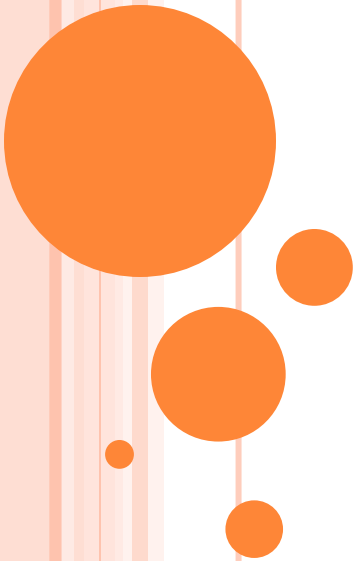

Битовые поля

Лекция №11

Исходные предпосылки

- Необходимость работы с отдельными битами и группами битов нестандартной длины
- Похожесть языка Си на Ассемблер
- Возможность более эффективного использования памяти за счет устранения избыточности
- Наличие в языке Си поразрядных операций
- Возможность организации битовых полей в пользовательских типах данных

Области применения битовых полей

- Включение/выключение светодиодных индикаторов
- Включение/выключение исполнительных механизмов
- Установление/сброс признаков

Поразрядные операции

- операции над каждым битом
  - Поразрядное И &
  - Поразрядное ИЛИ |
  - Поразрядное Исключающее ИЛИ ^
  - Порязрядное НЕ ~

  - У каждой операции – своя таблица истинности

Поразрядное И

- 0 & 0 = 0
- 0 & 1 = 0
- 1 & 0 = 0
- 1 & 1 = 1

Поразрядное ИЛИ

- 0 | 0 = 0
- 0 | 1 = 1
- 1 | 0 = 1
- 1 | 1 = 1

Поразрядное исключающее ИЛИ

- 0 ^ 0 = 0
- 0 ^ 1 = 1
- 1 ^ 0 = 1
- 1 ^ 1 = 0

Операции сдвига

- << - сдвиг влево
  - Переменная << количество разрядов
- >>сдвиг вправо
  - Переменная >> количество разрядов

- a << n соответствует a * 2^n
- a >> n соответствует a / 2^n
  - Ускоряются операции * и /
  - При делении нечетных чисел теряется младший разряд

Установление бита в 1

- int a=0,b=1,n=0;
- a=a | (b<<n);// установление 0-го бита
- n=2;
- a=a | (b<<n);// установление 2-го бита
- n=7;
- a=a | (b<<n);// установление 7-го бита

Сброс бита в 0

- int a=0,b=1,n=0;
- a=a &~(b<<n);// сброс 0-го бита
- n=2;
- a=a &~ (b<<n);// сброс 2-го бита
- n=7;
- a=a &~ (b<<n);// сброс 7-го бита

Изменение значения бита на противоположное

- int a=0,b=1,n=0;
- a=a ^ (b<<n);// инверсия 0-го бита
- n=2;
- a=a ^ (b<<n);// установление 2-го бита
- n=7;
- a=a ^ (b<<n);// установление 7-го бита

- //Мигание бита
- int i;
- for(i=0; i<8;i++)
- a=a^(b<<n)

Проверка состояния бита

- int a=0,b=1,n=0,c=0;
- if(a & (b<<n)) //если бит равен 1
- c=1;

Пример управления светодиодами

```
#define RED 0x1
#define YELLOW 0x2
#define GREEN 0x4

int leds=0;
leds=leds|RED;
leds=leds&~RED;
leds=leds|YELLOW;
leds=leds&~YELLOW;
leds=leds|GREEN;
leds=leds&~GREEN;
```

Битовые поля

- структуры, которые объединяют поля нестандартной длины (по несколько битов)
- Формат битовых полей

```
struct имя_структуры
{
unsigned имя_поля1: количество-битов1;
unsigned имя_поля2: количество-битов2;
.....
unsigned имя_поляN: количество-битовN;
};
//используется тип unsigned, чтобы все биты были информационными
```

Пример битовых полей

```
struct bit_field
{unsigned power : 1;//два состояния
unsigned  mode: 2; // ? состояния
};

bit_field c;
c.power=1;
c.mode=0x3;
```

Выводы

- Битовые поля позволяют гибко управлять режимами работы вычислительных устройств
- Битовые поля позволяют компактно представлять информацию