Digital Design and Computer Organization

Counters - 2

Reetinder Sidhu

Department of Computer Science and Engineering

DIGITAL DESIGN AND COMPUTER ORGANIZATION

Counters - 2

Reetinder Sidhu

Department of Computer Science and Engineering

COUNTERS - 2

Course Outline

- Digital Design
  - Combinational logic design
  - Sequential logic design
    - ★ **Counters - 2**
- Computer Organization
  - Architecture (microprocessor instruction set)
  - Microarchitecure (microprocessor operation)

Concepts covered

- Arbitrary Modulus Counters

What is the Modulus of a Counter?

Counter Modulus

- The number of states a counter sequences through before repeating is the **modulus** of the counter
  - An n-bit counter counting from 0 to $2^n - 1$ has a modulus of 2^n
- A counter of modulus k is also called a modulo k or mod k counter
- Counters studied so far have a modulus of 2^n
- How to construct counters whose modulus is not 2^n?

Arbitrary Modulus Counter

- Suppose we want a modulus k incrementing counter that counts from 0 to $k - 1$
- Select value n such that $2^{n-1} < k < 2^n$
- Start with an n-bit (modulus 2^n) incrementing counter
- Two things are required:
  - Ability to detect when the count value has reached $k - 1$
  - Ability to reset the count value to 0 when above happens

COUNTERS - 2

Arbitrary Modulus Counter

- Counter value can be reset to 0 if D flip-flops with reset are used
- An AND gate can be used to detect when the count value becomes $k - 1$
  - Some inputs may need to be inverted

Modulus 5 Incrementing Counter

- Count sequence: 000, 001, 010, 011, 100, 000, 001, ...

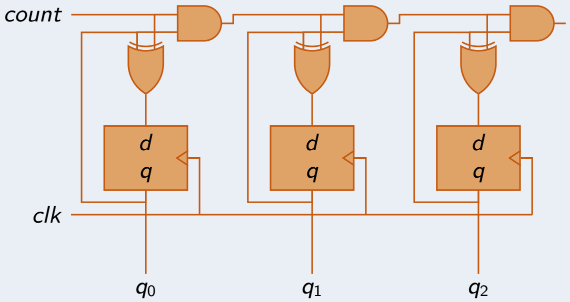
COUNTERS - 2

Arbitrary Modulus Counter

- Counter value can be reset to 0 if D flip-flops with reset are used
- An AND gate can be used to detect when the count value becomes $k - 1$
  - Some inputs may need to be inverted

Modulus 5 Incrementing Counter

- Count sequence: 000, 001, 010, 011, 100, 000, 001, ...
- Since $2^2 < 5 < 2^3$, we start with a 3-bit incrementing counter

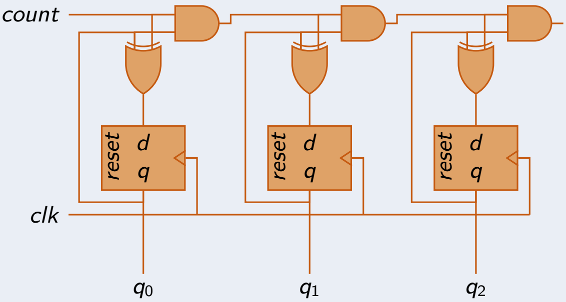


Counters - 2
Arbitrary Modulus Counter

- Counter value can be reset to 0 if D flip-flops with reset are used
- An AND gate can be used to detect when the count value becomes $k - 1$
  - Some inputs may need to be inverted

Modulus 5 Incrementing Counter

- Count sequence: 000, 001, 010, 011, 100, 000, 001, ...
- Since $2^2 < 5 < 2^3$, we start with a 3-bit incrementing counter
- D flip-flops with reset are used



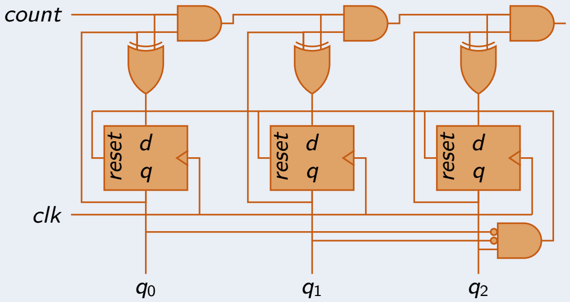
COUNTERS - 2

Arbitrary Modulus Counter

- Counter value can be reset to 0 if D flip-flops with reset are used
- An AND gate can be used to detect when the count value becomes $k - 1$
  - Some inputs may need to be inverted

Modulus 5 Incrementing Counter

- Count sequence: 000, 001, 010, 011, 100, 000, 001, ...
- Since $2^2 < 5 < 2^3$, we start with a 3-bit incrementing counter
- D flip-flops with reset are used
- AND gate used to detect count value 100 ($k - 1$)



Resettable Flip-Flop

Resettable Flip-Flop

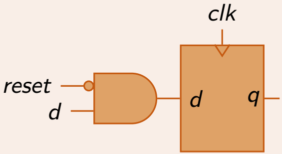
Flip-flop with additional **reset** signal used to store 0 irrespective of the input

Resettable Flip-Flop

Resettable Flip-Flop

Flip-flop with additional **reset** signal used to store 0 irrespective of the input

- Logic Diagram:

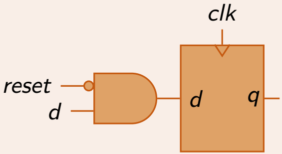


Counters - 2
Resettable Flip-Flop

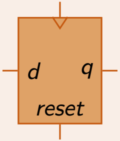
Resettable Flip-Flop

Flip-flop with additional **reset** signal used to store 0 irrespective of the input

- Logic Diagram:



- Symbol:

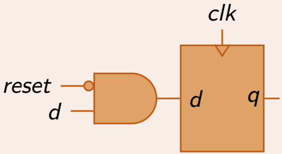


Resettable Flip-Flop

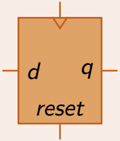
Resettable Flip-Flop

Flip-flop with additional **reset** signal used to store 0 irrespective of the input

- Logic Diagram:



- Symbol:



- At the rising edge of *clk*:

reset	*d*	*q*
0	0	0
0	1	1
1	0	0
1	1	0

COUNTERS - 2

Settable Flip-Flop

Settable Flip-Flop

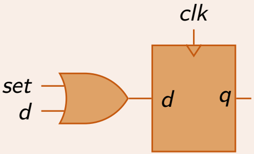
Flip-flop with additional **set** signal used to store 1 irrespective of the input

Settable Flip-Flop

Settable Flip-Flop

Flip-flop with additional **set** signal used to store 1 irrespective of the input

- Logic Diagram:

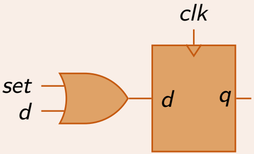


Counters - 2
Settable Flip-Flop

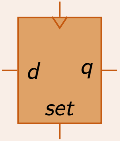
Settable Flip-Flop

Flip-flop with with additional **set** signal used to store 1 irrespective of the input

- Logic Diagram:



- Symbol:

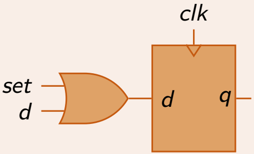


Counters - 2
Settable Flip-Flop

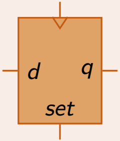
Settable Flip-Flop

Flip-flop with additional **set** signal used to store 1 irrespective of the input

- Logic Diagram:



- Symbol:



- At the rising edge of *clk*:

set	*d*	*q*
0	0	0
0	1	1
1	0	1
1	1	1

COUNTERS - 2

Ring Counter

- Counters constructed by connecting together flip-flops in closed loop, typically with a single 1 bit circulating among them, are called **ring counters**

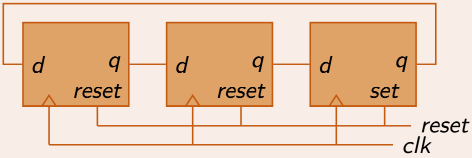
COUNTERS - 2
Ring Counter

- Counters constructed by connecting together flip-flops in closed loop, typically with a single 1 bit circulating among them, are called **ring counters**
- Modulus of an *n*-bit ring counter is *n*

COUNTERS - 2
Ring Counter

- Counters constructed by connecting together flip-flops in closed loop, typically with a single 1 bit circulating among them, are called **ring counters**
- Modulus of an *n*-bit ring counter is *n*

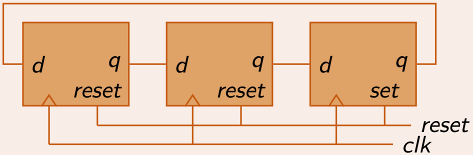
3-bit Ring Counter



Ring Counter

- Counters constructed by connecting together flip-flops in closed loop, typically with a single 1 bit circulating among them, are called **ring counters**
- Modulus of an *n*-bit ring counter is *n*

3-bit Ring Counter

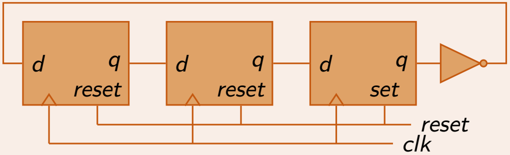


- Also called “one-hot” counters (as in one-hot encoding)

COUNTERS - 2

Think About It

3-bit Johnson (Twisted Ring) Counter



- What is the modulus of an *n*-bit Johnson counter?