Digital Design and Computer Organization

Counters - 1

Reetinder Sidhu

Department of Computer Science and Engineering

Digital Design and Computer Organization

Counters - 1

Reetinder Sidhu

Department of Computer Science and Engineering

Counters - 1

Course Outline

- Digital Design
  - Combinational logic design
  - Sequential logic design
    - ★ **Counters - 1**
- Computer Organization
  - Architecture (microprocessor instruction set)
  - Microarchitecure (microprocessor operation)

Concepts covered

- Counters

Counters - 1

Motivation

COUNTERS - 1

Motivation

- How does the timer/stopwatch app on your mobile work?

Motivation

- How does the timer/stopwatch app on your mobile work?
- The clock period of a 2GHz processor is 0.5 ns

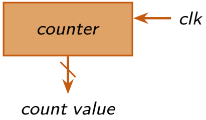
Counters - 1

Motivation

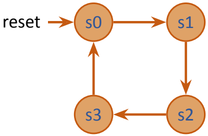
- How does the timer/stopwatch app on your mobile work?
- The clock period of a 2GHz processor is 0.5 ns
- Need a way to count clock periods based on which time intervals such as microseconds, seconds etc. can be measured

Counters

- Counter inputs and outputs:



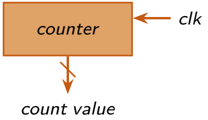
- Counter is a Moore type FSM having a state transition diagram in which nodes are arranged in a circle:



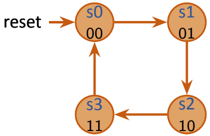
- While in general the sequential circuit design method should be used, common cases can be designed in more intuitive manner

Counters

- Counter inputs and outputs:



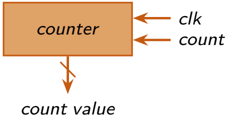
- Counter is a Moore type FSM having a state transition diagram in which nodes are arranged in a circle:



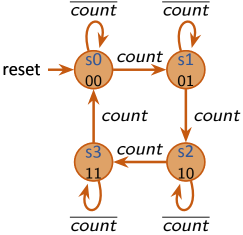
- While in general the sequential circuit design method should be used, common cases can be designed in more intuitive manner

Counters

- Counter inputs and outputs:



- Counter is a Moore type FSM having a state transition diagram in which nodes are arranged in a circle:



- While in general the sequential circuit design method should be used, common cases can be designed in more intuitive manner

COUNTERS - 1

Basic Incrementing Counter

- Consider an n-bit counter that counts from 0 to $2^n - 1$ (and back to 0)
  - Ex: a 2-bit counter, counts 00, 01, 10, 11, 00, 01, ...
- Every clock cycle we need to:
  - Store the current count value
  - Add 1 to it (increment it) to obtain the next value

COUNTERS - 1

Counter Components

Counters - 1

Counter Components

- How to store *n*-bits?

COUNTERS - 1

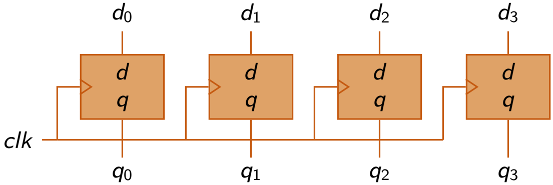
Counter Components

- How to store *n*-bits? *n*-bit register

COUNTERS - 1

Counter Components

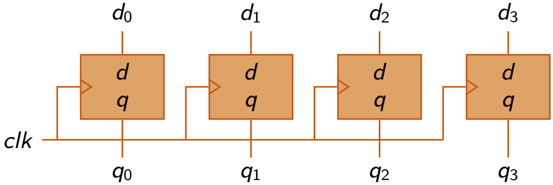
- How to store *n*-bits? *n*-bit register



COUNTERS - 1

Counter Components

- How to store *n*-bits? *n*-bit register

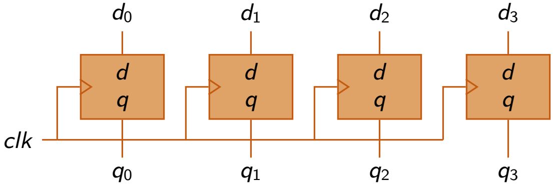


- How to increment an *n*-bit number?

COUNTERS - 1

Counter Components

- How to store *n*-bits? *n*-bit register

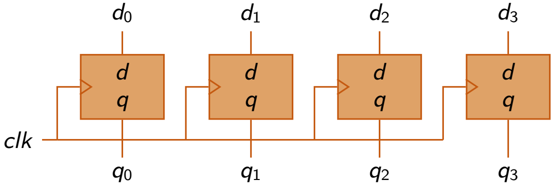


- How to increment an *n*-bit number? *n*-bit incrementer

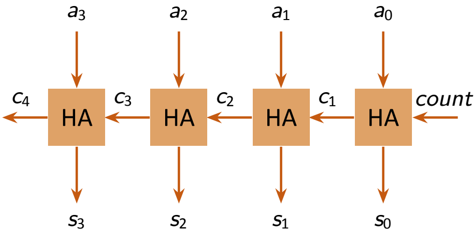
Counters - 1

Counter Components

- How to store *n*-bits? *n*-bit register



- How to increment an *n*-bit number? *n*-bit incrementer

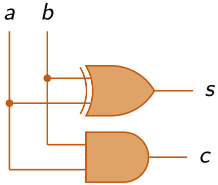


Counters - 1

Half Adder

a	b	s	c
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

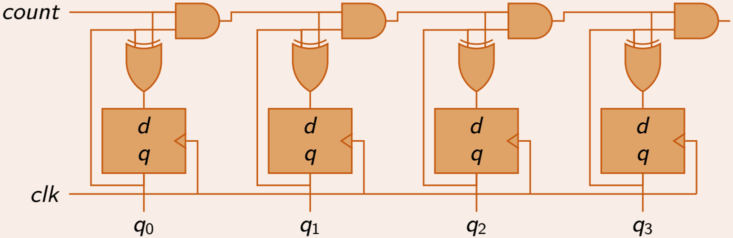
- Half adder logic circuit:



- SOP formulas:
  $s = \overline{a}b + a\overline{b}$
  $c = ab$
- Usually written in terms of XOR function as:
  $s = a \oplus b$
  $c = ab$

Counters - 1
Incrementing Counter

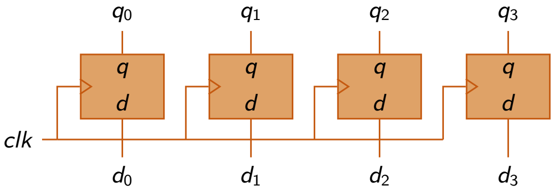
4-Bit Incrementing Counter



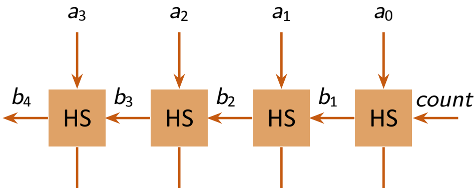
Counters - 1

Decrementing Counter

- An n-bit counter that counts back from $2^n - 1$ to 0 (and back to $2^n - 1$)
  - Ex: a 2-bit counter, counts 11, 10, 01, 00, 11, 10, ...
- How to store n-bits? n-bit register



- How to decrement an n-bit number? n-bit decrementer

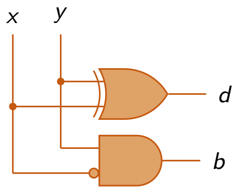


Half Subtractor

x	y	d	b
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

- SOP formulas:
  $d = \overline{x}y + x\overline{y}$
  $b = \overline{x}y$
- Usually written in terms of XOR function as:
  $d = x \oplus y$
  $b = \overline{x}y$

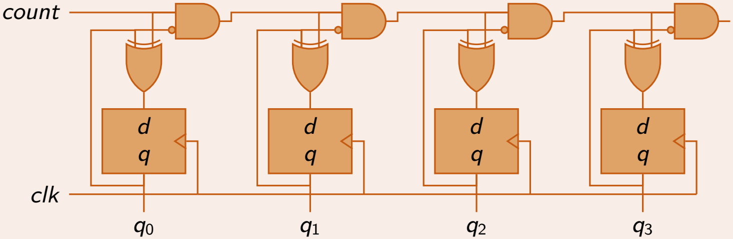
- Half subtractor logic circuit:



COUNTERS - 1

4-Bit Decrementing Counter

4-Bit Decrementing Counter



Counters - 1

Applications

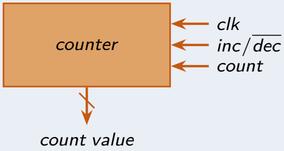
- Iteration (“loops”) in logic design
  - $n \times n$-bit shift-add multiplier requires n iterations
- Interrupt timers
  - OS scheduler preempts process after a fixed timeslice
- Software timeouts
  - Show error if webpage does not load in specified time interval

COUNTERS - 1

Think About It

Incrementer/Decrementer Counter

- Design an incrementer/decrementer counter

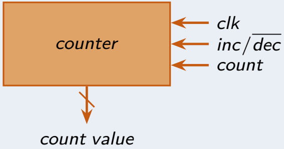


COUNTERS - 1

Think About It

Incrementer/Decrementer Counter

- Design an incrementer/decrementer counter



- Make the logic as simple as possible