Bit Operations

Bit Operations Topics

- Bitwise Operators
- Bit Fields

Bitwise Operators

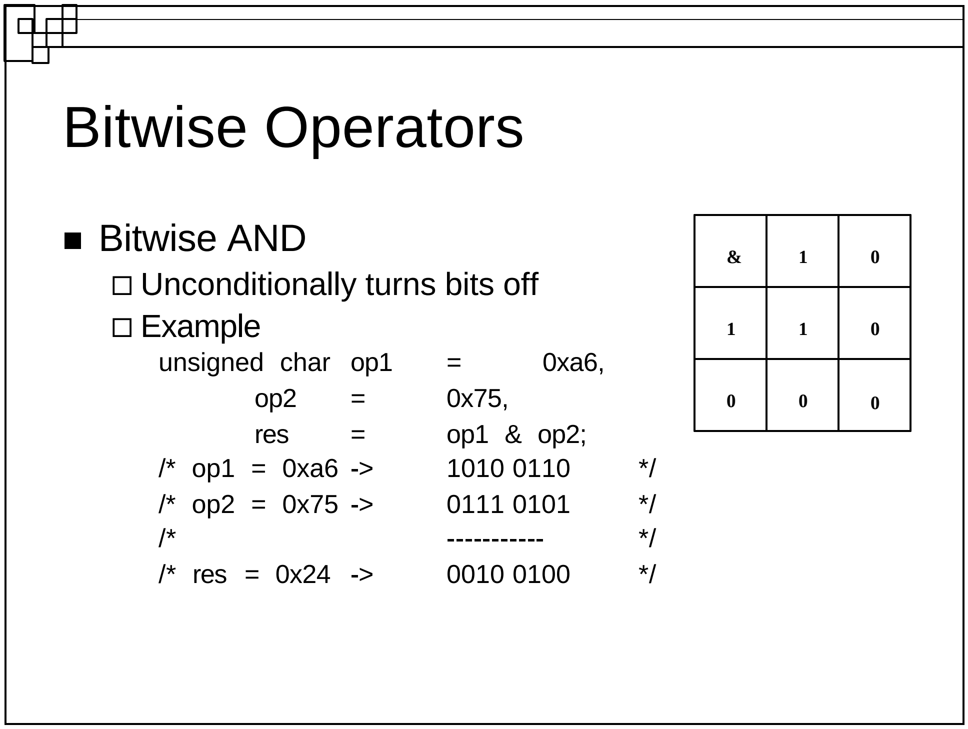
|	logical Inclusive OR
&	logical AND
~	logical COMPLEMENT
^	logical exclusive OR
>>	right shift
<<	left shift

Bitwise Operators

- Bitwise Inclusive OR
  - □ Unconditionally turn bits on
  - □ Example

```
unsigned  char   op1       =          0x36,
            op2       =          0x95,
            res        =          op1  |  op2;
/*  op1  =  0x36  ->          0011 0110         */
/*  op2  =  0x95  ->          1001 0101         */
/*                                 -----------          */
/*  res  =  0xb7   ->          1011 0111         */
```

|	1	0
1	1	1
0	1	0

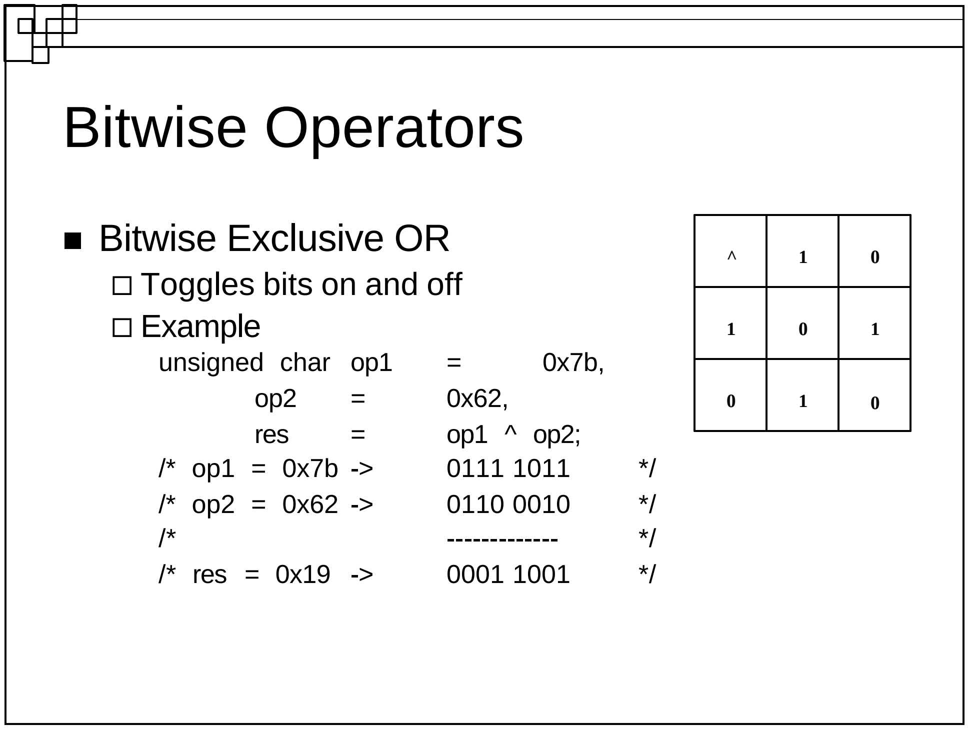


Bitwise Operators

- Bitwise AND
  - □ Unconditionally turns bits off
  - □ Example

```
unsigned  char  op1      =          0xa6,
          op2      =          0x75,
          res      =          op1  &  op2;
/*  op1  =  0xa6  ->        1010 0110        */
/*  op2  =  0x75  ->        0111 0101        */
/*                          -----------      */
/*  res  =  0x24  ->        0010 0100        */
```

&	1	0
1	1	0
0	0	0

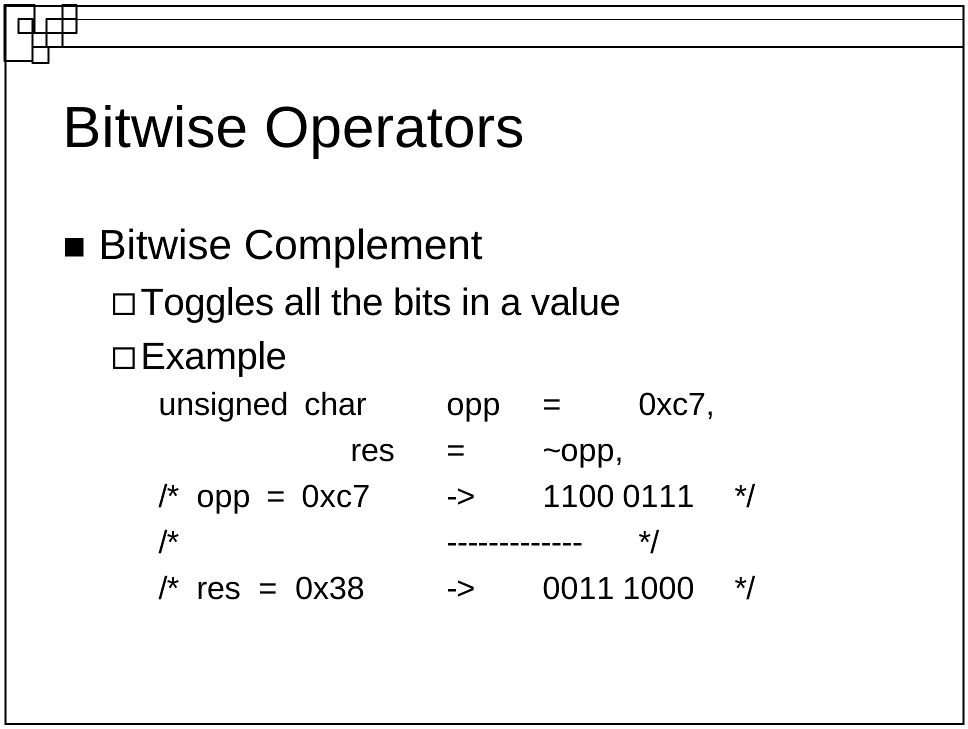


Bitwise Operators

- Bitwise Exclusive OR
  - □ Toggles bits on and off
  - □ Example

```
unsigned  char  op1      =          0x7b,
          op2      =          0x62,
          res      =          op1  ^  op2;
/*  op1  =  0x7b  ->        0111 1011        */
/*  op2  =  0x62  ->        0110 0010        */
/*                          -------------    */
/*  res  =  0x19  ->        0001 1001        */
```

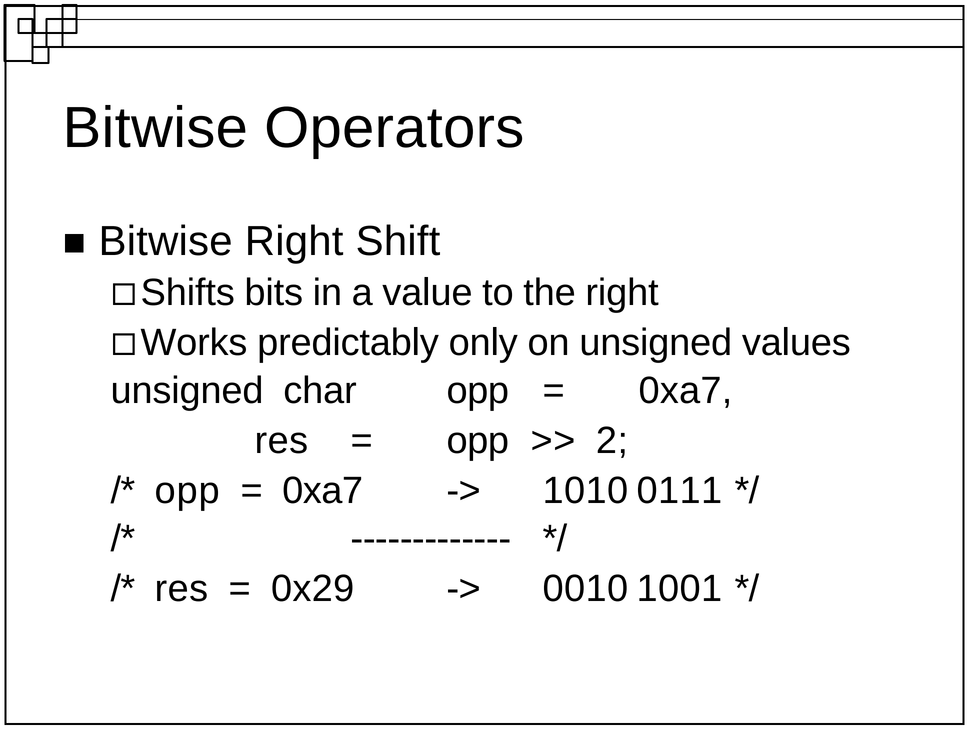
^	1	0
1	0	1
0	1	0



Bitwise Operators

- Bitwise Complement
  - □ Toggles all the bits in a value
  - □ Example

```
unsigned  char        opp     =        0xc7,
                 res       =        ~opp,
/*  opp  =  0xc7        ->       1100 0111     */
/*                         -------------      */
/*  res  =  0x38        ->       0011 1000     */
```



Bitwise Operators

- Bitwise Right Shift
  - □ Shifts bits in a value to the right
  - □ Works predictably only on unsigned values

```
unsigned  char        opp    =       0xa7,
             res    =       opp  >>  2;
/*  opp  =  0xa7        ->      1010 0111  */
/*                 -------------   */
/*  res  =  0x29        ->      0010 1001  */
```

Bitwise Operators

- Bitwise Left Shift
  - Shifts bits in a value to the left

```
unsigned  char        opp   =      0xa7,
                res    =      opp  <<  2;
/*  opp  =  0xa7        ->      1010 0111  */
/*                   -------------   */
/*  res  =  0x29        ->      1001 1100  */
```

Bitwise Operators

- Example
  - http://faculty.washington.edu/sproedp/advc/csamples/less22-1.c.html

Bitwise Fields

- Syntax
  - struct S

    ```
    struct S
    {
      unsigned two_bits : 2;
      unsigned four_bits : 4;
    };
    ```
  - two_bits occupies 2 bits and four_bits occupies 4 bits
  - may be padded
  - not normally portable

Bit Fields

- http://faculty.washington.edu/sproedp/advc/csamples/less22-2.c.html