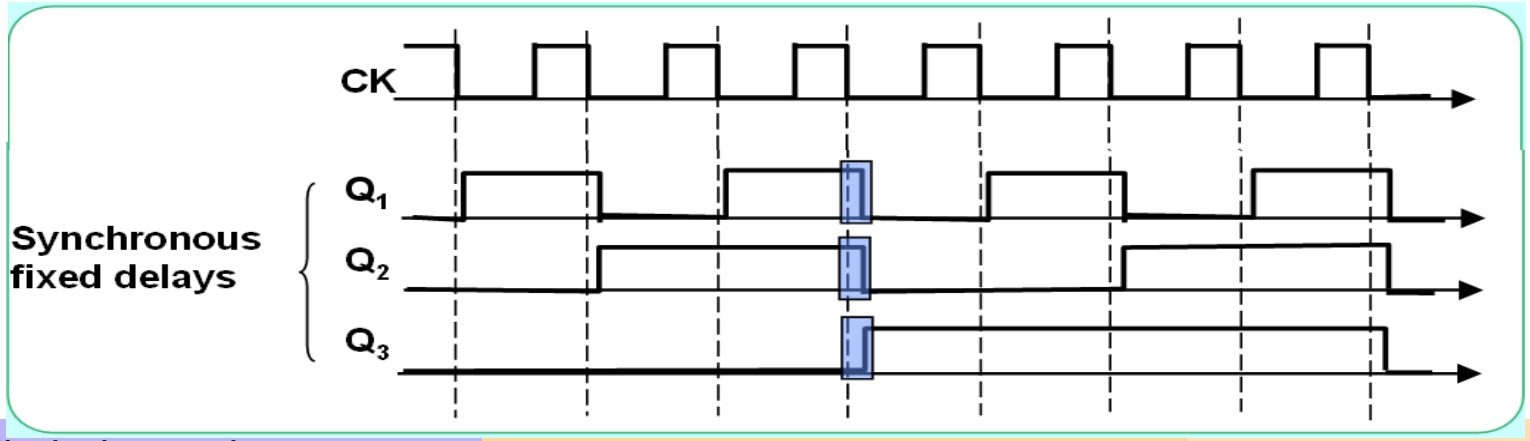
- Asynchronous counters
  - The various FFs receive different Clock signals
    - Different delays at various outputs
    - Possible races in transients (transient states)

- Synchronous counters
  - All FF receive the same Clock signal
  - All outputs switch with the same delay (**synchronously**)

Example of Synchronous Counter

- Modulus 2^N counter (N = 4) - Direct clock to all FFs

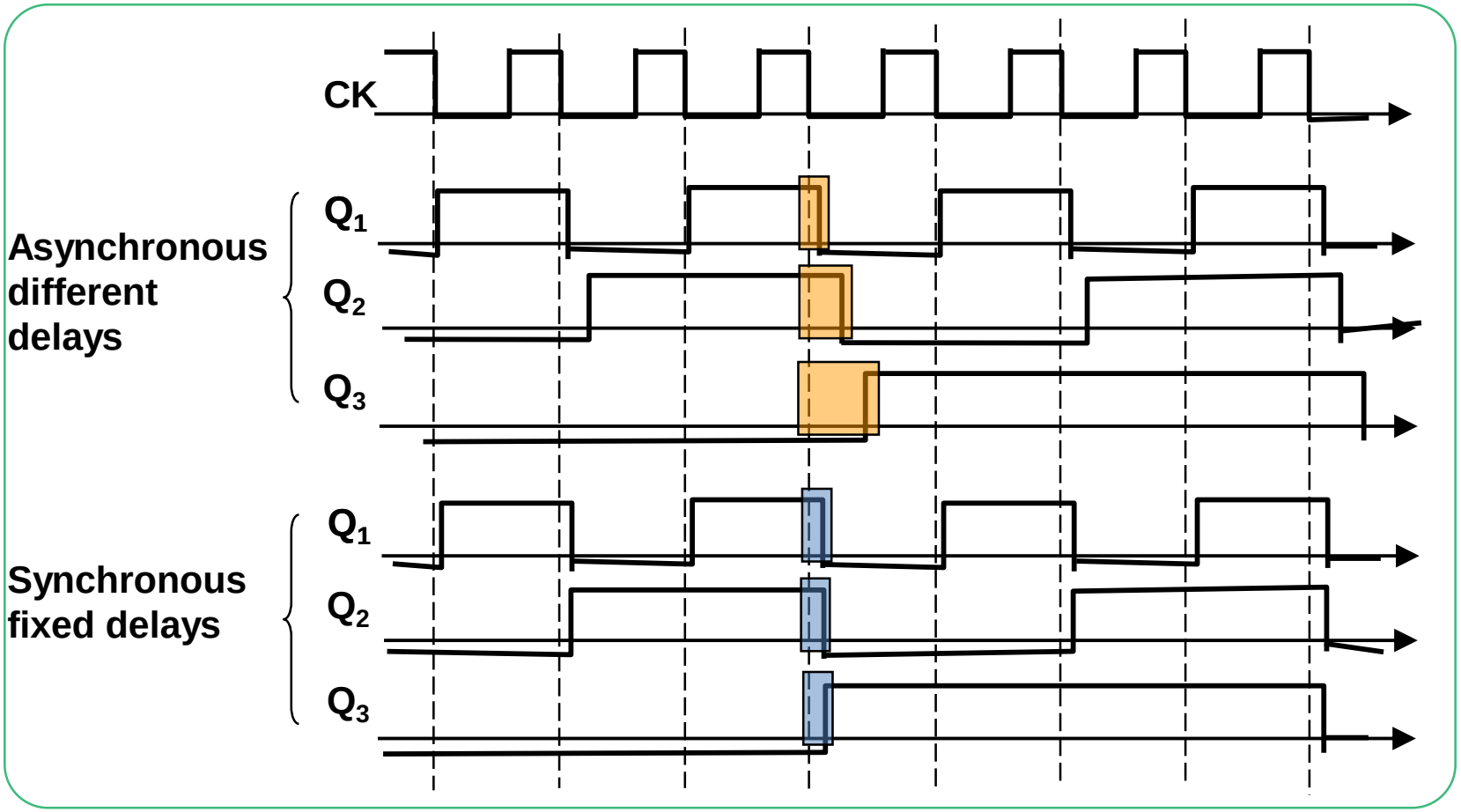




Synchronous Counter Characteristics

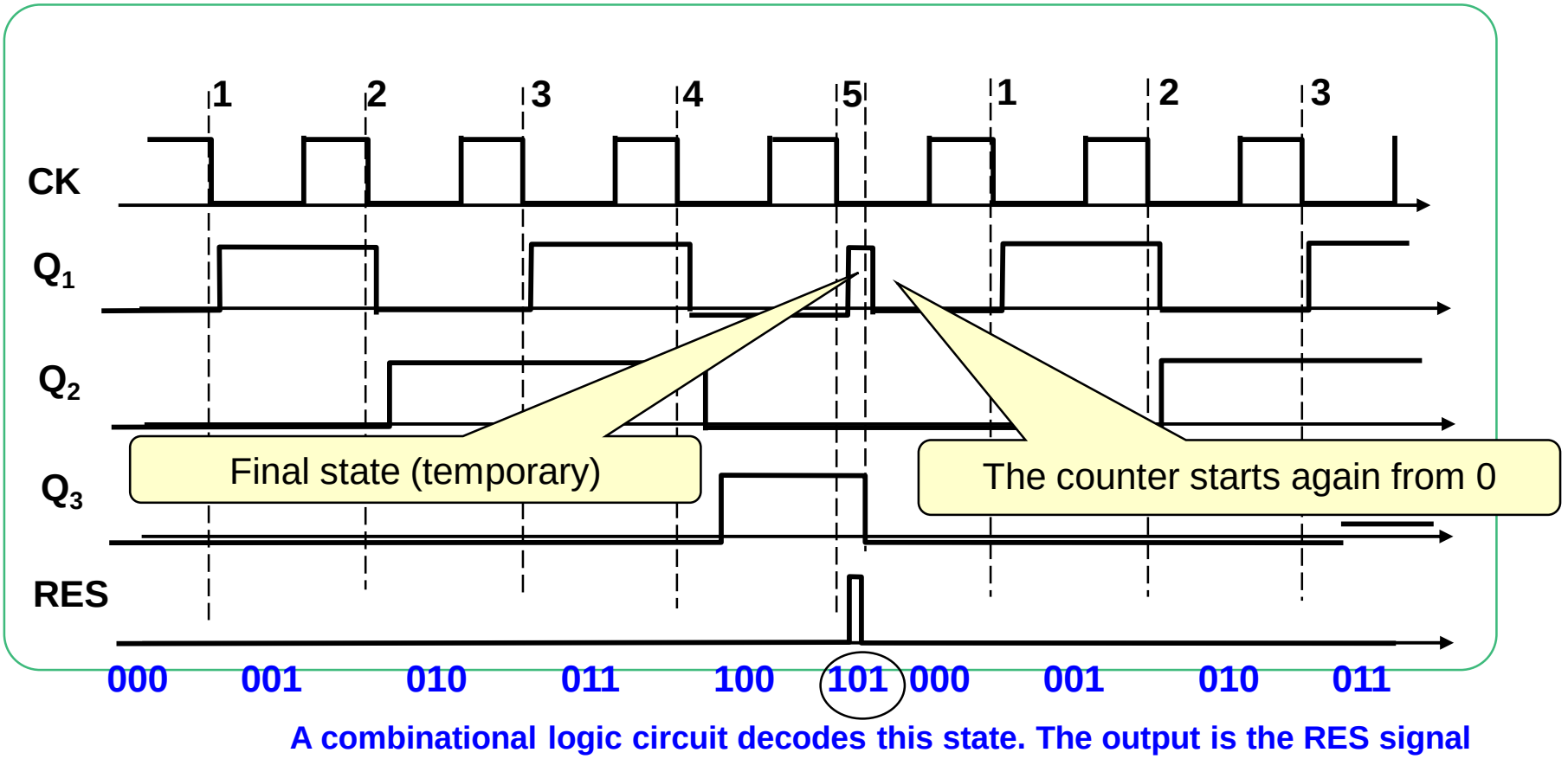
- They solve the propagation delay in ripple counters because all flip-flops are connected to the same clock signal
- Thus, each stage changes state at the same time
- Additional circuitry is used to determine which stages must change state on each clock pulse
- Faster than ripple counters, but more complex
- Available in many forms including up, down, up/down, and modulus-N counters

Sync/Async Comparison



More General: Any Module Counter

- Decodes the module (i.e., 5, 7, ...)
  - Issues reset RES to all the FFs (restart from 000…)

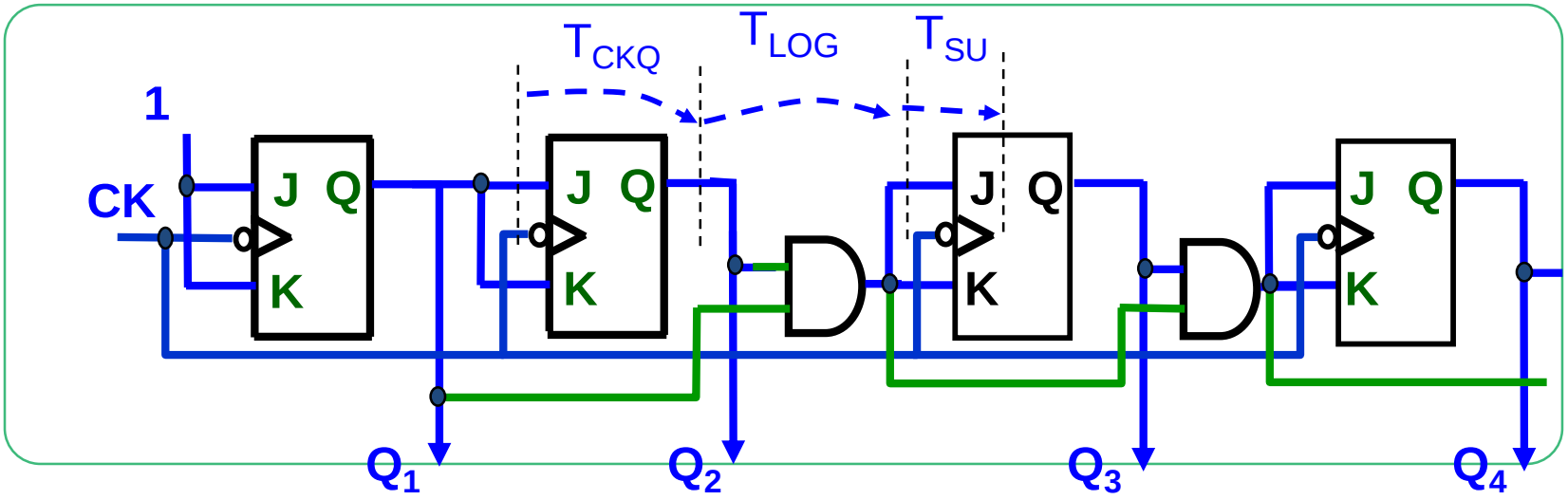


Timing Issues

- The signal through combinational logic circuits is delayed by the propagation time: T_P
- For each FF, we must satisfy timing requirements
  - Set up time
  - Hold time
- The signal through each FF is delayed by the propagation time $T_{CK \rightarrow Q}$

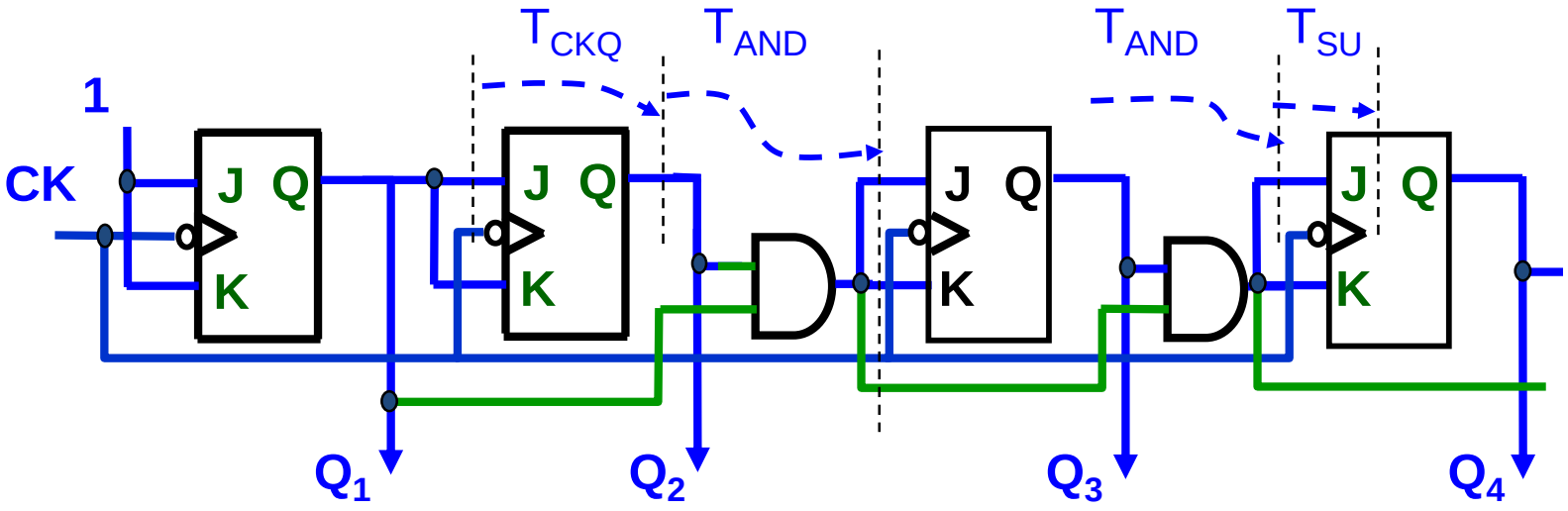
Maximum Operating Frequency (A)

- The maximum operating frequency depends on
  - FF delay CK → Q: T_{CKQ}
  - Combinational logic delay (AND gate): T_{LOG}
  - Setup time required by the FF: T_{SU}



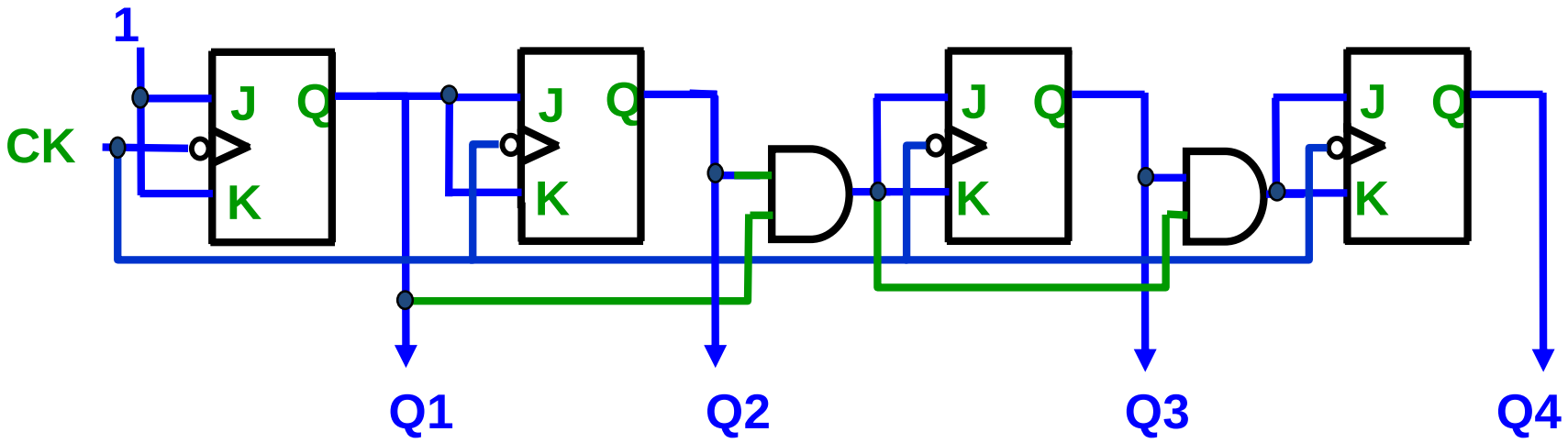
Maximum Operating Frequency (B)

- Longest delay path
  - FF4 combinational input is given by two NAND gates
    - Total delay: TD4 = $T_{CKQ} + T_{AND} + T_{AND} + T_{SU}$
    - Each additional stage adds a delay T_{AND}



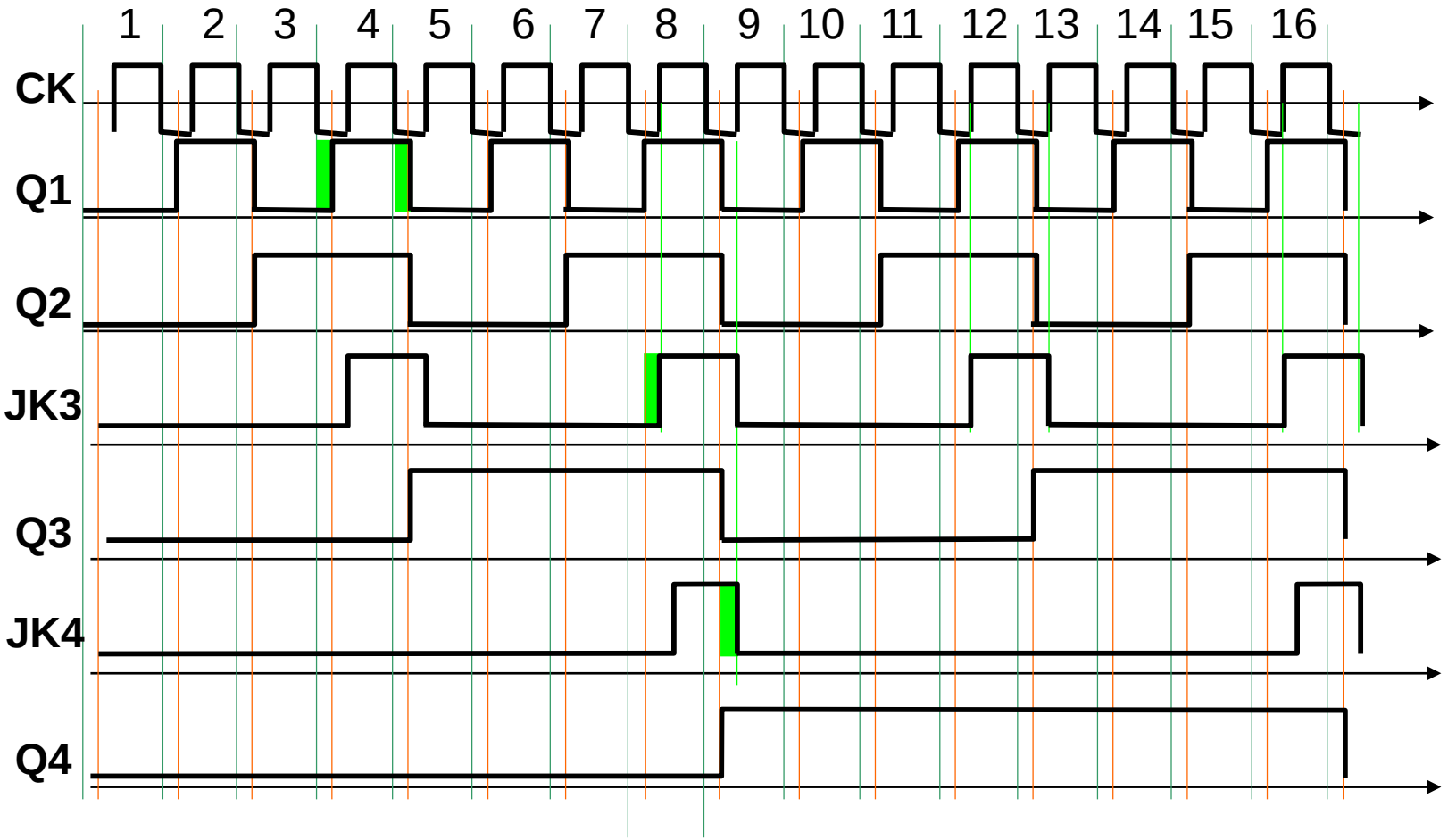
Exercise: Synchronous Counter

Draw the timing diagram of the outputs Qi
(initial condition Qi = 0 at t = 0)



Calculate the F_{MAX} given the following parameters:
T_{AND} = 10 ns; T_{CKQ} = 8 ns; T_{SU} = 3 ns

Synchronous Counter Outputs

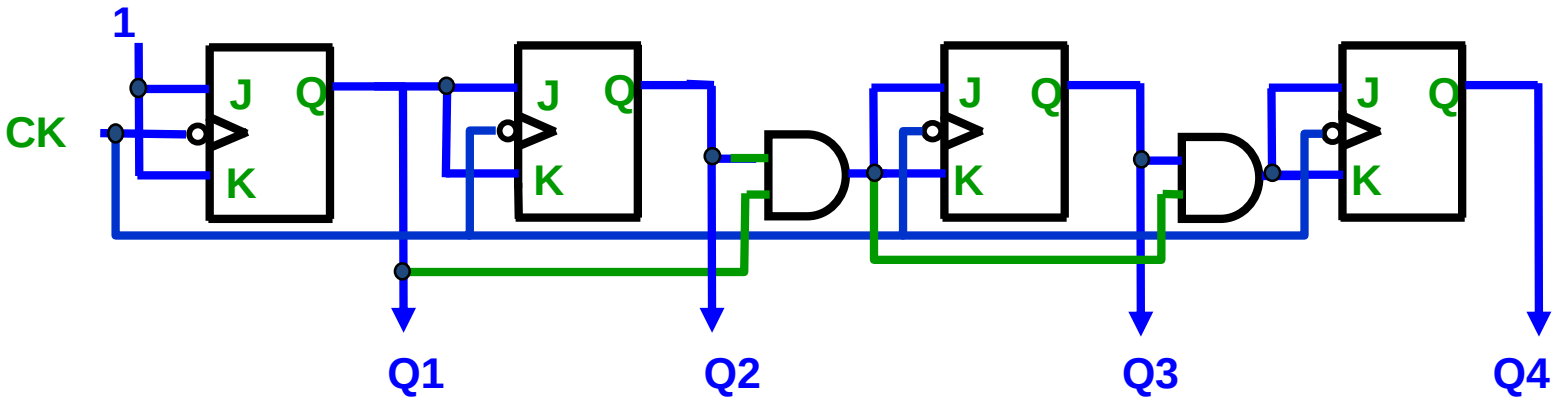


Synchronous Counter: F_{max}

- From 7th to 8th CK period
  - JK4 = (JK3, Q3); must be valid t_{SU} before the 8th clock edge
  - Output JK4 delay from 7th clock edge:

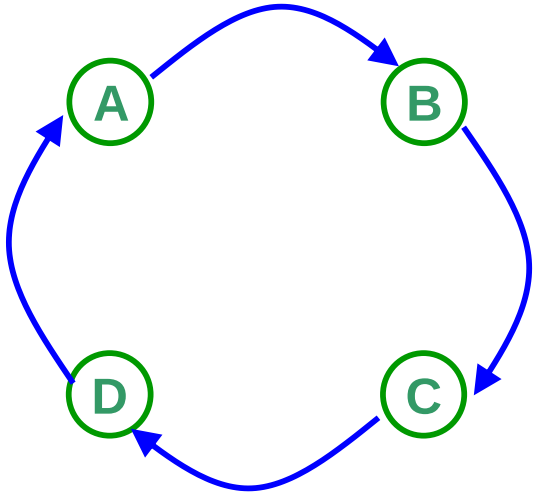
$$t_{CK\_Q} + t_{AND} + t_{AND},$$

$$\text{therefore } T_{CK} > t_{CK\_Q} + 2t_{AND} + t_{SU}$$



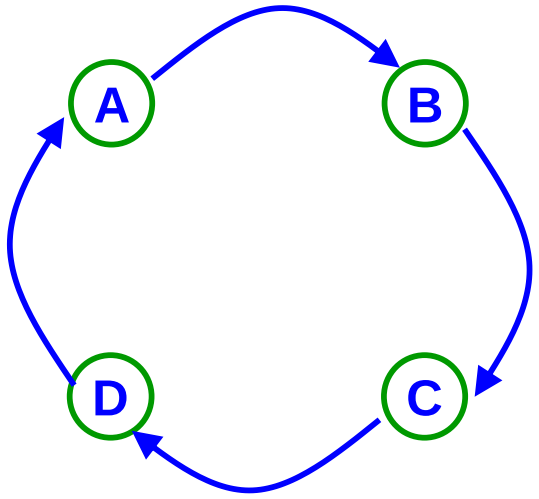
Finite States Machines (FSM)

- The system STATE is defined by the memory elements (FF) state
- State transitions are represented by arcs, conditioned by internal and input variables
- Each internal state has a corresponding OUTPUT state
- This representation is the *State Diagram*



Washing Machine FSM

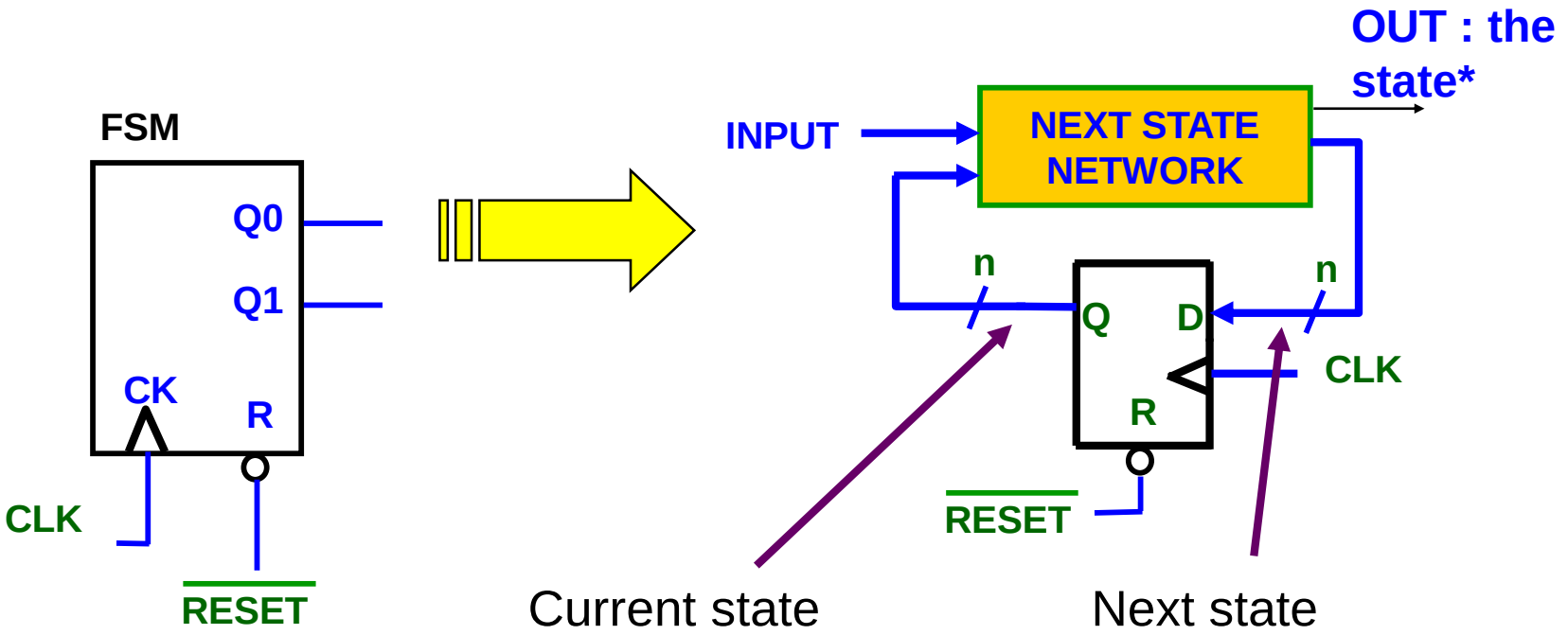
- The control unit for a washing machine is an example of simple sequencer
- Starting state: A; next B, C, D, A, …



STATE	Q_1	Q_0	
A	0	0	LOAD WATER
B	0	1	WASH
C	1	0	SPIN
D	1	1	DRY

Next State Network

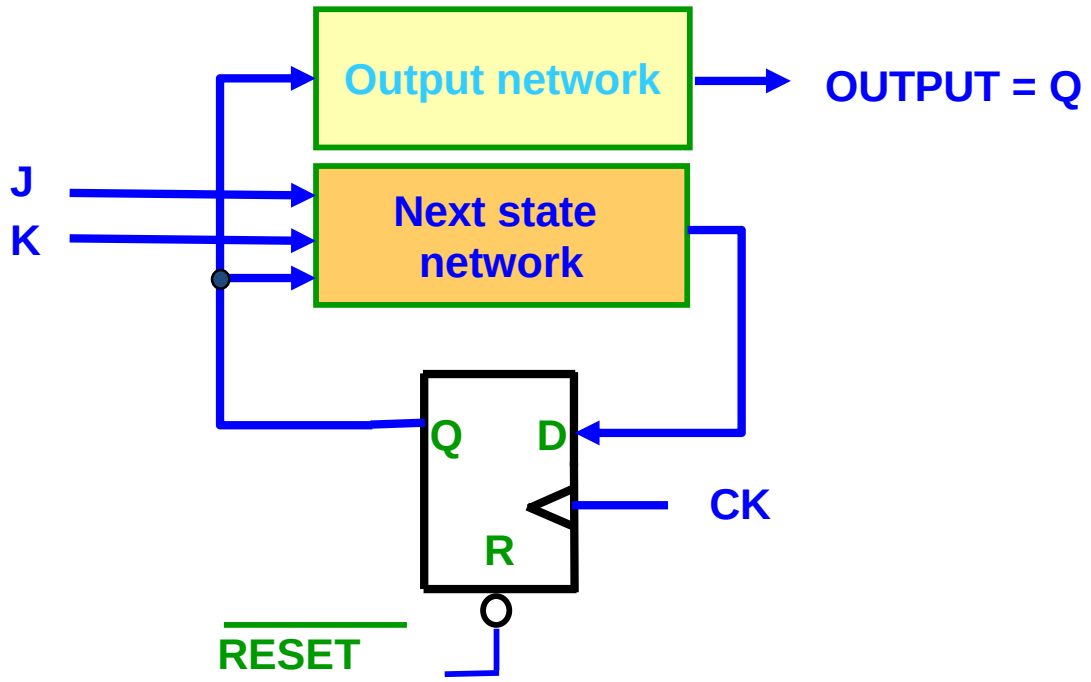
- The FSM advances one step on each clock trigger edge, according to the sequence defined by future state network



*** The new output state is a combinational logic operation between the input and the previous state (Output of the FF, memory element)**

Exercise: Design a Simple FSM

- Design an FSM that acts as a flip-flop JK
- It has two inputs (+ RESET): J, K
- It has two states
  - A (for output Q = H)
  - B (for output Q = L)
- The output network is just a wire connecting the FF to the output Q



State Diagram

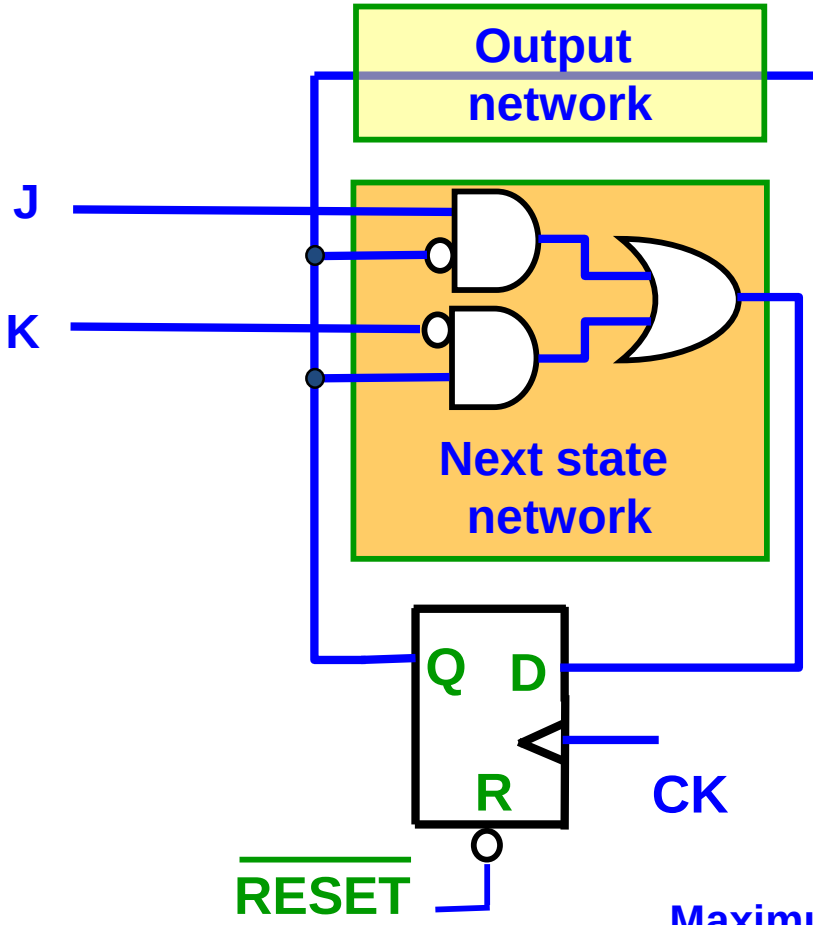
	J K	Current State (Q)	Future State (D)
Memory	L L	L	L
		H	H
K=H sets the output to L	L H	L	L
		H	L
J=H sets the output to H	H L	L	H
		H	H
Toggle	H H	L	H
		H	L

The next step state is determined by the equation:

$$D = Q_{n+1} = \overline{Q}\,J + Q\,\overline{K}$$

Combinational logic sets next state; inputs of the combinational logic circuits are: J, K and Q

FSM for JK FF



$D = Qn+1 = \bar{Q}\,J + Q\,\bar{K}$

$T_{ckmin} > T_{CK\text{-}Q} + Td_{max} + Tsu$

Maximum delay over the combinational logic network

Review Questions

- How many FFs needs a counter modulus 17 (in synchronous and asynchronous implementations)?
- Draw the schematic of an asynchronous divider modulus 32 using D-FFs.
- Draw the schematic (gates and FFs) of a serial-to-parallel converter.
- An FSM has 9 states. How many FFs does it need?
- Do synchronous counters have output delays dependent on the counting modulus?
- Connect two 4-bit SIPO registers to make an 8-bit SIPO.
- Which parameters determine the max clock frequency of an FSM?