

slide 3

Use Unsigned Data Types for Bit Operations

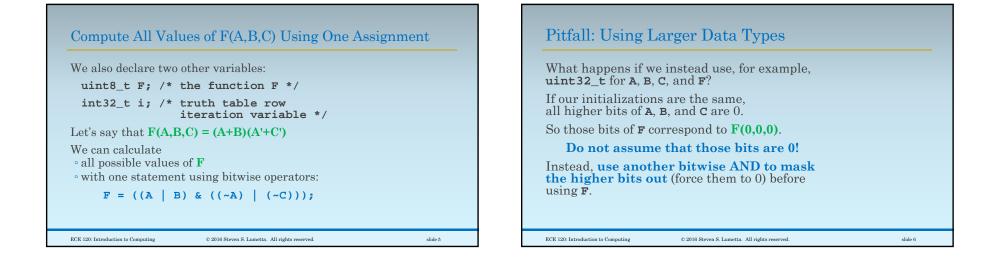
When working with bits, we use unsigned data types.
This choice is particularly important

when shifts are necessary,
but it's a good habit for all bit operations.

Our function has three input variables,

so we need eight bits
to represent the eight rows of the truth table.

Assign Truth Ta	ble Row Values to Input Bit	w Values to Input Bits	
We first declare the th	ree input variables:*		
uint8_t A = 0xF0;	/* A is 11110000 */		
uint8_t B = 0xCC;	/* B is 11001100 */		
uint8_t C = 0xAA;	/* C is 10101010 */		
Notice that each triple of bits (vertically) • forms a unique combination of input values • from 111 for bit 7 down to 000 for bit 0 .			
	name as an unsigned char but see the notes for details.		
ECE 120: Introduction to Computing	$\mathbb C$ 2016 Steven S. Lumetta. All rights reserved.	slide 4	



We Print One Row at a Time	
Now we can print the header for our truth table.	
<pre>printf ("A B C F\n");</pre>	
printf ("+\n");	
To print the rows, we use a single printf per row:	
for $(i = 0; 8 > i; i = i + 1)$ {	
/* printf goes here */	
}	
ECE 120: Introduction to Computing © 2016 Steven S. Lumetta. All rights reserved.	slide 7

The printf Merely Extracts Bits of F				
The input variables • are extracted from the loop index i , and • F (A , B , C) is the i th bit of F .				
printf ("%c %c %c %c\n",				
'0' + (0 != (i & 4)), /* A */				
'0' + (0 != (i & 2)), /* B */				
'0' + (0 != (i & 1)), /* C */				
'0' + (0 != (F & (1 << i))));				
ECE 120: Introduction to Computing © 2016 Steven S. Lumetta. All rights reserved.	slide 8			