CSE 2317 | Week 5 | Class 1 | Saturday · 11:00 AM – 12:20 PM | 80 min

Counters and registers

String flip-flops together and they count, divide, shift and remember words. Ripple or synchronous: the difference is when the outputs can be trusted. Class Test 1 in the last 20 minutes.

---

01 COLD OPEN 0–3 min

Seven, then… six, four, zero, eight

Last class's homework: two T flip-flops, each clocked by the one before, **count**. Add two more and you have a 4-bit counter.

It sits at **0111 = 7**. The clock edge arrives. What values appear on the four outputs, in order, before it reaches 8?

THINK DEEPER

Each flip-flop takes t_{pd} = 20 ns to respond, and each is clocked by the previous output. List the outputs at 0, 20, 40, 60 and 80 ns.

7 → 6 → 4 → 0 → 8. For 60 ns the counter shows values that are not the count. A display flickers; a decoder looking for "0" fires a false pulse. Today: why it happens and how the synchronous counter fixes it.

A STRONG ANSWER MENTIONS

- Only FF0 sees the clock. FF1 waits for Q0 to fall, FF2 for Q1, FF3 for Q2.
- Three of the bits fall one after another; then the top bit rises.

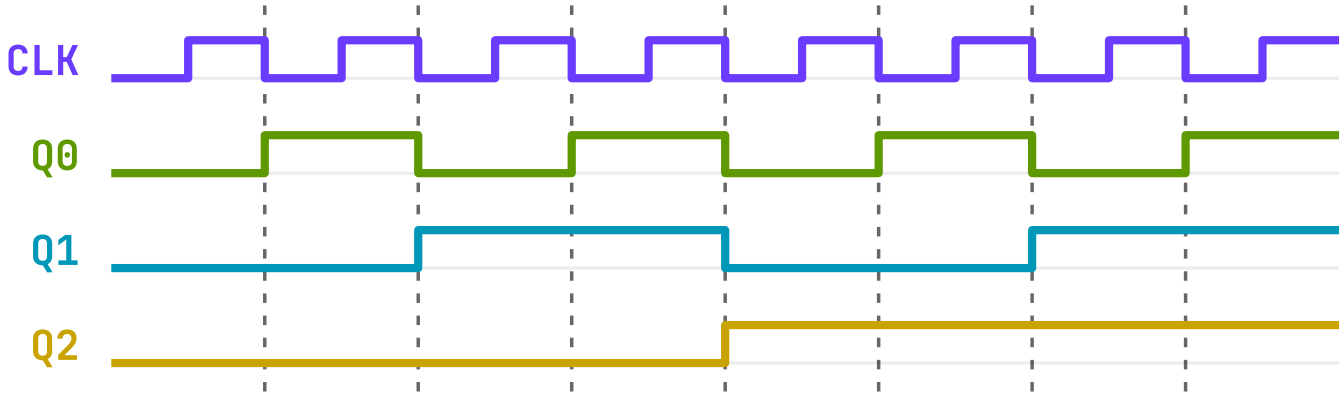
---

02 CONCEPT 3–7 min

The ripple counter: each flip-flop clocks the next

- n T flip-flops with T = 1 (or JK with J = K = 1). The clock drives only FF0; every Q_i is the clock of FF_{i+1}.
- **Negative-edge** flip-flops: FF_{i+1} toggles when Q_i falls 1 → 0, exactly when binary counting carries: an **up**-counter. Clock the next stage from $\bar{Q}_i$ (or use positive-edge flip-flops) and it counts **down**.
- Every stage halves the frequency: a free **frequency divider**.

$$N = 2^n \text{ states}, \qquad f_{Q_{n-1}} = \frac{f_{clk}}{2^n}, \qquad t_{settle} = n\,t_{pd}$$



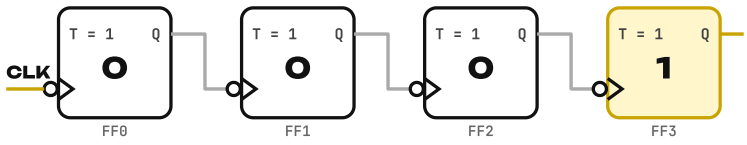
Ideal (zero-delay) waveforms: Q2Q1Q0 counts 0, 1, 2 … 7. The real ones are shifted by one t_{pd} more at each stage.

03 LIVE LAB 7–12 min

Live lab: watch the carry ripple

Counters: ripple vs synchronous Negative-edge T flip-flops, one flip-flop delay = one small step. Watch the outputs just after each falling clock edge: a ripple counter passes through values that are not the count.

auto clock



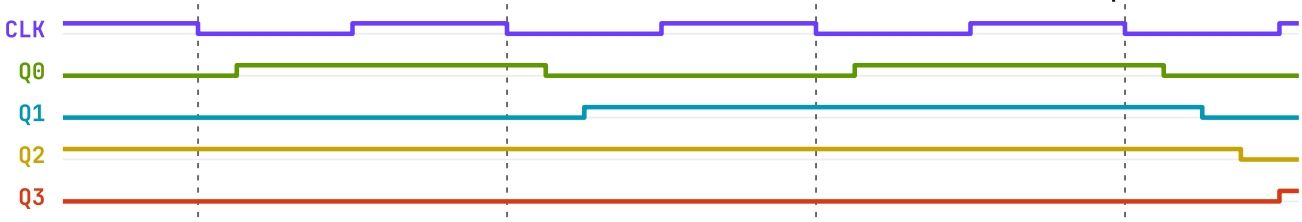
1000 **8** $Q_3Q_2Q_1Q_0$

7 → 6 → 4 → 0 → 8

SETTLES AFTER
up to 4 t_{pd}

FALSE STATES SEEN
4

Clock half-period: **4 t_{pd}**



04 WORKED EXAMPLE 12–17 min

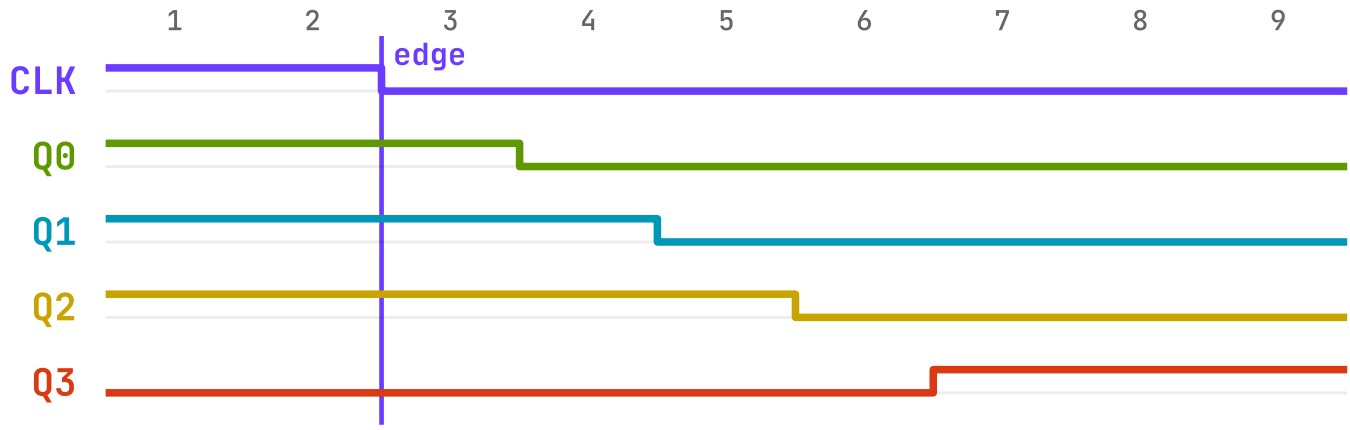
Worked example: ripple delay and f_{max}

GIVEN

A 4-bit ripple up-counter of negative-edge T flip-flops, t_{pd} = 20 ns each. It holds 0111. The count is read by a circuit that needs it stable for a strobe time t_s = 30 ns before the next edge.

FIND

(a) The output values and when they occur after the clock edge. (b) The maximum clock frequency.



One slot = 20 ns. The clock falls at the marker.

Method: ripple counter delay (Mano; Taub & Schilling)

SOLUTION

1. The clock falls at t = 0. FF0 toggles one delay later: Q0 1 → 0 at **20 ns**. The outputs now read 0110 = **6**.
2. Q0 fell, so it clocks FF1: Q1 1 → 0 at **40 ns** (0100 = 4). Q1 fell, so FF2: Q2 1 → 0 at **60 ns** (0000 = 0).
3. Q2 fell, so FF3: Q3 0 → 1 at **80 ns**. Only now do the outputs show the right count, 1000 = 8.

$$7 \to \underbrace{6 \to 4 \to 0}_{\text{false states}} \to 8$$

4. Worst case, the last flip-flop settles n·t_{pd} after the edge, and the count must then be read (strobed) before the next edge:

$$T_{clk} \geq n\,t_{pd} + t_s = 4(20) + 30 = 110 \text{ ns} \;\Rightarrow\; f_{max} = 9.1 \text{ MHz}$$

> The flip-flops alone could toggle much faster. The limit comes from waiting for the ripple, and it grows with every stage you add.

ANSWER

(a) 0 ns: 7, 20 ns: 6, 40 ns: 4, 60 ns: 0, 80 ns: 8. (b) T ≥ 110 ns, f_{max} ≈ 9.1 MHz.

05 CONCEPT 17–20 min

Mod-N: cut the count short

- Choose the fewest flip-flops with $2^n \geq N$. Detect the state **N** with a NAND of the Q's that are 1 in N, and feed it to every flip-flop's asynchronous **CLR̄**.
- **Decade (mod-10)**: N = 10 = 1010, so NAND(Q3, Q1). The counter reaches 1010, is cleared to 0000 within a few nanoseconds, and counts 0 … 9.
- Price: state 1010 still appears, briefly, as a **glitch**. The 74LS90 decade counter is built this way.

0 0000	1 0001	2 0010	3 0011
4 0100	5 0101	6 0110	7 0111
8 1000	9 1001	10 1010	

CLR̄ → 0000

FROM OUR LAB

In the counter lab (slide 3), pick **Mod-10** and watch 9 → 8 → 10 → 0: the 10 lasts one gate delay.

06 YOUR TURN 20–24 min

Your turn · 1: a mod-6 ripple counter

1 EXAM CQ

Design a **mod-6** ripple up-counter from negative-edge T flip-flops with active-low CLR̄.

(a) How many flip-flops? (b) Which gate drives CLR̄? (c) The clock is 60 kHz: what are the frequency and duty cycle of the top output?

SOLUTION

1. $2^2 = 4 < 6 \leq 2^3 = 8$: **3 flip-flops**, 2 unused states (6, 7).
2. 6 = 110: clear when Q2 = Q1 = 1:

$$\overline{CLR} = \overline{Q_2 Q_1}$$

3. Sequence 0, 1, 2, 3, 4, 5, (6 briefly), 0. Q2 is 1 only in states 4 and 5: high 2 periods out of 6.

$$f_{Q_2} = \frac{60\ \text{kHz}}{6} = 10\ \text{kHz}, \qquad \text{duty} = \tfrac{2}{6} = 33\,\%$$

ANSWER

3 flip-flops, CLR̄ = NAND(Q2, Q1). Q2 is a 10 kHz square wave with 33 % duty cycle (not 50 %).

07 CONCEPT 24–28 min

Synchronous counters: one clock for everyone

- All flip-flops share the clock, so all change **together**, one t_{pcq} after the edge. No ripple, no false states.
- Binary up-counter: a bit toggles when every bit below it is 1:

$$T_0 = 1, \quad T_1 = Q_0, \quad T_2 = Q_0 Q_1, \quad T_3 = Q_0 Q_1 Q_2 \qquad T_{clk} \geq t_{pcq} + t_{AND} + t_{su}$$

The minimum period no longer grows with n (with parallel AND gates). Any sequence at all can be designed with the excitation table:

Q → Q⁺	J K	D	T
0 → 0	0 x	0	0
0 → 1	1 x	1	1
1 → 0	x 1	0	1
1 → 1	x 0	1	0

PRO TIP

Excitation = the characteristic table read backwards: "I need this transition; what inputs give it?" JK has the most x's, so the simplest gates.

08 WORKED EXAMPLE 28–35 min

Worked example: a synchronous mod-5 counter from JK flip-flops

GIVEN

JK flip-flops, common clock. Count 0, 1, 2, 3, 4, 0, ...

FIND

J and K for each flip-flop, and what happens if it powers up in an unused state.

$Q_2Q_1Q_0$	Q^+	J_2K_2	J_1K_1	J_0K_0
000	001	0x	0x	1x
001	010	0x	1x	x1
010	011	0x	x0	1x
011	100	1x	x1	x1
100	000	x1	0x	0x

$Q_2Q_1Q_0$	Q^+	J_2K_2	J_1K_1	J_0K_0
101–111	x	xx	xx	xx

Method: counter design with the JK excitation table (Mano)

SOLUTION

1. Sequence 000 → 001 → 010 → 011 → 100 → 000. States 101, 110, 111 never occur: their inputs are don't-cares (x).
2. For each flip-flop and each present state, read the transition Q → Q⁺ and look up the JK excitation table (0→0: 0x, 0→1: 1x, 1→0: x1, 1→1: x0).
3. Simplify each column with a K-map (the x's help):

$$J_0 = \bar{Q}_2,\ K_0 = 1 \qquad J_1 = K_1 = Q_0 \qquad J_2 = Q_1Q_0,\ K_2 = 1$$

4. Check the unused states: 101 → 010, 110 → 010, 111 → 000. All lead back into the count: the counter is **self-starting**.

> Always check: a counter that powers up in an unused state and stays there is a classic exam trap and a real field failure.

ANSWER

$J_0 = \bar{Q}_2$, $K_0 = 1$; $J_1 = K_1 = Q_0$; $J_2 = Q_1Q_0$, $K_2 = 1$. Self-starting: 5, 6 → 2 and 7 → 0.

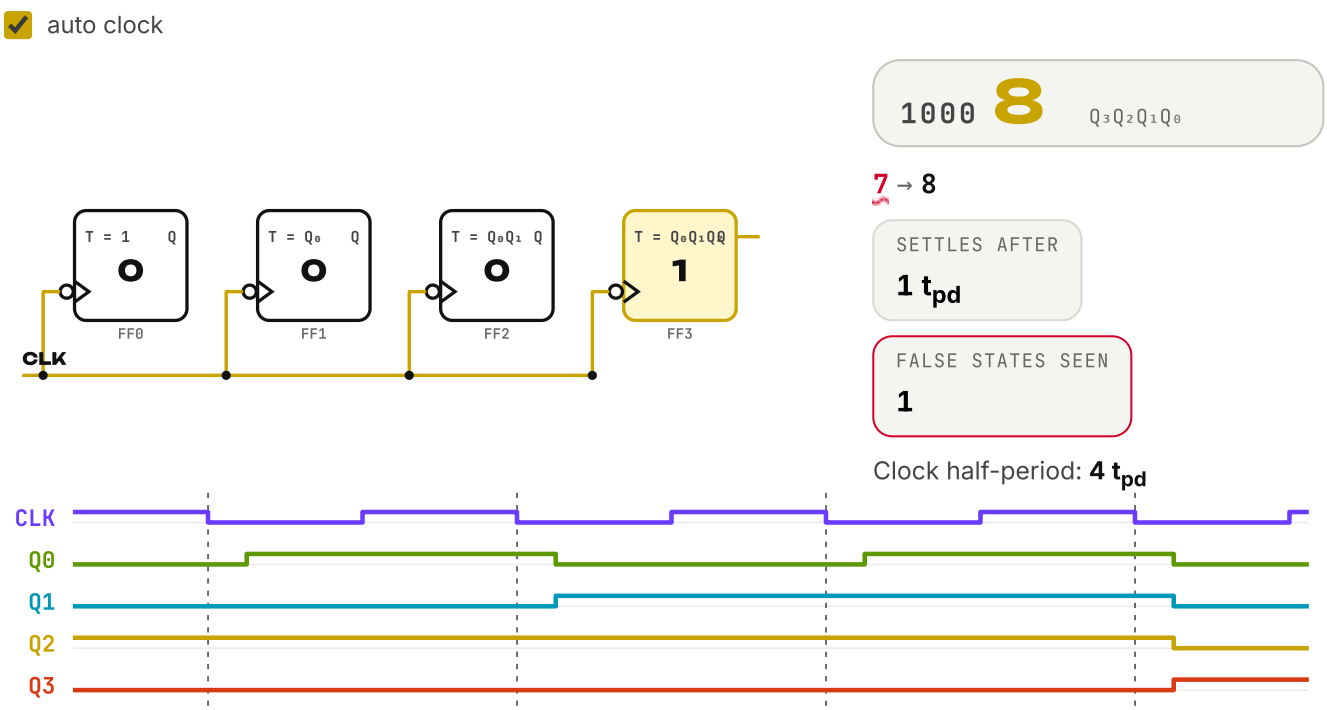
---

09 LIVE LAB 35–38 min

Live lab: the same

count, synchronously

Counters: ripple vs synchronous Negative-edge T flip-flops, one flip-flop delay = one small step. Watch the outputs just after each falling clock edge: a ripple counter passes through values that are not the count.



10 CONCEPT 38–42 min

Registers: words that move

- A **register** is n D flip-flops on one clock: it stores an n-bit word. Connect each Q to the next D and it **shifts** one place per edge.
- Four ways in and out: **SISO** (delay line), **SIPO** (serial → parallel: receive a UART byte), **PISO** (parallel → serial: send it), **PIPO** (plain storage).
- Close the loop and it counts: **ring counter** (Q_{n-1} → D_0, one circulating 1, n states) and **Johnson counter** ($\bar{Q}_{n-1}$ → D_0, 2n states).

COUNTER (N FLIP-FLOPS)	STATES	DECODE
Binary	2^n	n-input AND
Ring	n	none (one-hot)
Johnson	2n	2-input AND

! GOTCHA

Ring and Johnson counters have **unused states** too (16 − 4 = 12 for a 4-bit ring). Power up in one and it never recovers unless you design it to.

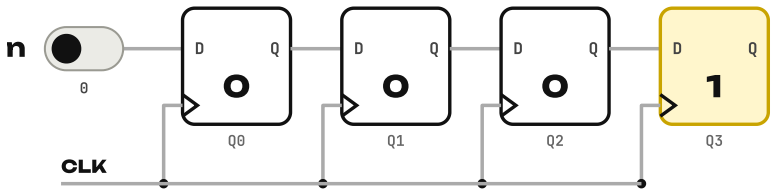
11 LIVE LAB 42–46 min

Live lab: shift, ring, twist

Shift registers, ring and Johnson counters

At every rising edge each flip-flop copies its left neighbour. Feed the last output back to the first and the register counts. Click a flip-flop to corrupt its bit.

auto clock

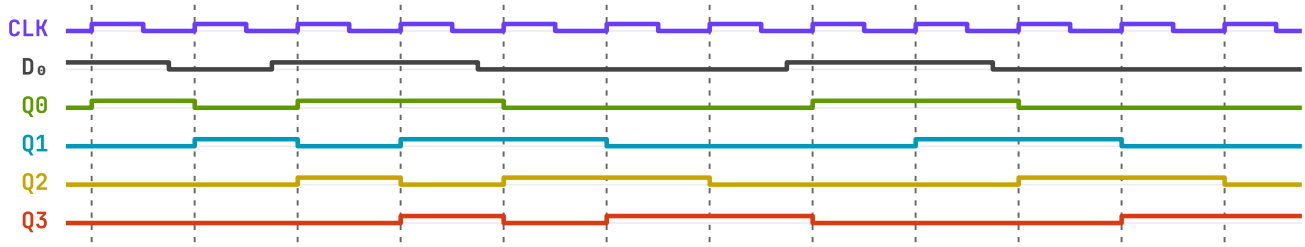


$Q_0Q_1Q_2Q_3$

0001

Recent states, newest first:

0001 0011 0110 1100 1000 0001 0011 0110



12 YOUR TURN 46–50 min

Your turn · 2: the Johnson counter

2 EXAM CQ

A 4-bit shift register, $\bar{Q}_3$ fed back to D_0, starts at 0000 and is clocked at 8 MHz.

(a) List the states until it repeats. (b) Frequency and duty of each output? (c) Show that each state can be decoded with a **2-input** gate.

SOLUTION

1. States ($Q_0Q_1Q_2Q_3$):

$$0000 \to 1000 \to 1100 \to 1110 \to 1111 \to 0111 \to 0011 \to 0001 \to 0000$$

2. **8 states** = 2n. Each output is high for 4 of the 8 clocks: f = 8 MHz / 8 = **1 MHz, 50 % duty**, each delayed one clock from the previous.

3. Each state is the only one with a particular adjacent pair, e.g. state 1100 is the only one with Q_1 = 1 and Q_2 = 0:

$$S_{1100} = Q_1\bar{Q}_2, \qquad S_{0000} = \bar{Q}_0\bar{Q}_3, \qquad S_{1111} = Q_0Q_3$$

ANSWER

8 states; every output is 1 MHz, 50 % duty, shifted by 1/8 period; each state is decoded by one 2-input AND gate.

13 REAL WORLD 50–53 min

Real world: clocks, signs and the prayer-time display

The digital clock above the mosque door or on the wall of a Dhaka pharmacy is counters all the way down: a 32.768 kHz crystal, a **15-stage ripple divider** (2^{15} = 32,768) giving exactly 1 Hz, then mod-10 and mod-6 counters for seconds and minutes, mod-12 or mod-24 for hours.

The scrolling LED sign over a shop is shift registers: a microcontroller pushes each column serially into a chain of 74HC595s, then latches all outputs at once.

BME CORNER

A pulse oximeter or ECG monitor computes heart rate by counting clock ticks between beats: a synchronous counter, read on a known edge. A ripple counter read mid-change could report 0 ms or 1000 ms for one beat.

> Because 32,768 = 2^{15}: plain divide-by-2 stages reach 1 Hz with no decoding logic, and a small, low-power tuning-fork crystal works well at that frequency. Every watch uses it.

---

14 POP QUIZ 53–57 min

Pop quiz

1. **A 5-bit ripple counter is clocked at 1 MHz. The frequency at the last output (Q4) is…**

   A. 200 kHz
   B. 62.5 kHz
   C. 31.25 kHz ✓
   D. 5 MHz

   Each stage divides by 2: 1 MHz / 2^5 = 31.25 kHz.

2. **Why can a ripple counter show false states?**

   A. the flip-flops are level-triggered
   B. its flip-flops change one after another, each one delay later ✓
   C. it has no clear input
   D. it uses T flip-flops

   Each flip-flop is clocked by the previous one, so a change like 0111 → 1000 ripples through 0110, 0100 and 0000 first.

3. **The fewest flip-flops for a mod-12 counter is…**

   A. 3
   B. 4 ✓
   C. 6
   D. 12

   Smallest n with $2^n \geq 12$: n = 4 (16 states, 4 unused).

4. **A 4-bit Johnson counter has how many states?**

   A. 4
   B. 8 ✓
   C. 15
   D. 16

   A twisted ring of n flip-flops cycles through 2n states: 0000, 1000, 1100, 1110, 1111, 0111, 0011, 0001.

5. **An 8-bit serial-in, parallel-out register (74HC595) is loaded from a microcontroller. How many clock pulses does a new byte need?**

   A. 1
   B. 4
   C. 8 ✓
   D. 16

   One bit enters per clock edge; after 8 edges all 8 bits are in place (then a separate latch clock shows them at once).

15 RECAP 57–60 min

Three things to remember

01

Ripple: each flip-flop clocks the next. Simple, divides by 2^n, but settles $n \cdot t_{pd}$ after the edge and shows false states.

02

Synchronous: one clock, design with the excitation table (JK: 0x, 1x, x1, x0), check unused states. Mod-N ripple: NAND the 1-bits of N into $CL\bar{R}$.

03

Registers store and shift words (SISO, SIPO, PISO, PIPO). Ring: n states, Johnson: 2n states, decoded by 2-input gates.

PRO TIP

Homework: design a synchronous **mod-6** counter (0 … 5) with D flip-flops and check its unused states. Next class: memories and PLAs.

16 RECAP 60–80 min

Class Test 1

Class Test 1 · 20 minutes

COVERS WEEKS 1–4

Diode gates (DRL), RTL and DTL · the BJT switch and MOS gates · TTL, ECL, I^2L and CMOS · latches and flip-flops (timing diagrams, characteristic equations, setup and hold).

RULES

Closed book, answer on the question paper. Calculators allowed, phones away. Show the working for the calculation questions.

Questions are handed out on paper. Today's counters are **not** in this test.

S. M. Sakeef Sani
Adjunct Lecturer, CSE · Northern University Bangladesh
smsakeefsani.hbitcorp.com