Penn State Abington CMPEN 271 Lecture Set #17

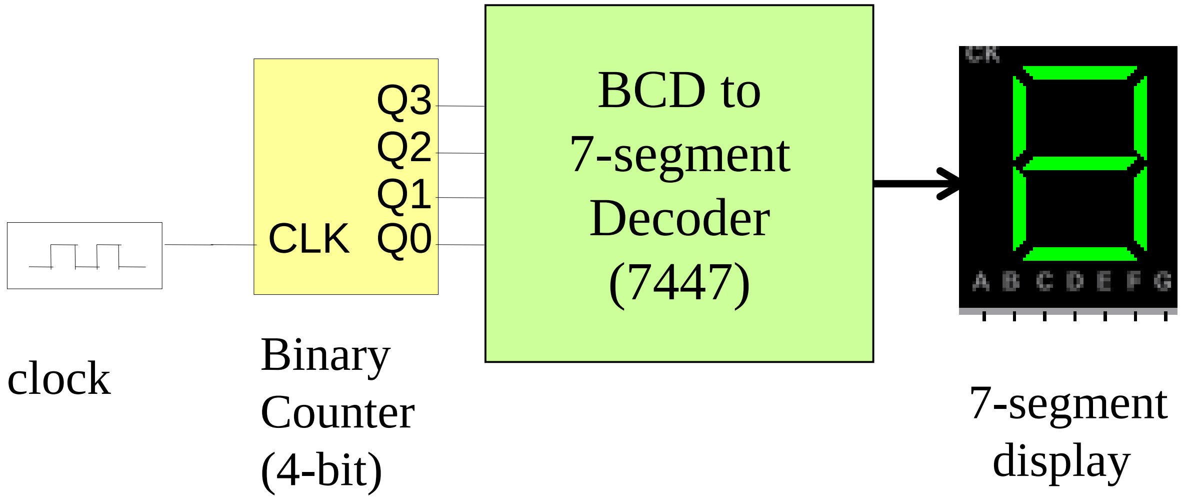
JK Flip Flops, Ripple Counters

R. Avanzato © 2017-2018

<u>Topics:</u>

• JK flip-flops	**Part 1 of 4 🡸**
• JK FF applications - "ripple" counter	Part 2 of 4
• Timing Diagrams	
• Asynchronous FF inputs	
• BCD counter	
• HW #9 A,B (counters)	Part 3 of 4
• Review Questions	Part 4 of 4

Problem: Design a Binary Counter



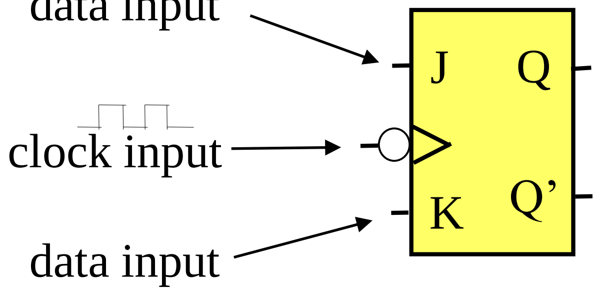
<u>Questions:</u>

1. Describe the operation of the above circuit.
2. What is the count sequence of the counter?
3. What values are displayed on the 7-segment display?
4. How would you design the binary counter?
5. How would you design a counter with sequence 0 to 9?
6. What is difference between **<u>binary counter</u>** and **<u>BCD counter?</u>**

Goals for this Lecture

- We know how positive and negative-edge triggered **D flip-flops** work.
- D flip flops were very useful to build **shift register and ring counters**. We used ring counters to control motors and to control LEDs.
- It is **not obvious** how we can use D flip-flops to build a counter (but our goal is to build a counter today).
- So, we will introduce a new flip-flop, called **the JK flip-flop**, which has special properties that will allow us to quickly and easily make counter circuits.
- **Remember**: a flip-flop stores 1 bit (0 or 1). 2 states.

JK Flip-flop - 1

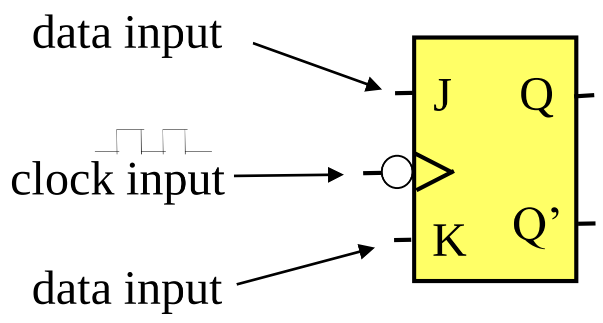


J	K	CLK	Q	Q’	
0	0	↓	Qold	Q’old	
1	0	↓	1	0	(set)
0	1	↓	0	1	(reset, clear)
1	1	↓	Q’	Q	(**toggle**)

JK flip-flops have 2 data inputs – one input is labeled “**J**”, and the other input is labeled “**K**”. It can also been seen from the block diagram above that this particular JK flip-flop is **negative-edge triggered**, which means the flip-flop will observe the J and K inputs and Q will change (or be triggered) only during the negative edge of the clock (when the clock goes from 1 to 0)

- When J = 0 and K = 0, then Q will stay in the same state (during neg. edge of clock)
- When J = 1 and K = 0, then Q will become 1 at the negative edge of the clock.
- When J = 0 and K = 1, then Q will become 0 at the negative edge of the clock.
- When **J = 1 and K = 1**, then Q will **“toggle”** states; that is, if the old Q was 0, then Q will flip to 1, and if the old Q was 1, then Q will flip to 0.
- Q’ is always the complement (opposite) of Q

JK Flip-flop - 2

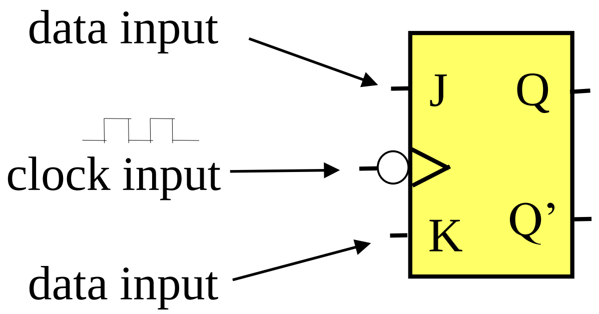


J	K	CLK	Q	Q’	
0	0	↓	Qold	Q’old	
1	0	↓	1	0	(set)
0	1	↓	0	1	(reset, clear)
1	1	↓	Q’	Q	(toggle)

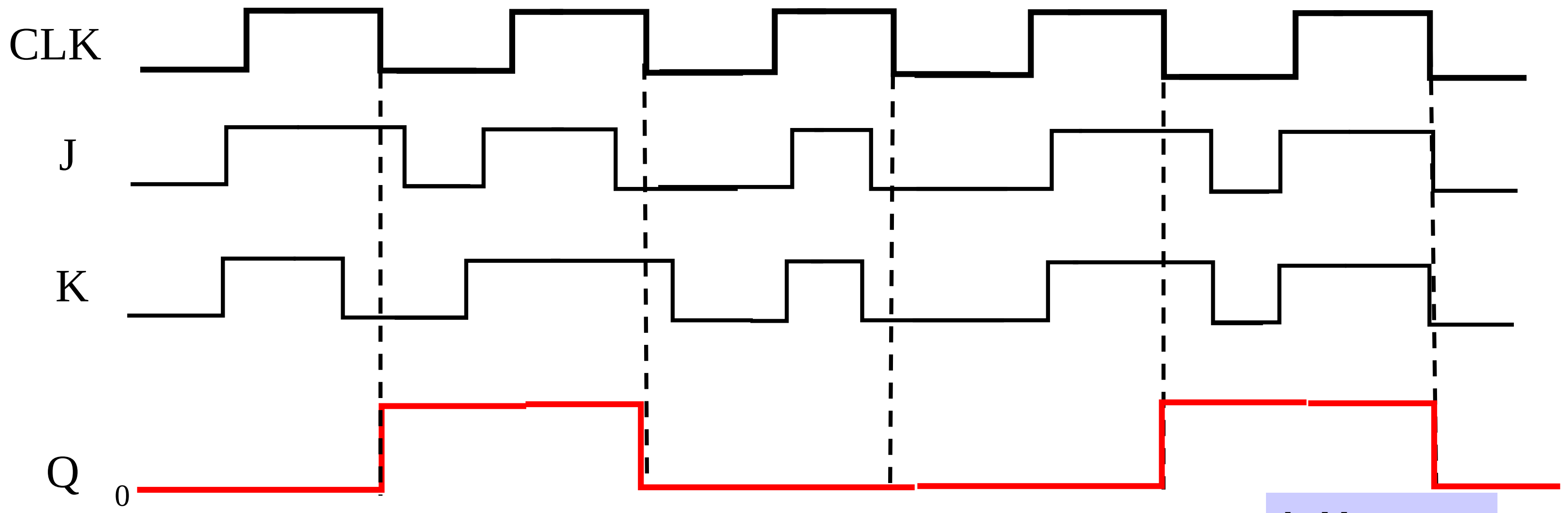
JK flip-flops have the interesting feature of “**toggle**” mode when both J=1 and K=1. This allows for the easy construction of counters. JK flip-flops exist as negative-edge triggered (shown above), positive-edge triggered, and pulse triggered. We will concentrate on the **negative-edge triggered** JK flip flop. Negative-edge triggered JK flip-flops are a common building block for “ripple” binary counters. **Remember: a flip-flop or latch stores one bit.**

Some students find it helpful to connect the symbol “K” with “Klear” (clear). Then J must mean “Set”. (It would have been easier if “J” was labeled “S”, and “K” was labeled “R” – but it wasn’t!))

JK Flip-flop - 3

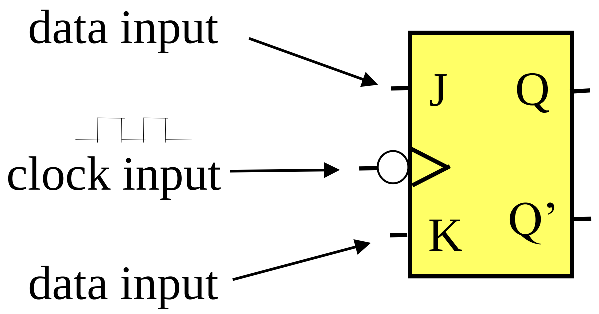


J	K	CLK	Q	Q'	
0	0	↓	Qold	Q'old	
1	0	↓	1	0	(set)
0	1	↓	0	1	(reset, clear)
1	1	↓	Q'	Q	(toggle)

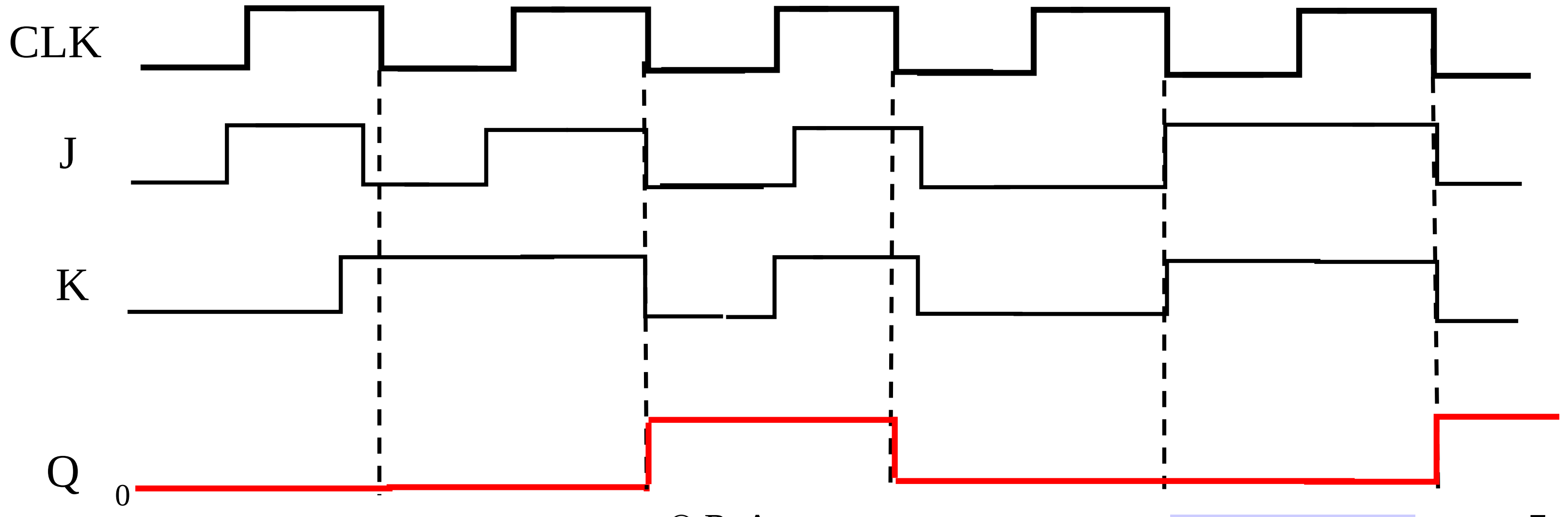


hold time = 0

JK Flip-flop - 4

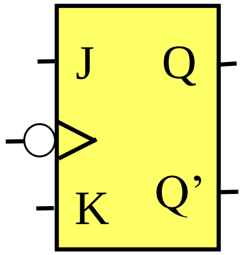


J	K	CLK	Q	Q’	
0	0	↓	Qold	Q’old	
1	0	↓	1	0	(set)
0	1	↓	0	1	(reset, clear)
1	1	↓	Q’	Q	(toggle)



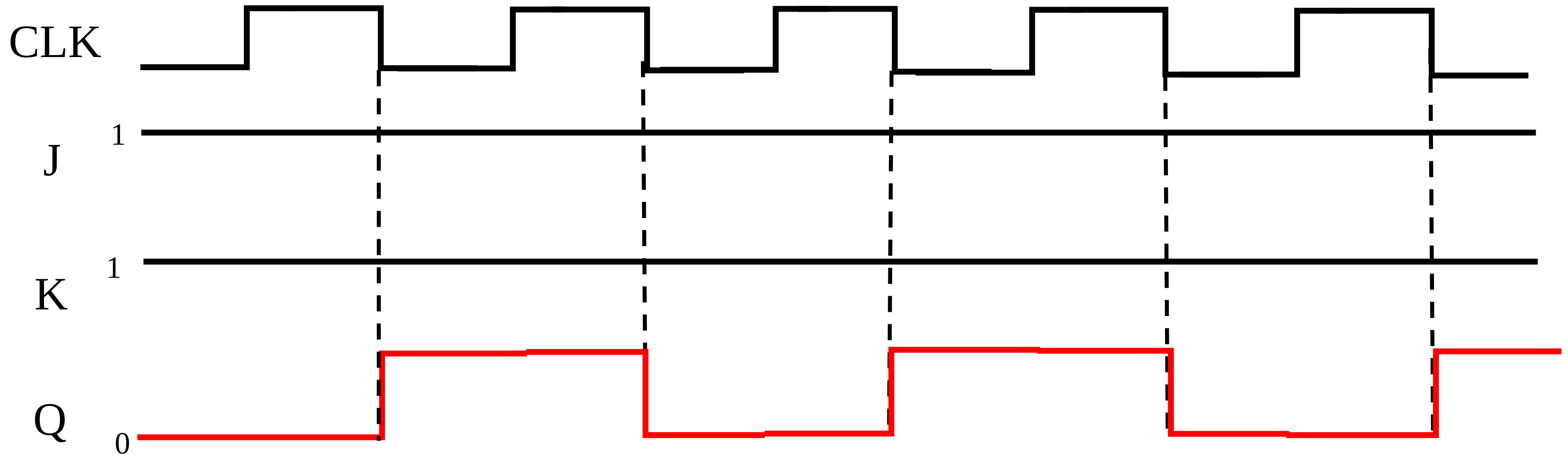
hold time = 0

JK Flip-flop - 5



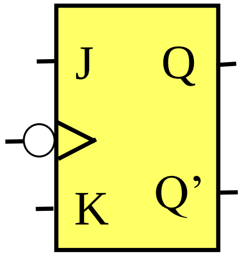
J	K	CLK	Q	Q’	
0	0	↓	Qold	Q’old	
1	0	↓	1	0	(set)
0	1	↓	0	1	(reset, clear)
1	1	↓	Q’	Q	(toggle)

Note: JK flip-flop is in “toggle” mode when J=1 and K=1. This behavior is useful in the design of “ripple” binary counters.

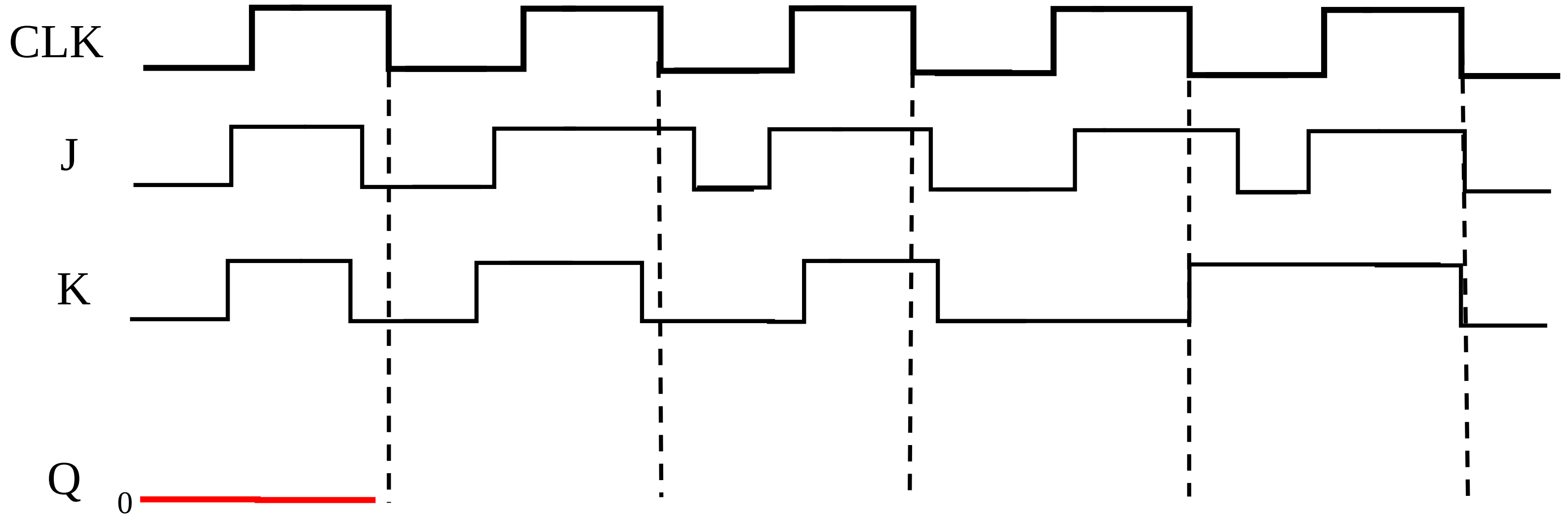


hold time = 0

JK Flip-flop - Exercise



J	K	CLK	Q	Q’	
0	0	↓	Qold	Q’old	
1	0	↓	1	0	(set)
0	1	↓	0	1	(reset, clear)
1	1	↓	Q’	Q	(toggle)



© R. Avanzato

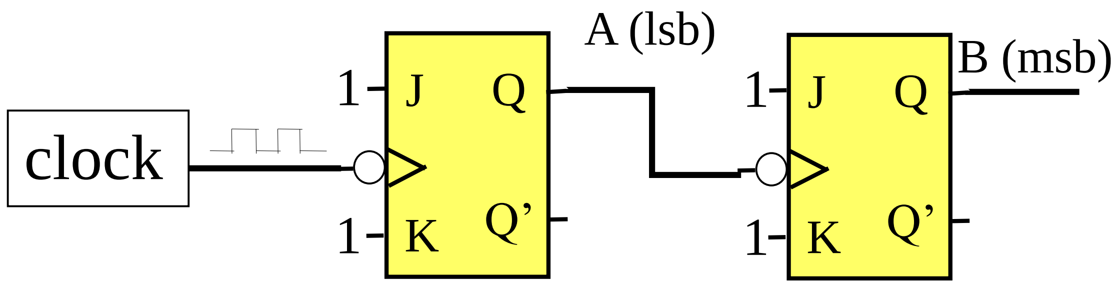
hold time = 0

Penn State Abington
CMPEN 271
Lecture Set #17
JK Flip Flops, Ripple Counters
R. Avanzato ©

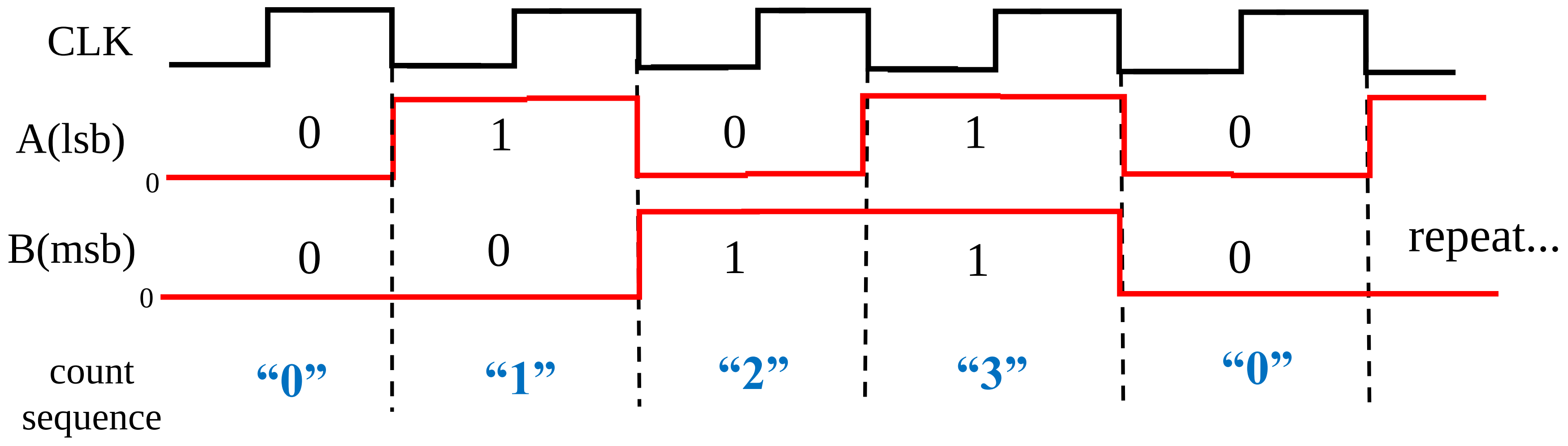
Topics:

• JK flip-flops	Part 1 of 4
• JK FF applications - “ripple” counter	**Part 2 of 4 ←**
• Timing Diagrams	
• BCD counter	
• HW #9 A,B (counters)	Part 3 of 4
• Review Questions	Part 4 of 4

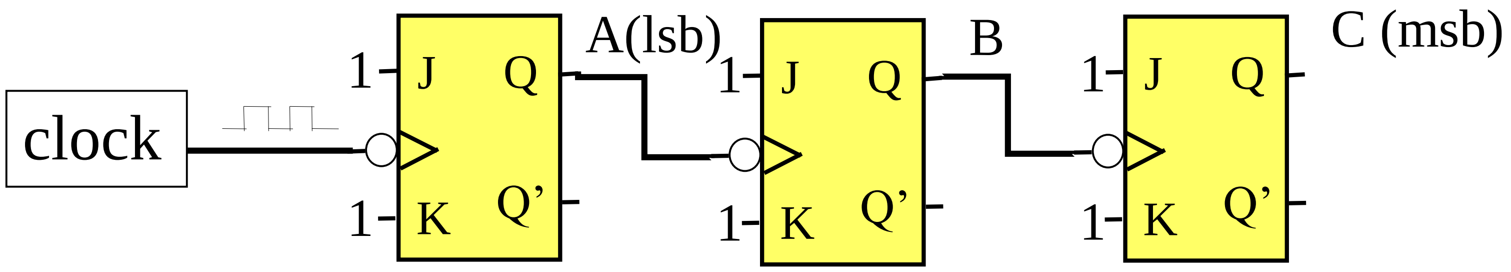
2-bit Ripple Counter



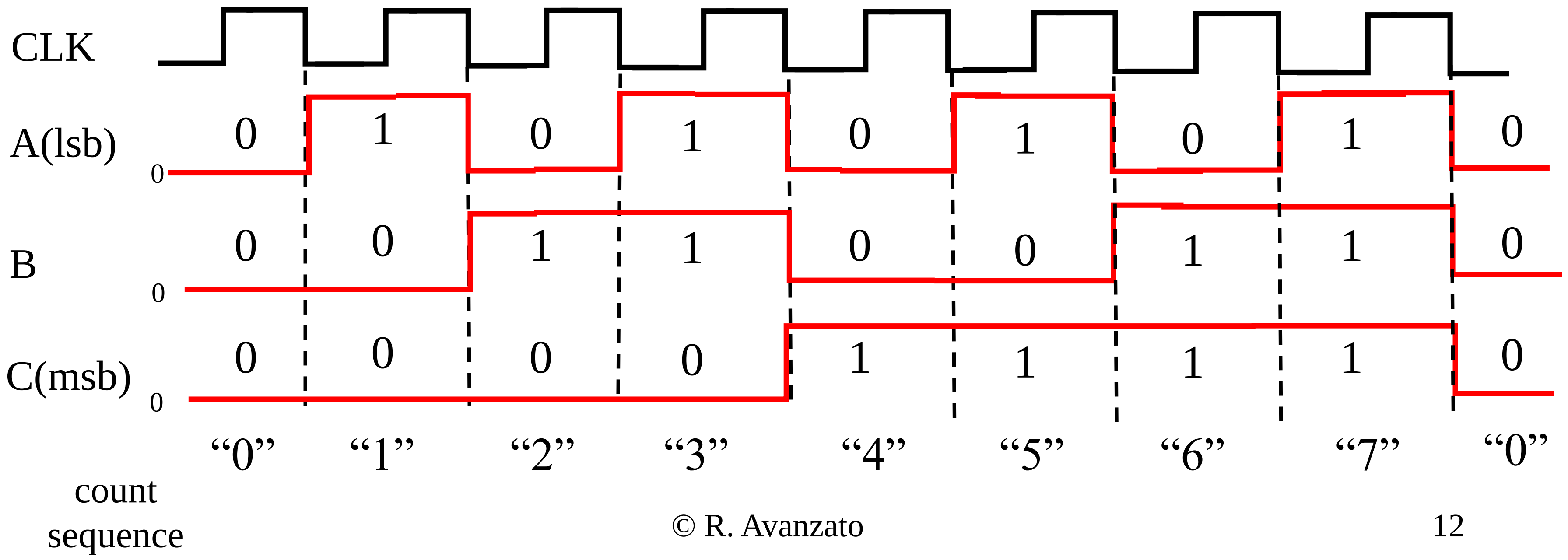
Note: 2-bit ripple counter. Count sequence is 00, 01, 10, 11 and repeat. Also called a MOD- 4 (4 states) counter or a Divide-by-4 counter. Each JK flip-flop is in "toggle" mode (J=1 and K=1). Need a FF for each bit. What is the frequency of signal A? B? (relative to the clock)



3-bit Ripple Counter



Note: 3-bit ripple counter. Count sequence is 000 to 111 and repeat. Also called a MOD- 8 (8 states) counter or a Divide-by-8 counter. Each JK flip-flop is in "toggle" mode (J=1 and K=1). If clock freq = 1KHz what is frequency of A? B? C? What about FF propagation delays? (assume negligible for now).



Ripple Counter Concepts

- **Consider an n-bit ripple counter**
  - Requires n JK flip-flops (each in the toggle mode)
  - Count sequence is 0 to 2^n - 1, and repeats.
  - Divide-by-n counter (consider msb wire)
  - MOD-n counter (n distinct states)

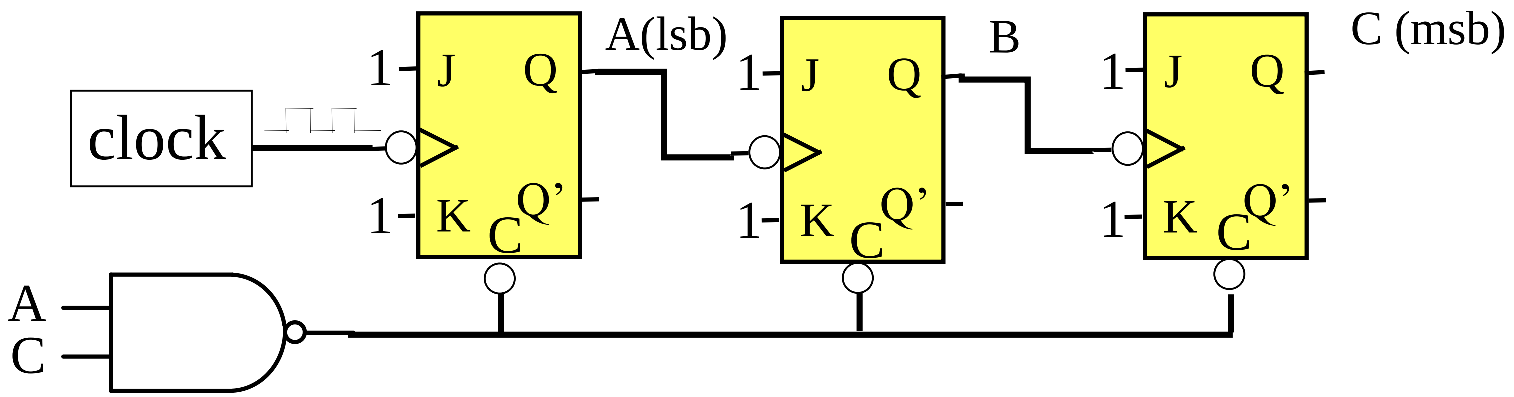
- A **ripple counter** is sometimes called an **"asynchronous"** counter because the FFs do not switch simultaneously. That is, the clocks of the FFs are not all directly connected to the master clock. (Compare to a shift register.) Counters designed as state machines are synchronous counters (more later in the course). The advantage of ripple counters is that they are easy to design and construct, but are not suited for very high-speed applications.

- **How many FFs** do we need to construct a MOD-16 counter? MOD-32?

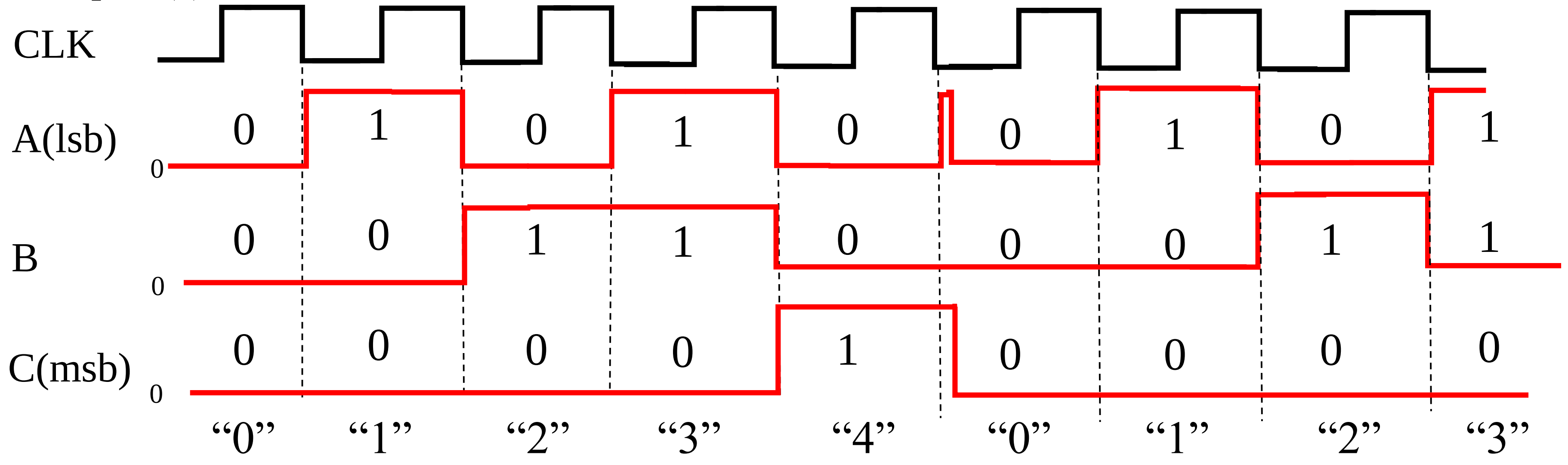
- **How do you design a MOD-10 counter**? MOD-5? MOD-17? Answer: use asynchronous inputs of JK FF.

- Investigate **the impact of propagation delays** when the number of flip-flops increase in a ripple counter.

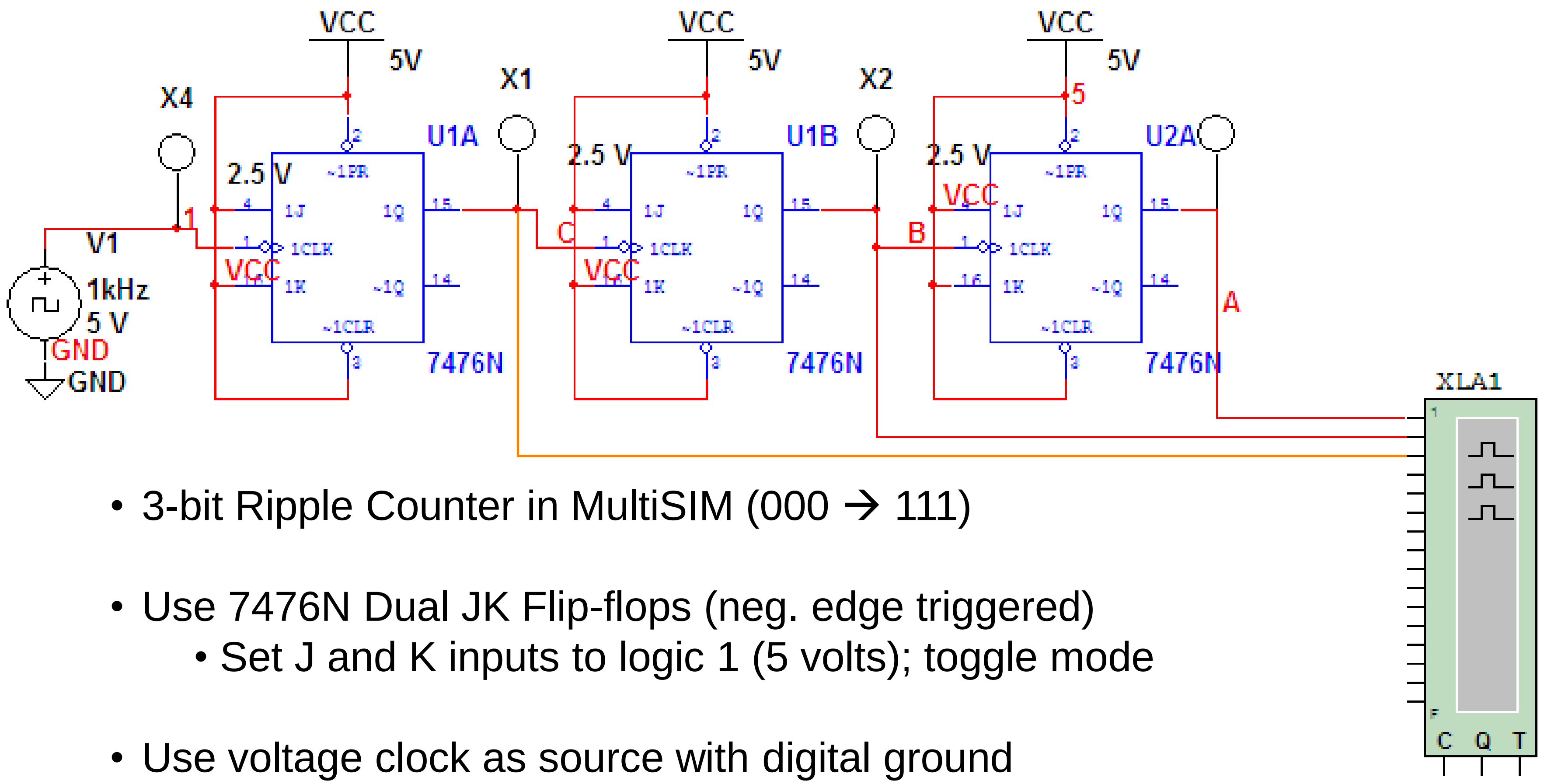
MOD-5 Ripple Counter



Note: MOD-5 ripple counter (0-4). Count sequence is 000 to 100 and repeat. Clear all FFs (set to 0) when count=101. Ignore momentary "glitch(es)" or spike(s) at count=5. Assume count=5 is too short in duration to be considered.

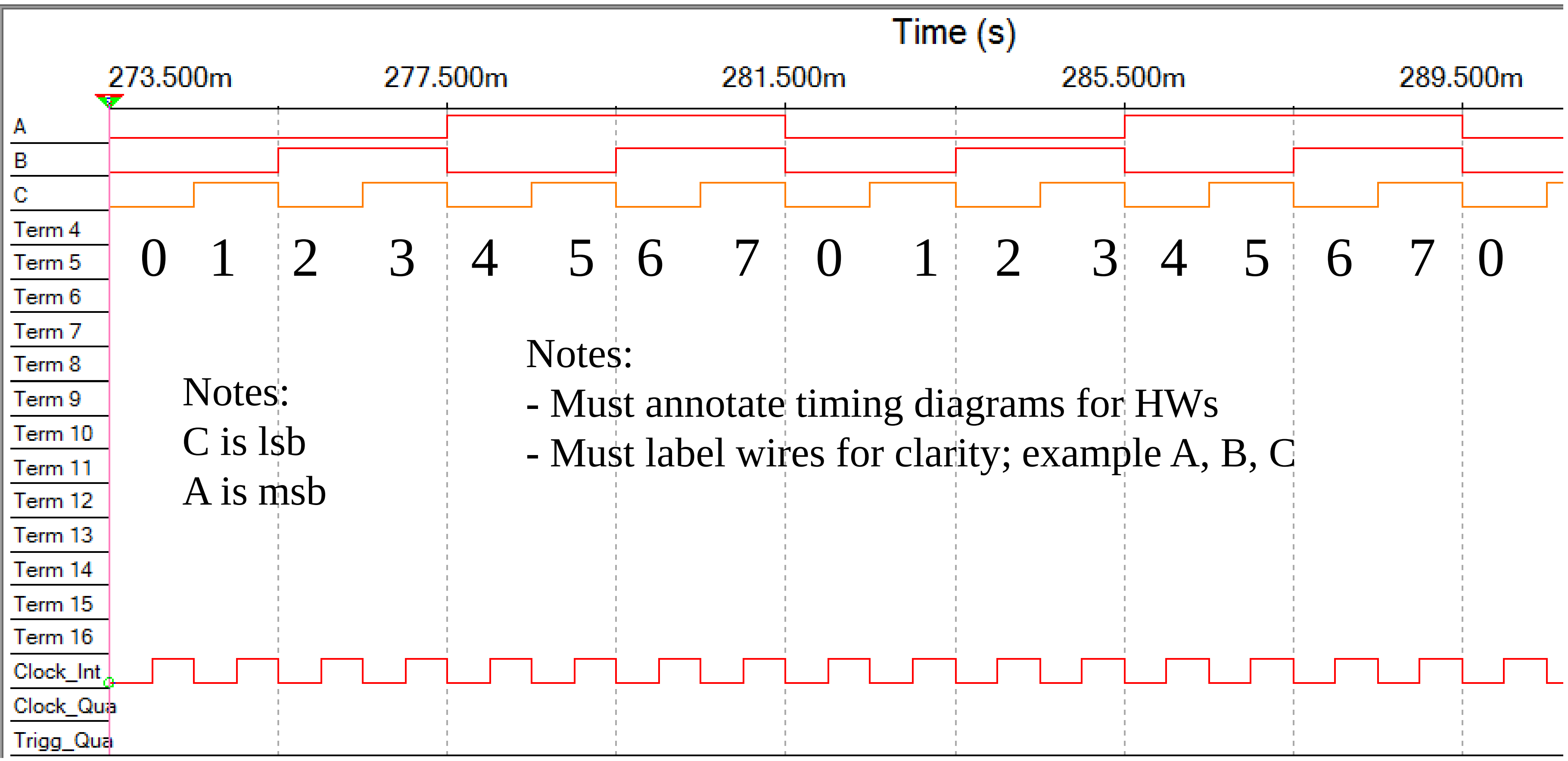


Ripple Counter Simulation (Multisim)



- 3-bit Ripple Counter in MultiSIM (000 → 111)
- Use 7476N Dual JK Flip-flops (neg. edge triggered)
  - Set J and K inputs to logic 1 (5 volts); toggle mode
- Use voltage clock as source with digital ground
- Use Logic Analyzer (LA) to view timing diagrams (double click on LA)
- Label wires A, B, C (click on wires →properties)

Ripple Counter Simulation (Multisim)

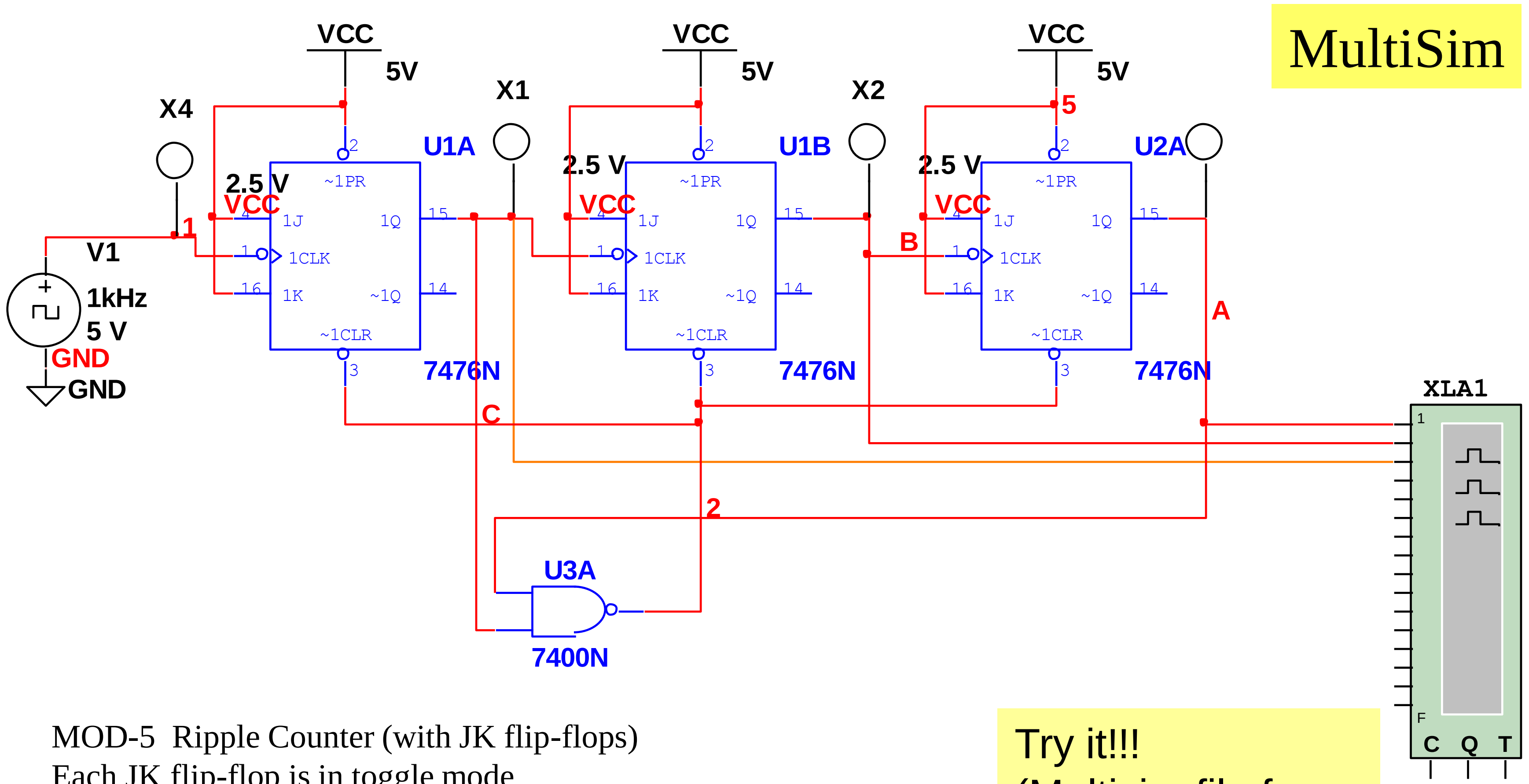


Notes:
C is lsb
A is msb

Notes:
- Must annotate timing diagrams for HWs
- Must label wires for clarity; example A, B, C

1) Logic Analyzer → Set → internal clock → set freq. same as clock source
2) Trigger → set → negative edge
3) Set clocks/div to 1 or 2 (number of clock pulses per division)
4) MultiSIM → Tools → Capture Screen Area (copy and paste into docs)
5) Use stop and reset (run); use horizontal scroll bar.

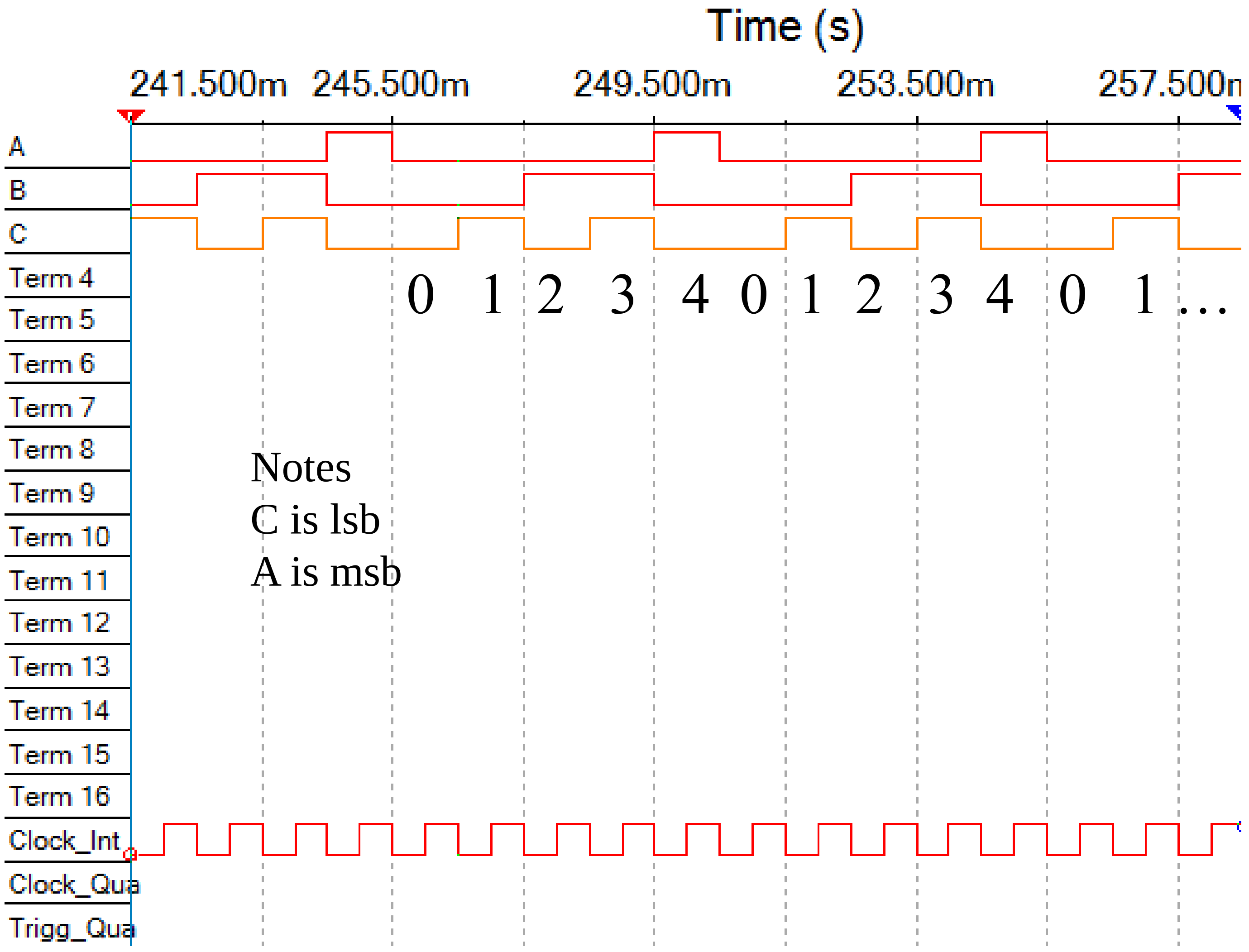
Ripple Counter Simulation (MOD-5)



MOD-5 Ripple Counter (with JK flip-flops)
Each JK flip-flop is in toggle mode
Count sequence is 0 (000) to 4 (100) then repeat.
Reset (clear) all flip flops when count = 5 (101)
There are total of 5 states (0, 1, 2, 3, 4)
The timing diagram in LA is best viewed at a higher frequency, (example: 1KHz)

Try it!!!
(Multisim file for this circuit is on Canvas)

Ripple Counter Simulation (MOD-5)

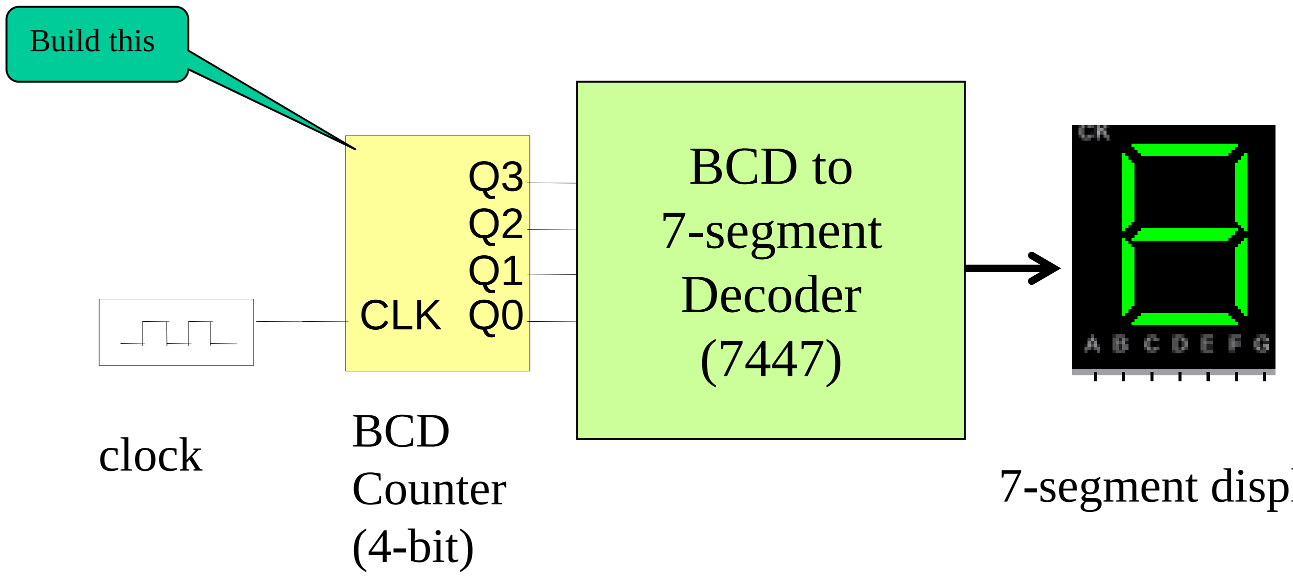


Notes
C is lsb
A is msb

MultiSim

Ripple BCD Counter (MOD-10)

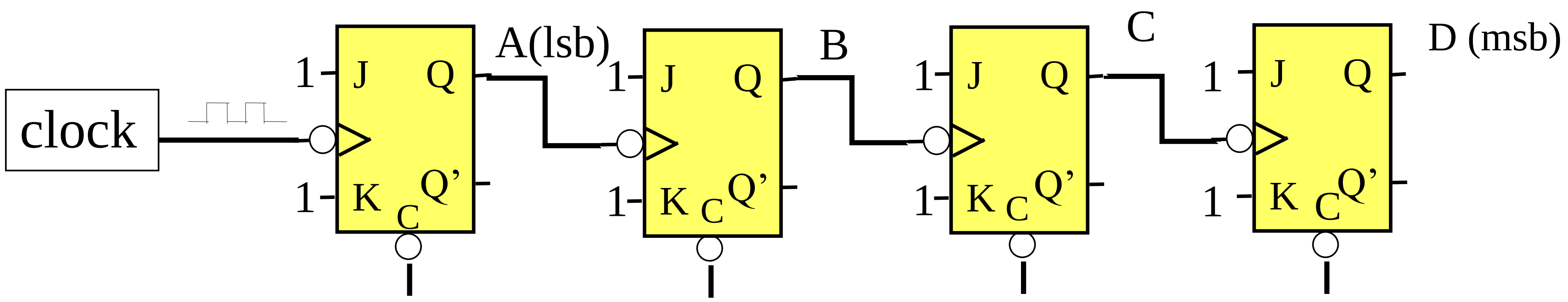
- Design a **<u>MOD-10 (also known as a decade counter or BCD) ripple counter</u>**. What is the count sequence? Are there any glitches or spikes produced? Show timing diagram in a simulation package. Name a few applications of a decade counter. Connect this BCD counter to a 7-segment decoder (ex: 7447) and a 7-segment display
- Simulate in Multisim (or equivalent circuit simulation software)

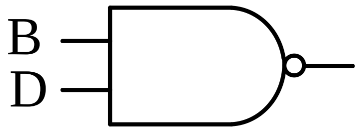


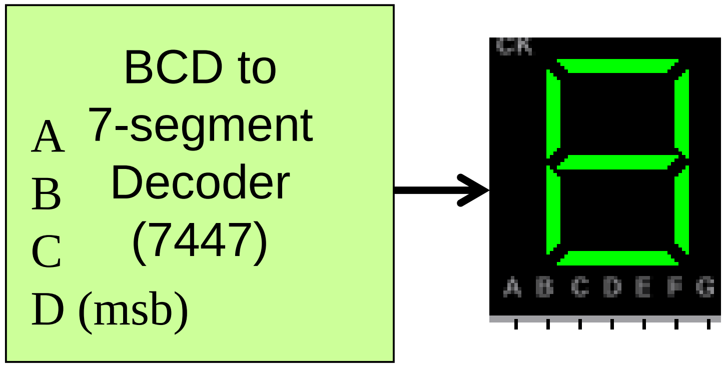
You need this circuit for a future HW problem

Ripple BCD Counter (MOD-10)

- Design a **<u>MOD-10 (also known as a decade counter or BCD) ripple counter</u>**.

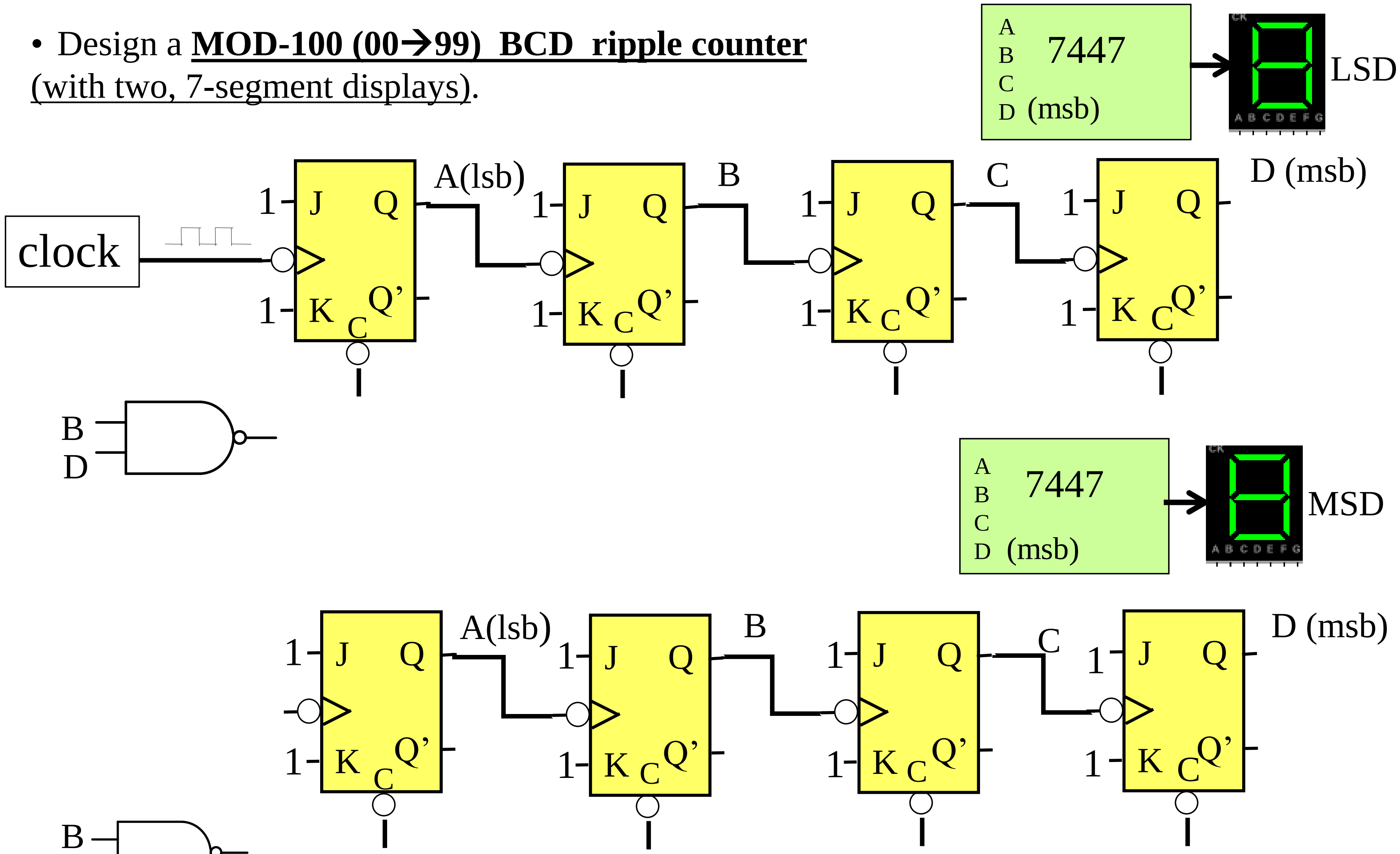






Ripple BCD Counter (MOD-100)

- Design a **MOD-100 (00→99) BCD ripple counter** (with two, 7-segment displays).



How do you modify to create MOD-50 counter (or MOD-n for any n)?

Ripple Counter Exercises

- Design a **4-bit, MOD-16 ripple counter**. How many FFs are needed? What is the count sequence? What is the frequency of the lsb? msb?

- Design a **5-bit, MOD-32 ripple counter**. How many FFs are needed? What is the count sequence? What is the frequency of the lsb? msb?

- Design a **MOD-6 ripple counter**. How many FFs are needed? What is the count sequence? Are there any glitches or spikes produced? Show timing diagram in a simulation package.

- **Design a MOD-n ripple counter.** Let n be any integer. For example, 7, 8, 9, 10, 11, 30, 100, etc.

- Design **ripple counter** with count sequence 2,3,4,5, repeat. There are several options.

- Assume a 8MHz controller board needs a 2Mhz clock signal to control a slower device. **Design a circuit** to supply a 2Mhz signal from the 8Mhz clock signal. That is, the input to the circuit is a 8MHz clock signal, and the output is 2Mhz clock signal. Use LogicWorks or MultiSim

JK Flip-flop Summary

- JK flip-flops have **two data inputs**: J (set) and K (reset). If both J and K are set to 1, then the JK flip-flop is in "toggle" mode.
- We will use JK flip-flops which are **negative edge triggered**.
- You can assume the ***hold time*** for edge-triggered JK flip-flops is zero. This means that if the J and/or K inputs are changing at the same time as the clock transition, then the FF "sees" the values of J and K just prior to the clock transition.
- A common applications of JK flip-flops (FF) is a **binary ripple counter**.
- Ripple counters are called "**asynchronous" circuits** because each FF is **not** connected directly to the external clock. (Only one JK FF is connected directly to the ext. clock)
- In a ripple counter, both the J and K inputs of each FF are connected to 1 (+5 volts), and therefore, each JK FF is in **toggle mode**.
- In a **ripple counter**, the Q output of one JK FF is connected to the clock of the adjacent JK FF.
- An n-bit ripple counter **contains "n" JK flip-flops** and can count from 0 to (2^n)-1.
- When a ripple binary counter reaches the maximum count value, it automatically reset **back to zero** on the next clock pulse and this behavior repeats.
- A mod-n ripple counter has **n count states**.
- In order to modify a ripple counter to reset to zero at an arbitrary count value (which is not the maximum value), it is possible to use a **NAND gate connected** to the active-low reset of each FF.
- Ripple counters can be used to **reduce a clock frequency** to a lower frequency.

General Flip-flop Concepts

- **Edge-triggered flip-flops** such D flip-flops and JK flip-flops are more advanced than level-triggered D latches. Edge-triggered FFs allow for simple and predictable circuit designs.
- Another type of flip-flops called **"master-slave" flip-flops** were the predecessors of edge-triggered flip-flops, and are not in common use today. (See Mano, Wakerly for more details)
- A common application of D flip-flops is a **shift register or a ring counter** (which are both synchronous circuits).
- A common application of JK flip-flops is a **ripple binary counter** (which is an asynchronous circuit).
- Flip-flops (both D and JK flip-flops) can have **asynchronous inputs** to set or clear the FF. Asynchronous FF inputs have priority over data inputs. Asynchronous inputs are inputs that can be activated independent of the clock (i.e. not synchronized to the clock).
- A single flip-flop (either D or JK) can store **one bit** of information. A flip-flop has 2 **states** (when Q=0 and when Q=1).
- A **T flip-flop** is another type of flip-flop which is equivalent to a JK flip-flop in toggle mode. A T flip-flop has one input T and when T=1, then the flip-flop toggles (Q=Qold'), and when T= 0, the T flip-flop remains in its current state (no change).
- In this course, we will **focus on D Flip-flops and JK flip flops**

What you should know about FFs

- Be able to complete a **timing diagram** for a D or JK flip-flop.
- Know and be able to interpret the **block diagram** of a D or JK flip-flop (including asynchronous rest and clear)
- Be able to design an n-bit **shift register** (for any n) from D flip-flops. Show timing diagrams.
- Be able to design an **n-bit ring counter** (any n) and applications from D flip-flops. Know appropriate initialization of flip-flops. Be able to generate timing diagrams.
- Know the relationship between the following concepts: 1) MOD-n counter, 2) divide-by-n counter, 3) count sequence of a counter.
- Be able to design a **MOD-n (for any n) ripple counter** using JK flip-flops. Be able to identify any glitches or spikes in output. Generate timing diagrams for all outputs and identify counting sequence. Be able to specify the frequency of each FF output given the clock frequency.
- Be able to design circuits using ring counters and/or ripple counters **to turn on and turn off devices at specified intervals**. Example: turn on device for 5 seconds, then off for 10 seconds, then repeat.

Penn State Abington
CMPEN 271
Lecture Set #17
JK Flip Flops, Ripple Counters

R. Avanzato ©

<u>**Topics:**</u>

• JK flip-flops	Part 1 of 4
• JK FF applications - "ripple" counter	Part 2 of 4
• Timing Diagrams	
• BCD counter	
• HW #9 A,B (counters)	**Part 3 of 4 🡐**
• Review Questions	Part 4 of 4

HW #9A: Binary Ripple Counter (see schedule)

#1. Design and simulate a **binary ripple counter** with the MOD equal to the last 2 digits (LSDs) of your PSU email address. (If the last 2 digits of your PSU email address is less than 11 or is any power of 2, then use MOD=50). For example, if the last 2 digits of your PSU email address is 73, then you should design and simulate a MOD-73 counter (counts from 0 to 72, then repeats.). Example, if your PSU email address is XYZ5435, then you would design a MOD-35 counter. If your PSU email address is XYZ5432, then you would design a MOD-50 counter. If your PSU email address is XYZ2405, then you would design a MOD-50 counter.

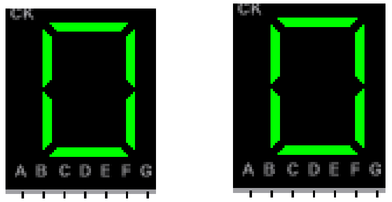
Use **negative-edge triggered JK flip flops** in your design and any other logic gates as needed. The output in Multisim should be set of binary **LED indicators** (be careful with labeling). Include timing diagrams (with reset) with markups (use Logic Analyzer). Include critical portions of the timing diagrams. Include timing diagram markups for 2 counts before the reset to 0, count=0, and 2 counts after the reset to 0. Show count values in both binary and in decimal. Label msb and lsb. For example, if you have to design a MOD 50 counter (count sequence 0 to 49) then you would show binary and decimal markups on timing diagram for counts 48, 49, 0, 1, 2.

HW #9B: BCD Ripple Counter (see schedule)

#2. Design and simulate a **BCD ripple counter** with the MOD equal to the last 2 digits, (LSDs) of your PSU email address. (If the last 2 digits of your PSU email address are less than 11 or a power of 2, then let MOD=50). See instructor for additional details.

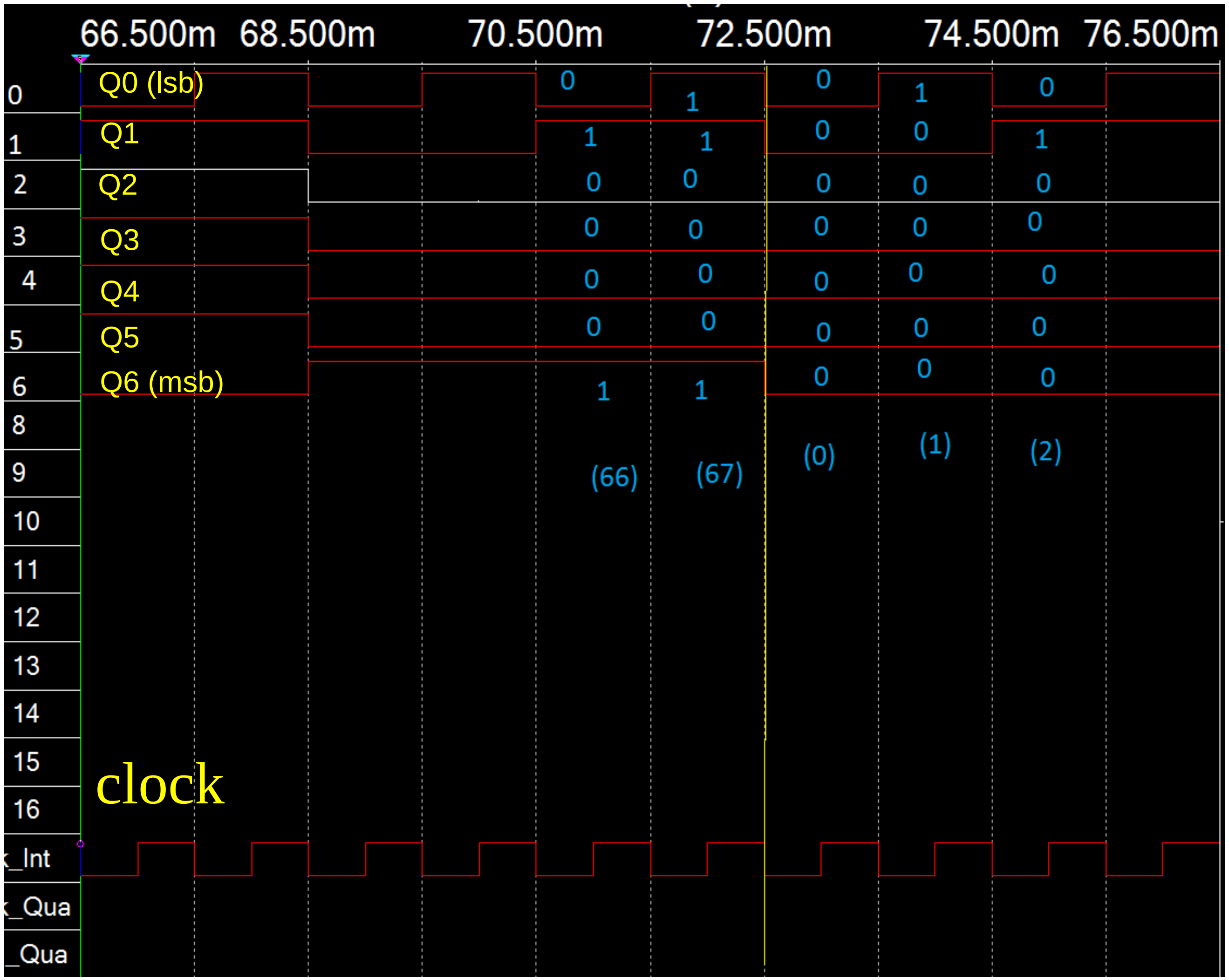
The output to this circuit are **two, 7-segment LED displays**. Use **7447** (or equivalent) drivers. Use negative-edge triggered JK flip flops in your design and any other logic gates as needed. Include timing diagrams (with reset) with markups. Include critical portions of the timing diagrams with annotations. Include timing diagram markups for 2 counts before the reset to 0, count = 0, and 2 counts after the reset to 0. Show count values in both binary and in decimal. Label msb and lsb for each counter/digit. For example, if you have to design a MOD 50 counter (count sequence 0 to 49) then you would show binary and decimal markups on timing diagram for counts 48, 49, 0, 1, 2.
Hint: start with the 00 to 99 BCD counter discussed earlier in this lecture. Make sure the 00 to 99 BCD counter fully works with the 7-segment displays in Multisim and then modify to reset at a new count value.

MSD
(most sign.
digit)



LSD
(least sign.
digit)

Example: Timing Diagrams for HW #9A Binary Counter MOD 68 (counts 0 to 67)



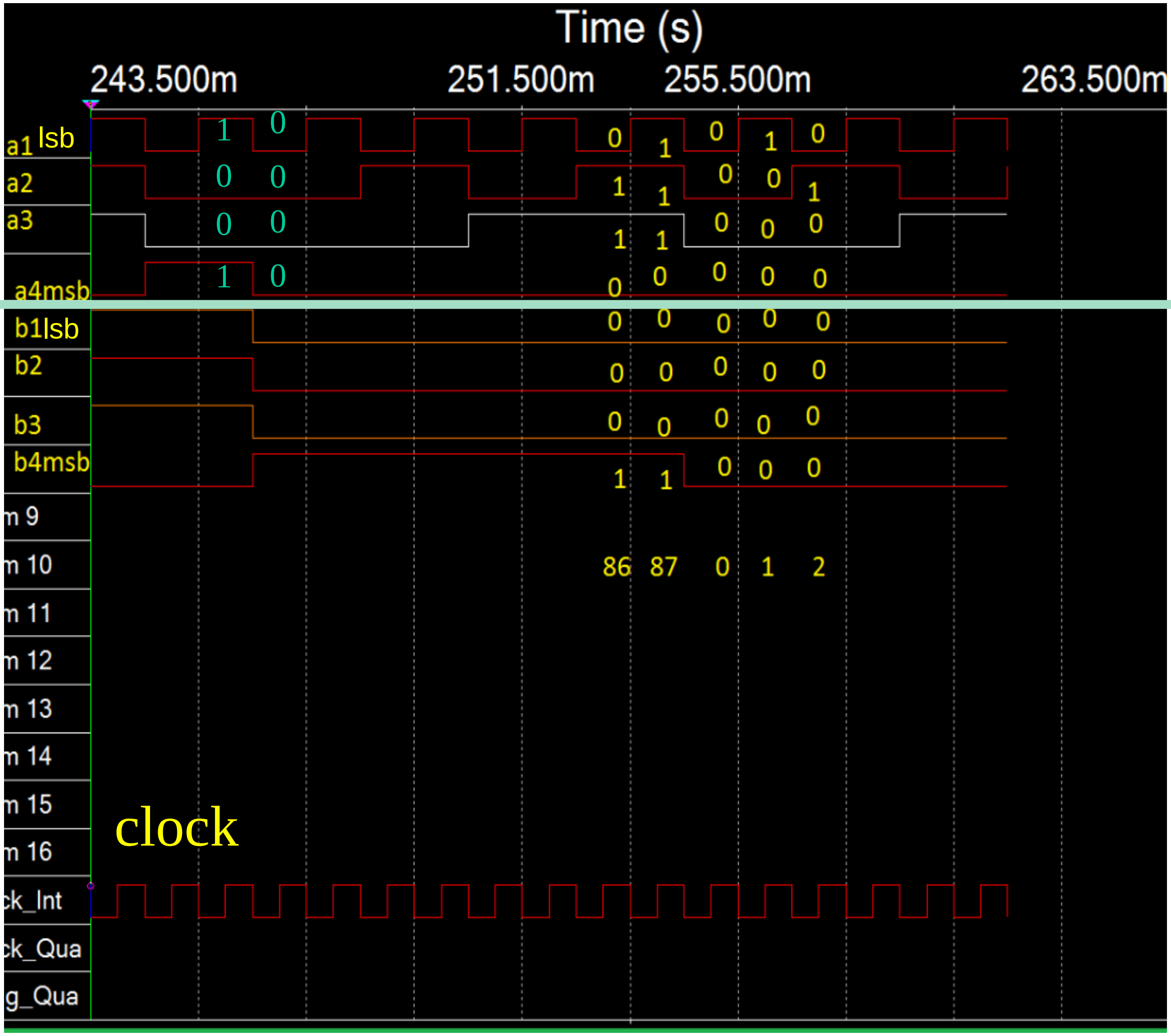
How many FFs are needed?

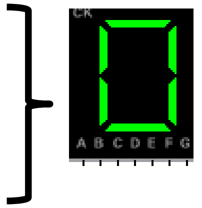
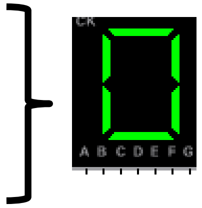
Can you connect output of binary counter directly to 7-segment LEDs?

Example: Timing Diagrams for HW #9B
BCD Counter MOD 88 (counts 00 to 87 BCD)

MOD-10
CTR
(LSD)

MOD-10
CTR
(MSD)



How many FFs required?

What is advantage of BCD counter?

Disadvantage?

Application of BCD counters? (clocks, timers, etc.)

Note: preferred notation is a0, a1, a2, a3

Penn State Abington CMPEN 271 Lecture Set #17

JK Flip Flops, Ripple Counters

R. Avanzato ©

<u>Topics:</u>

• JK flip-flops	Part 1 of 4
• JK FF applications - “ripple” counter	Part 2 of 4
• Timing Diagrams	
• BCD counter	
• HW #9 A,B (counters)	Part 3 of 4
• Review Questions	**Part 4 of 4 🡐**

Review Questions

#1.- How many bits are stored in a single JK flip flop?

a) 4   b) 3   c) 2   d) 1

#2.- To place a JK flip-flop in "toggle" mode, the data inputs should be

a) J=1, K=0   b) J=0, K=0   c) J=0, K=1   d) J=1, K=1

#3.- To "set" a JK flip-flop, the inputs should be (assuming initial Q is unknown)

a) J=1, K=1   b) J=0, K=0   c) J=0, K=1   d) J=1, K=0

#4.- To "reset" or "clear" a JK flip-flop, the inputs should be (assuming initial Q is unknown)

a) J=1, K=1   b) J=0, K=0   c) J=1, K=0   d) J=0, K=1

#5.- If J=1 and K=1 for a JK flip-flop and the initial Q = 0, what are the outputs after one clock pulse?

a) Q = 1, Q’ = 1   b) Q = 0, Q’ = 0

c) Q = 1, Q’ = 0   d) Q = 0, Q’ = 1

Review Questions

#6.- A 3-bit ripple counter is also known as a

a) MOD-3 counter b) MOD-4 counter c) MOD-8 counter
d) MOD-16 counter

#7.- A 3-bit ripple counter is also known as a

a) divide-by-3 counter b) divide-by-4 counter c) divide-by-8 counter
d) divide-by-16 counter

#8.- A MOD-9 ripple counter has a count sequence of

a) 0 to 3 b) 0 to 8 c) 0 to 9 d) 0 to 10

#9.- A MOD-9 ripple counter requires how many JK flip-flops?

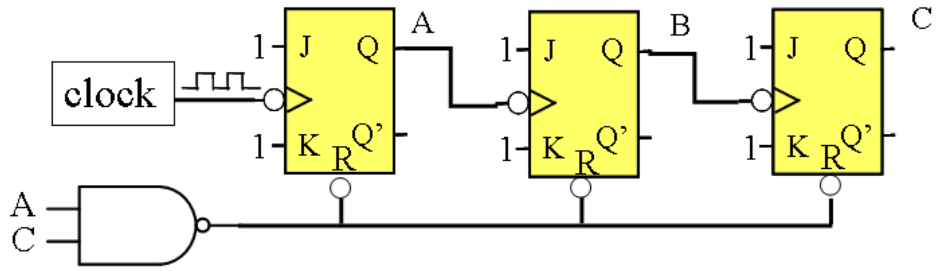
a) 2 b) 3 c) 4 d) 5

#10.- If the clock input to a 3-bit ripple counter is 1KHz, then the frequency of the msb of the counter is

a) 125Hz b) 1KHz c) 250Hz d) 500Hz

Review Questions

Consider the ripple counter circuit below: Note: R → reset or clear asynch. input



#11.- How many FFs are contained in the circuit above?

a) 2 b) 3 c) 4 d) 8

#12.- What is the MOD of the ripple counter above?

a) MOD 2 b) MOD 3 c) MOD 5 d) MOD 8

#13.- What is the count sequence of the ripple counter above?

a) 0 to 4 b) 1 to 4 c) 1 to 5 d) 0 to 5 e) 0 to 1