Нижегородский государственный университет им. Н.И. Лобачевского

Факультет Вычислительной математики и кибернетики
Кафедра Математического обеспечения ЭВМ

Множества. Битовые поля.

Пирова А.Ю.

Нижний Новгород, 2013 г.

Битовые операции в С

- Сдвиг вправо (деление на степень двойки):

  $1100101_2 >> 2 =$

Битовые операции в С

- Сдвиг вправо (деление на степень двойки):

  $1100101_2 >> 2 = 11001_2$

- Сдвиг влево (умножение на степень двойки):

  $1100101_2 << 1 =$

Битовые операции в С

- Сдвиг вправо (деление на степень двойки):

  $1100101_2 >> 2 = 11001_2$

- Сдвиг влево (умножение на степень двойки):

  $1100101_2 << 1 = 1100101_2$

- Побитовое отрицание, ~:

  $\sim 1100101_2 =$

Битовые операции в С

- Сдвиг вправо (деление на степень двойки):

  $1100101_2 >> 2 = 11001_2$

- Сдвиг влево (умножение на степень двойки):

  $1100101_2 << 1 = 1100101_2$

- Побитовое отрицание, ~:

  $\sim 1100101_2 = 11010_2$

- Побитовое “и”, &:

  $1100101_2 \ \& \ 10010111_2 =$

Битовые операции в С

- Сдвиг вправо (деление на степень двойки):

  $1100101_2 >> 2 = 11001_2$

- Сдвиг влево (умножение на степень двойки):

  $1100101_2 << 1 = 1100101_2$

- Побитовое отрицание, ~:

  $\sim 1100101_2 = 11010_2$

- Побитовое “и”, &:

  $1100101_2 \,\&\, 10010111_2 = 101_2$

- Побитовое “или”, |:

  $1100101_2 \mid 10010111_2 =$

Битовые операции в С

- Сдвиг вправо (деление на степень двойки):

  $1100101_2 >> 2 = 11001_2$

- Сдвиг влево (умножение на степень двойки):

  $1100101_2 << 1 = 1100101_2$

- Побитовое отрицание, ~:

  $\sim 1100101_2 = 11010_2$

- Побитовое “и”, &:

  $1100101_2 \,\&\, 10010111_2 = 101_2$

- Побитовое “или”, |:

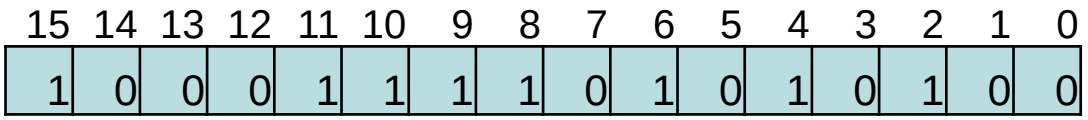
  $1100101_2 \mid 10010111_2 = 11110111_2$

Битовые поля

- Пример: на приборе N лампочек. Каждая лампочка либо горит, либо не горит. Нужно судить о состоянии прибора по лампочкам.
  - Как хранить?

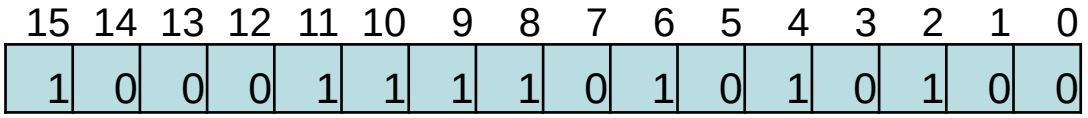
Битовые поля

- Пример: на приборе N лампочек. Каждая лампочка либо горит, либо не горит. Нужно судить о состоянии прибора по лампочкам.
  - Как хранить?
- Представим информацию о состоянии прибора в битах: i-й бит – i-я лампочка



Битовые поля

- Пример: на приборе N лампочек. Каждая лампочка либо горит, либо не горит. Нужно судить о состоянии прибора по лампочкам.
  - Как хранить?
- Представим информацию о состоянии прибора в битах:
  i-й бит – i-я лампочка

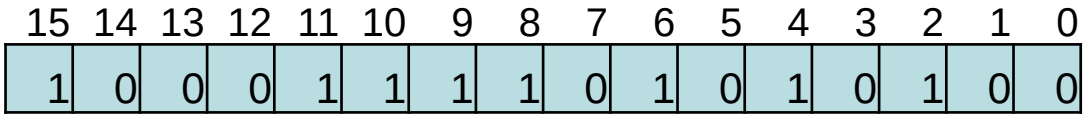


unsigned short state = 36692

- Такой способ хранения – хранение в *битовых полях*

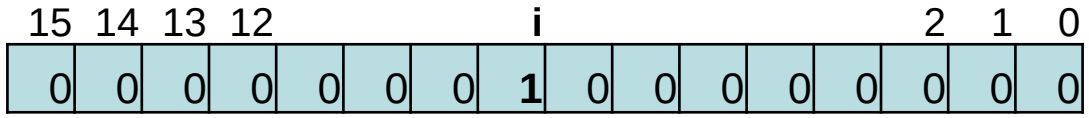
Битовые поля

- Пример: на приборе N лампочек. Каждая лампочка либо горит, либо не горит. Нужно судить о состоянии прибора по лампочкам.
  - Как хранить?
- Представим информацию о состоянии прибора в битах: i-й бит – i-я лампочка



unsigned short state = 36692

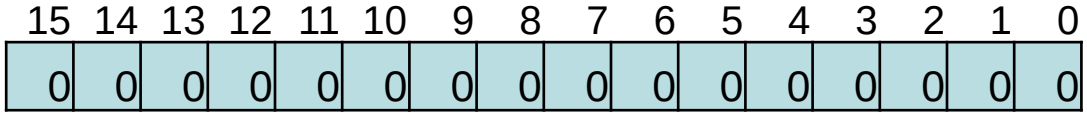
- Такой способ хранения – хранение в *битовых полях*
- *Битовая маска* бита i:



1 << i

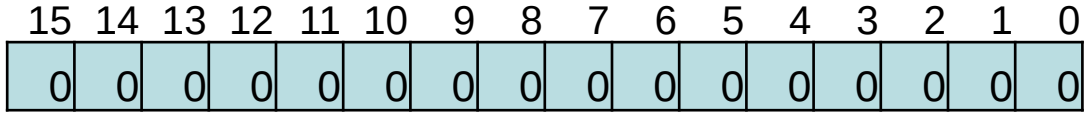
Операции с битовыми полями

- Все лампочки выключены:



Операции с битовыми полями

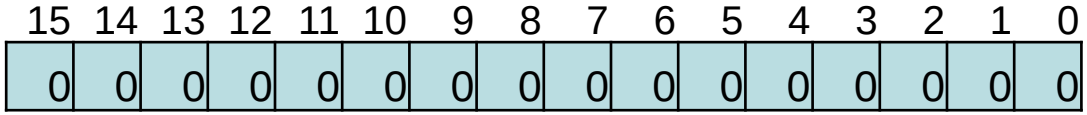
- Все лампочки выключены:



state = 0

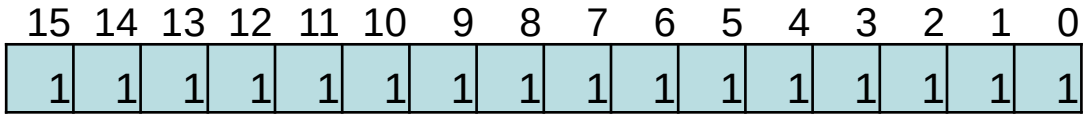
Операции с битовыми полями

- Все лампочки выключены:



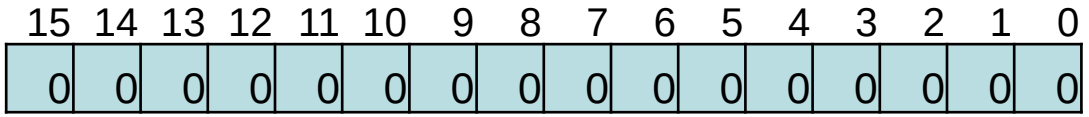
state = 0

- Все лампочки включены:



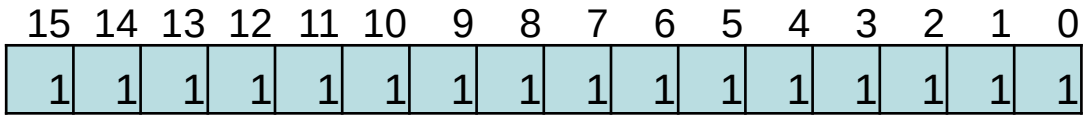
Операции с битовыми полями

- Все лампочки выключены:



state = 0

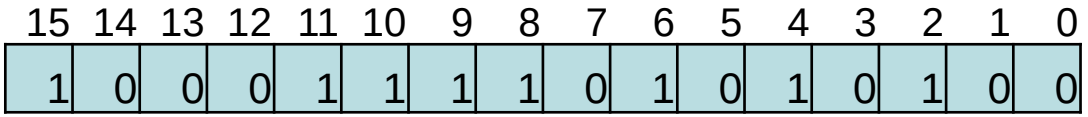
- Все лампочки включены:



state = $2^{16} - 1$

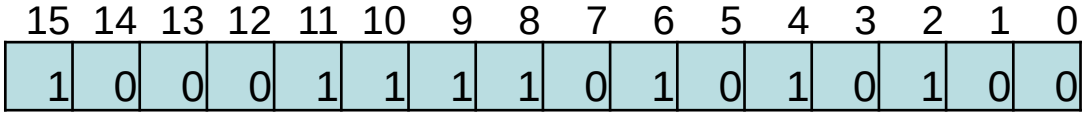
Операции с битовыми полями

- Текущее состояние:



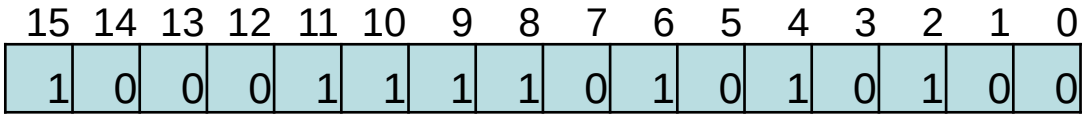
state = 36692

- Включим 7-ю лампочку (7-й бит):



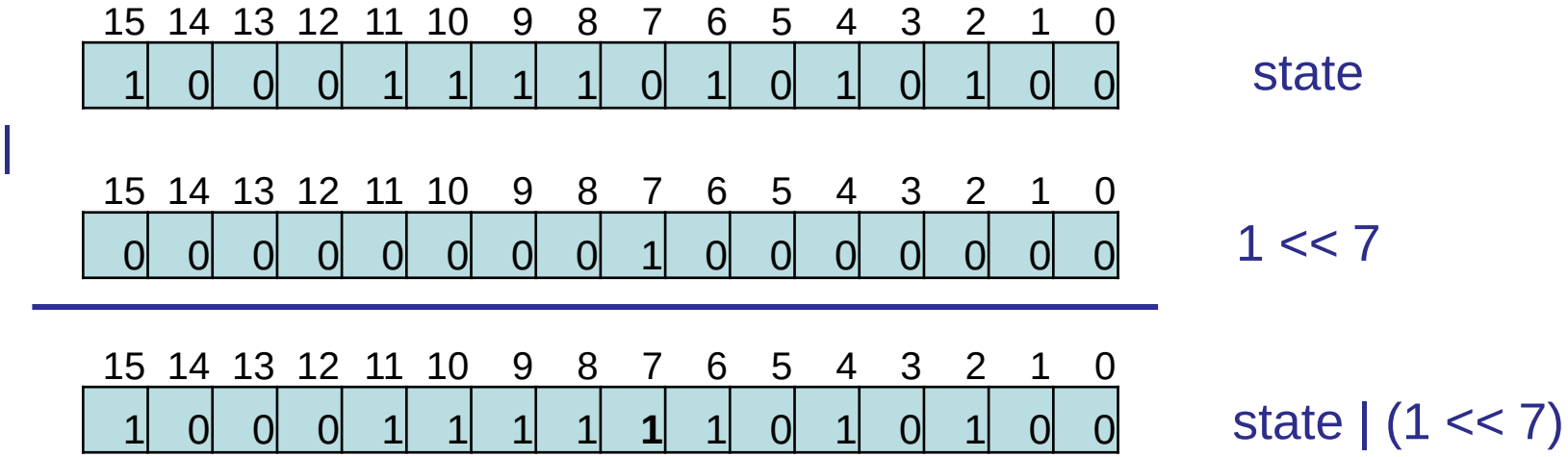
Операции с битовыми полями

- Текущее состояние:



state = 36692

- Включим 7-ю лампочку (7-й бит):



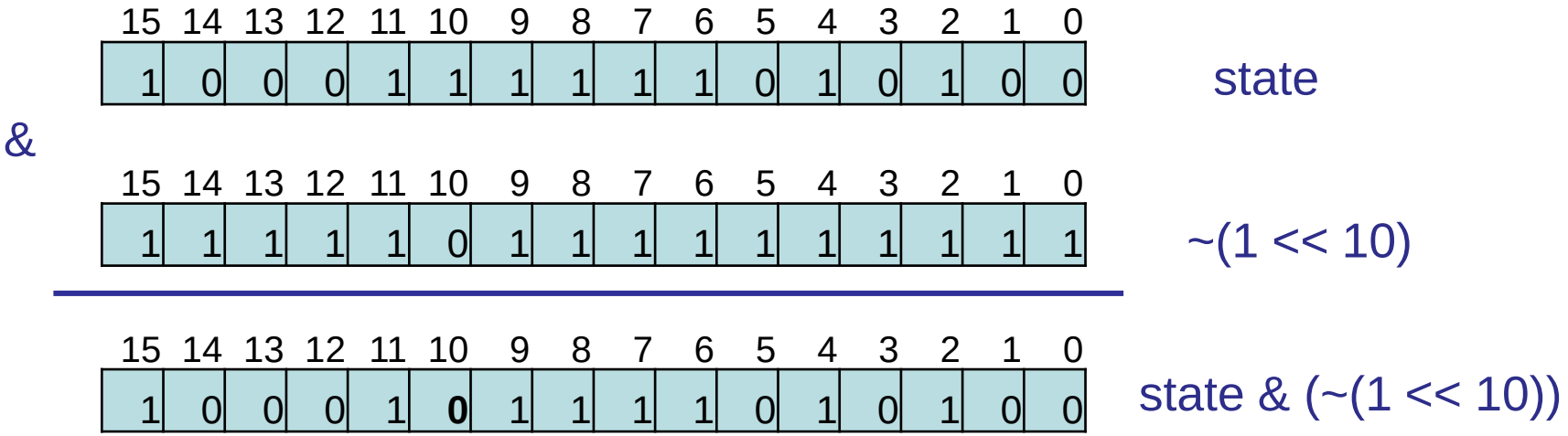
Операции с битовыми полями

- Выключим 10-ю лампочку (10-й бит):

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	0	0	0	1	1	1	1	1	1	0	1	0	1	0	0

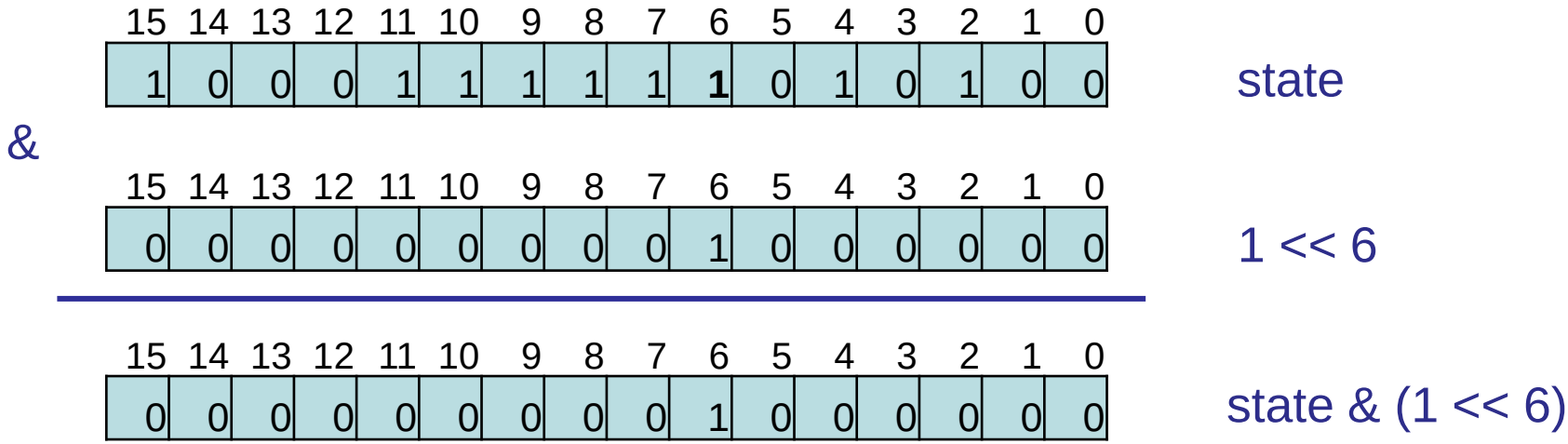
Операции с битовыми полями

- Выключим 10-ю лампочку (10-й бит):



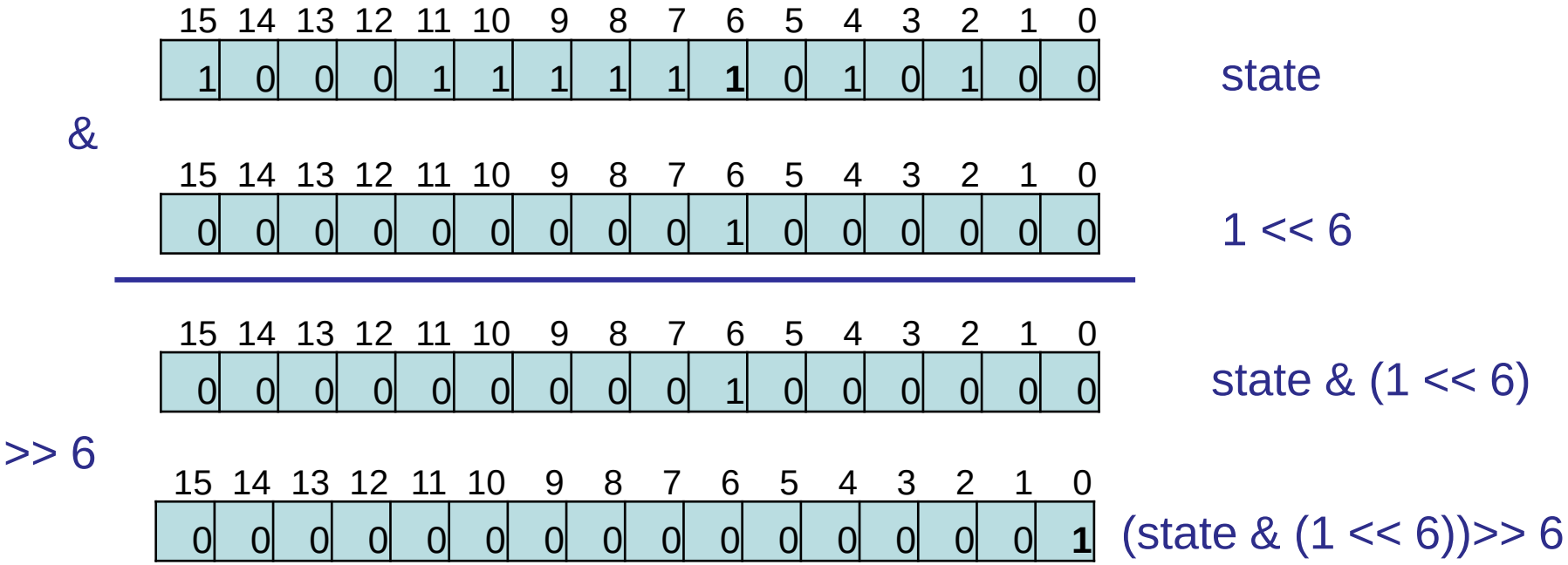
Операции с битовыми полями

- Состояние 6-й лампочки?



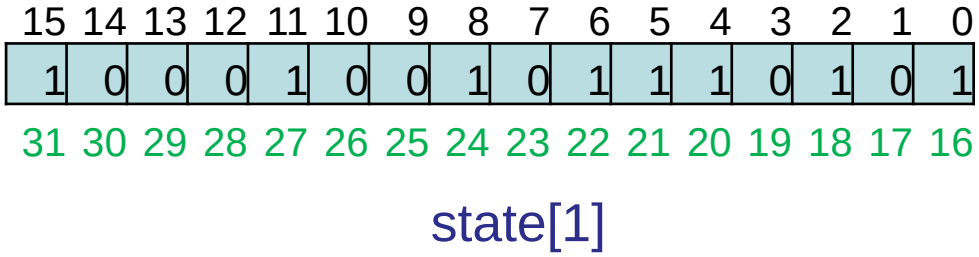
Операции с битовыми полями

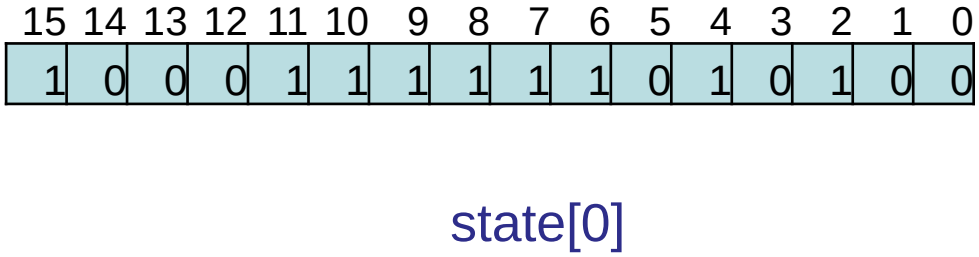
- Состояние 6-й лампочки?



Длинное битовое поле

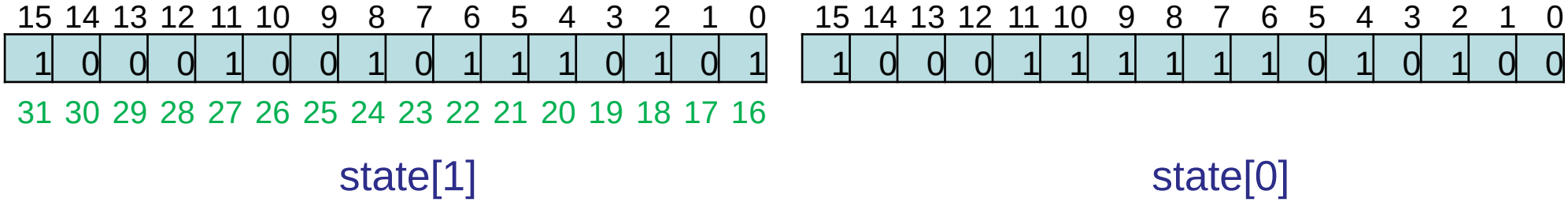
- Что делать, если лампочек больше 16? Больше 32?
  - Массив битовых полей unsigned short state[2]





Длинное битовое поле

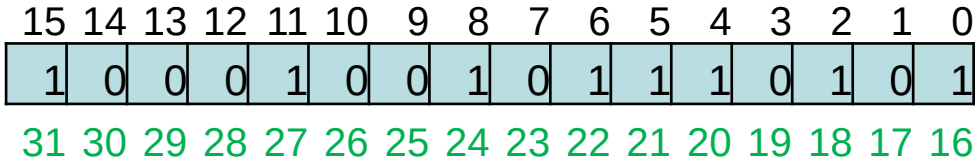
- Что делать, если лампочек больше 16? Больше 32?
  - Массив битовых полей unsigned short state[2]



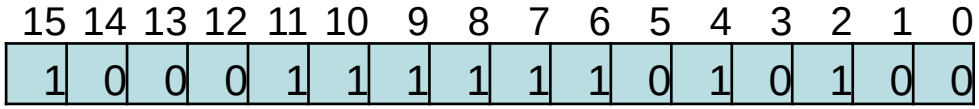
- Битовая маска для лампочки 25:

Длинное битовое поле

- Что делать, если лампочек больше 16? Больше 32?
  - Массив битовых полей unsigned short state[2]

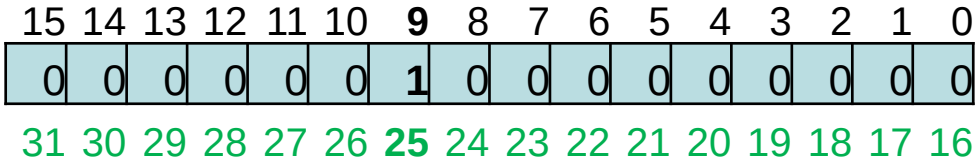


state[1]



state[0]

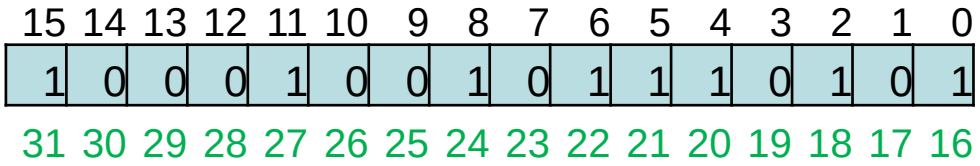
- Битовая маска для лампочки 25:



1 << (25 % 16) = 1 << (25 & 15)

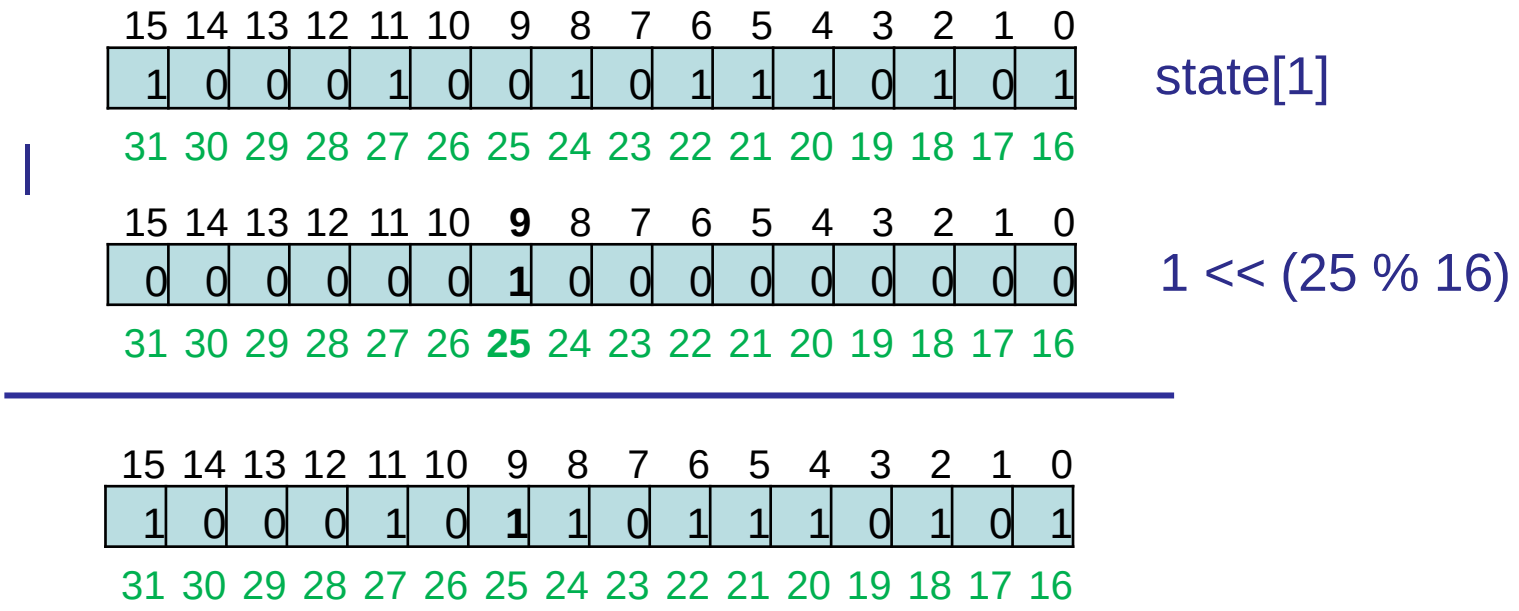
Длинное битовое поле

- Включим бит (лампочку) 25:



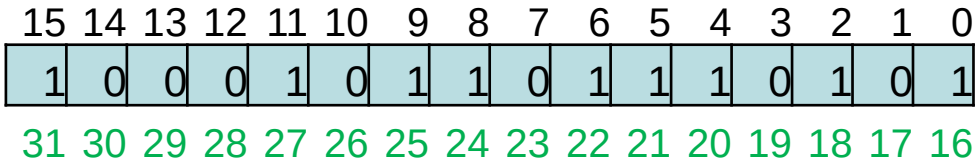
Длинное битовое поле

- Включим бит (лампочку) 25:



Длинное битовое поле

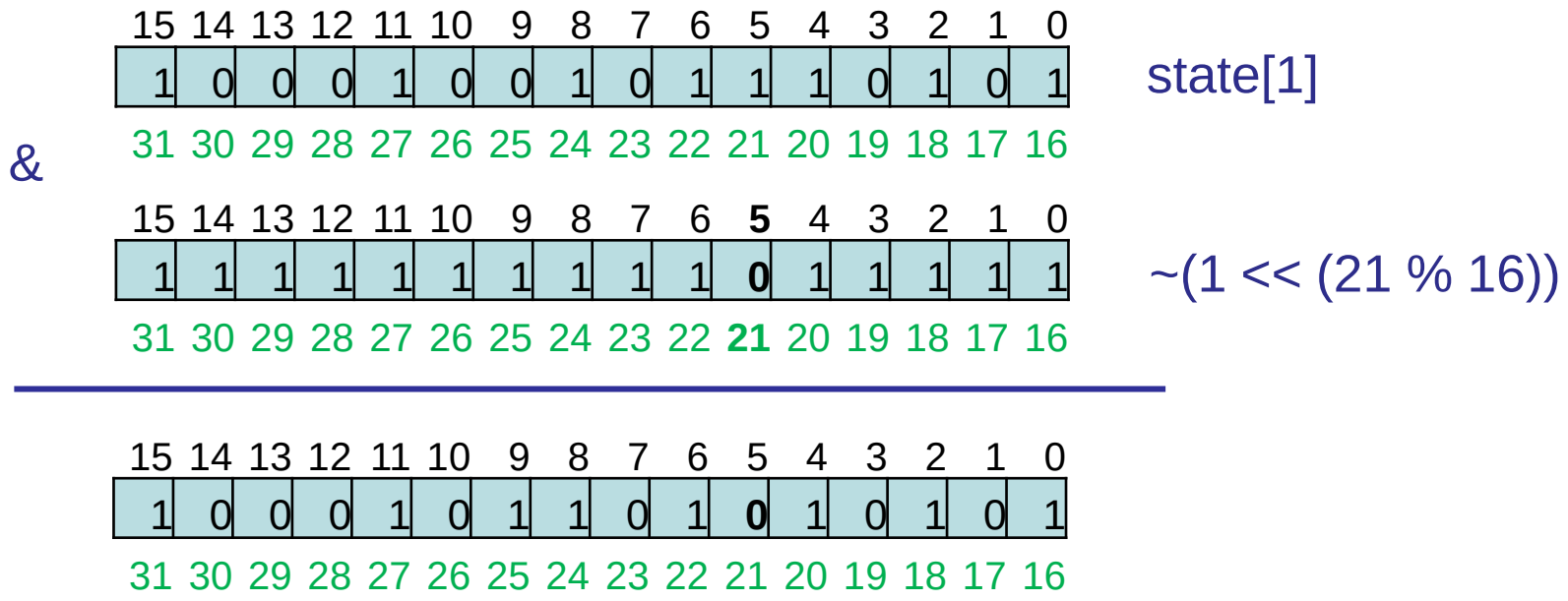
- Выключим бит (лампочку) 21:



state[1]

Длинное битовое поле

- Выключим бит (лампочку) 21:



Программная реализация

- Класс «Битовое поле»
- Поля:
  - Максимальное количество битов
  - Память для представления поля (массив-хранилище)
  - Число элементов в массиве-хранилище
- Методы:
  - Конструкторы
  - Деструктор
  - Установить, очистить бит
  - Узнать значение бита

Программная реализация

- Перегрузка операторов:
  - =, ==, |, &, ~
  - Ввод-вывод

Применение

- Множество – занумерованные элементы
- Реализуем операции над множествами с использованием битовых полей

Программная реализация

- Класс «Множество»
- Поля:
  - Максимальное число элементов множества
  - Память для представления множества – битовое поле
- Методы:
  - Конструкторы
  - Деструктор
  - Вставить / удалить элемент из множества
  - Объединение, пересечение, дополнение множества (через перегрузку)