Cascaded Counters
A Mod4 Counter
(A 2-bit counter)

<table>
<thead>
<tr>
<th>CLR</th>
<th>INC</th>
<th>Q1</th>
<th>Q0</th>
<th>N1</th>
<th>N0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
A Mod4 Counter With a Rollover Signal

<table>
<thead>
<tr>
<th>CLR</th>
<th>INC</th>
<th>Q1</th>
<th>Q0</th>
<th>N1</th>
<th>N0</th>
<th>Rollover</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Signal Rollover can be used to tell other circuitry that counter is rolling over to all 0's
Cascading 2 Mod4 Counters

Increment higher digit's counter when lower digit's counter is rolling over.

Digit1

Digit0

Count sequence

Mod4 Counter

Mod4 Counter

Rollover1

Rollover0

INC

CLR

INC

CLR

clk

2

clk

2

clk

ECE 238L
3 Digits’ Worth

Could build a digital watch or clock circuit this way with Mod60 and Mod24 counters

Increment higher digit’s counter when lower digit’s counter is rolling over

Can do this with any counter that has a Rollover output

ECE 238L
Cascading Counters

• A counter will increment only when
  - The counter below it is at its terminal count and it is being incremented
  • That is the definition of the Rollover signal

• Some people try to tie Rollover to the clk input of the next higher counter
  - Bad idea... Very bad idea...
  - Violates our Globally Synchronous policy
  - Doesn’t work as intended
Sequence should be:
...-12-13-20-21-22-23-30-...

but we get:
...-12-23-20-21-22-33-30-...

?????

DO NOT TIE CLK inputs on modules to anything but the clock !!!!!!

Even if you *tinker* until you get the right count sequence, you must guarantee that signal Rollover0 has no hazards

Digit0 transition from 1-2 makes this difficult if not impossible
Sequence should be:
...-12-13-20-21-22-23-30-...

but we get:
...-12-23-20-21-22-33-30-...

?????

DO NOT TIE CLK inputs on modules to anything but the clock !!!!!!

Even if you tinker until you get the right count sequence, you must guarantee that signal Rollover0 has no hazards

Digit0 transition from 1-2 makes this difficult if not impossible

Possible hazard
Ripple Counters

• When you tie a rollover-like signal to a clock on the next higher digit \(\Leftrightarrow\) ripple counter

• A ripple counter is an ASYNCHRONOUS counter
  - Transitions are not all synchronized to the clock
  - Different flip flops change at different times
  - Similar to gated clocks (seen earlier)

• Asynchronous circuits are an advanced topic
Another Common Ripple Counter

Sequence is:

0000
0001
0010
0011
0100
0101
0110
0111
1000
1001
1010
1011
1100
1101
1110
1111
0000

So what is the problem?
Timing Diagram

- Q0 changes in response to clock edge.
- Only after it changes does Q1’s FF get a clock.
- Only after that does Q2’s FF get a clock.
- Net effect is that all the FF’s change at different times.
- Logic depending on Q3 has very little time to react before next clock edge.
Do Not Use Asynchronous or Ripple Counters
Mod4 Counter

\[(\text{CEin} \cdot \text{Reset}')/\text{CEout}\]
Note: Both CLK and CE are required to change the flip-flop
Mod4 Counter

<table>
<thead>
<tr>
<th>CEin Q1 Q0</th>
<th>N1</th>
<th>N0</th>
<th>CEout</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0 0</td>
<td>0 0</td>
<td>0 0</td>
<td>0</td>
</tr>
<tr>
<td>0 0 1</td>
<td>0 1</td>
<td>1 0</td>
<td>0</td>
</tr>
<tr>
<td>0 1 0</td>
<td>1 0</td>
<td>0 0</td>
<td>0</td>
</tr>
<tr>
<td>0 1 1</td>
<td>1 1</td>
<td>1 0</td>
<td>0</td>
</tr>
<tr>
<td>1 0 0</td>
<td>0 0</td>
<td>0 1</td>
<td>1</td>
</tr>
<tr>
<td>1 0 1</td>
<td>1 0</td>
<td>0 0</td>
<td>0</td>
</tr>
<tr>
<td>1 1 0</td>
<td>1 1</td>
<td>1 0</td>
<td>0</td>
</tr>
<tr>
<td>1 1 1</td>
<td>0 0</td>
<td>0 1</td>
<td>1</td>
</tr>
</tbody>
</table>

Note: No change when CEin=0

CEout = CeQ_1Q_0

N1 = Ce’Q_1 + Q_1Q_0’ + CeQ_1’Q_0

N0 = Ce’Q_0 + CeQ_0’
Cascaded Ripple Counter

Stage 2 increments is too early
Cascaded Ripple Counter

Why is Next Stage increment too early?

Stage 1

<table>
<thead>
<tr>
<th>CEout</th>
<th>master loads</th>
<th>slave loads</th>
</tr>
</thead>
<tbody>
<tr>
<td>CEout from Stage 1 is used as the clock for Stage 2</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Stage 2

<table>
<thead>
<tr>
<th>CEout</th>
<th>master loads</th>
<th>slave loads</th>
</tr>
</thead>
<tbody>
<tr>
<td>CEout from Stage 1 is used as the clock for Stage 2</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Cascaded Ripple Counter

A Ripple Counter is an ASYNCHRONOUS counter since it’s transitions are NOT synchronized to the system clock.

In a SYNCHRONOUS counter, all stages transition with the system clock.
Cascaded Synchronous Counter

Stage 2 is incremented at the correct time.
Cascaded Synchronous Counter

System Clock: master loads — slave loads — master loads — slave loads

Stage 1: 10 → 11 → 00

CEout

Stage 2 master loads

Stage 2 slave loads

Stage 2: 00 → 01
Cascaded Counters

The more stages you add to the counter, the bigger the discrepancy between Synchronous and Asynchronous counters.
Cascaded Synchronous Counter

$CE_{in}$ controls when the S.M. can change

$CE_{in}$ also controls when the $CE_{out}$ will be generated
Cascaded Synchronous Counter

Note that the CEout from each stage is synchronized with the system clock and with each other. The stage transitions are all synchronized with the system clock.
Cascaded Synchronous Counter

Lesson: **Always** use SYNCHRONOUS counters and timers.
A Mod4 Counter

<table>
<thead>
<tr>
<th>CLR</th>
<th>INC</th>
<th>Q1</th>
<th>Q0</th>
<th>N1</th>
<th>N0</th>
<th>RO</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Count Value

Terminal Count

D Q

IFL

Clr

Inc
A Mod4 Counter

<table>
<thead>
<tr>
<th>CLR INC Q1 Q0</th>
<th>N1 N0 RO</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0 0 0</td>
<td>0 0 0</td>
</tr>
<tr>
<td>0 0 0 1</td>
<td>0 1 0</td>
</tr>
<tr>
<td>0 0 1 0</td>
<td>1 0 0</td>
</tr>
<tr>
<td>0 0 1 1</td>
<td>1 1 0</td>
</tr>
<tr>
<td>0 1 0 0</td>
<td>0 1 0</td>
</tr>
<tr>
<td>0 1 0 1</td>
<td>1 1 0</td>
</tr>
<tr>
<td>0 1 0 1</td>
<td>1 1 0</td>
</tr>
<tr>
<td>0 1 1 1</td>
<td>1 0 1</td>
</tr>
<tr>
<td>1 - - -</td>
<td>0 0 0</td>
</tr>
</tbody>
</table>

You wouldn't really build it like this!
Cascaded Counters

MOD4

MOD4

MOD4

Inc

Clk

Clr

Clk

Clr

Clk

Inc

Inc

Inc
Cascaded Counters

Assume that the second timer is already at the terminal count
Cascaded Counters
Cascaded Counters

Diagram showing cascaded MOD4 counters with inputs Clk,Clr, and outputs Inc.
Cascaded Counters

MOD4

Inc

CV

TC

Clk

Clr

MOD4

Inc

CV

TC

Clk

Clr

MOD4

Inc

CV

TC

Clk

Clr

Inc

ECE 238L
Cascaded Counters

![Cascaded Counters Diagram]

The diagram shows a cascaded counter system with three MOD4 counters. Each counter has inputs for Clk, Cntr, and an output for Inc. The counters are interconnected with signals for Clk, Cntr, and Inc, forming a cascaded system.
It looks like the Inc signal will ripple from counter to counter. Why is this any different from the toggle flip-flop example?
A Mod4 Counter

The right way!
A Mod4 Counter

MOD4

Clear

Increment

Roll Over

Clock

Count Value