Bonus Lecture: Bit Manipulation in C

EECS 388 – Fall 2022

- Binary to Hex to Decimal conversion!

- 1 hex symbol is 4 binary bits
- 1 hex symbol represents decimal values from 0 to 15
  - 0 – F
- 0xAB is 8 bits (1 bytes)
- 0xA is 4 bits (not one bytes!)

- Two's Complement Numbers

e.g., a 4-bit signed variable:

Positive Values		**Negative Values**	
Decimal	Binary	Decimal	Binary
7	0 000	-1	1 111
6	0 001	-2	1 110
5	0 010	-3	1 101
4	0 011	-4	1 100
3	0 100	-5	1 011
2	0 101	-6	1 010
1	0 110	-7	1 001
0	0 111		

Bitwise Operation Facts

XOR

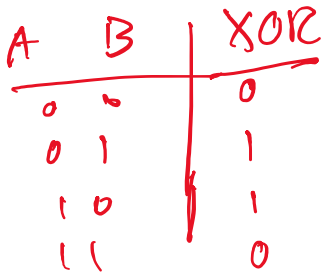
- var ^ 0s =
- var ^ 1s =
- var ^ var =

AND

- var & 0s =
- var & 1s =
- var & var =

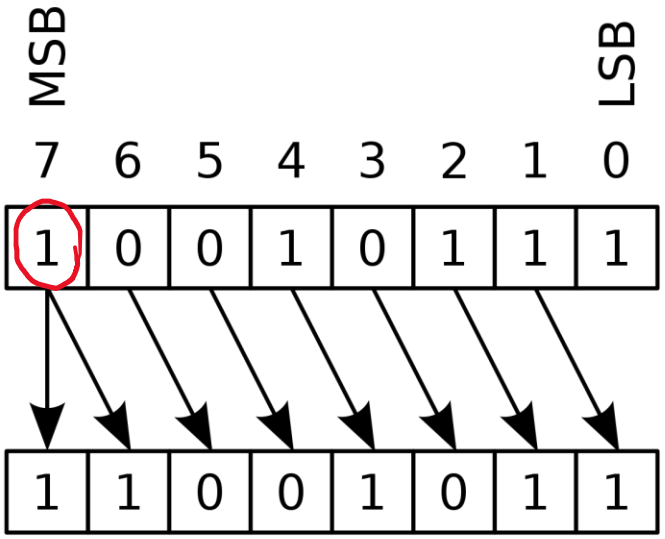
OR

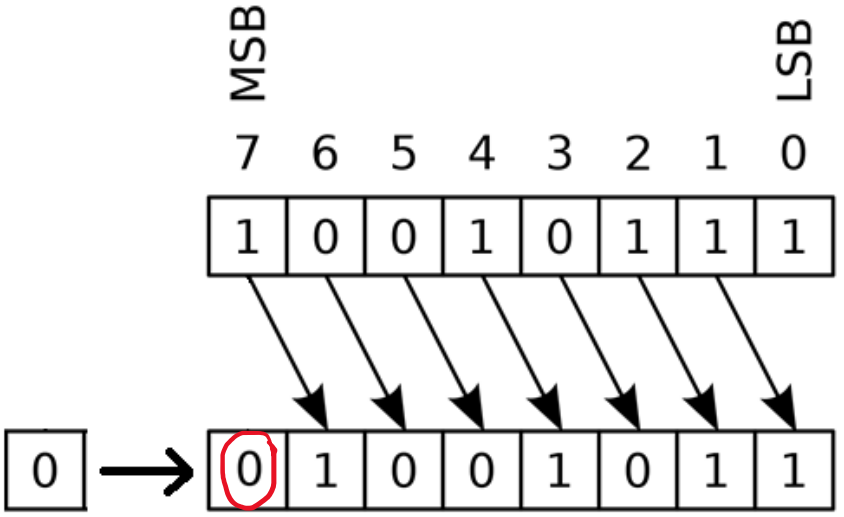
- var | 0s =
- var | 1s =
- var | var =



- Arithmetic vs. logical shift

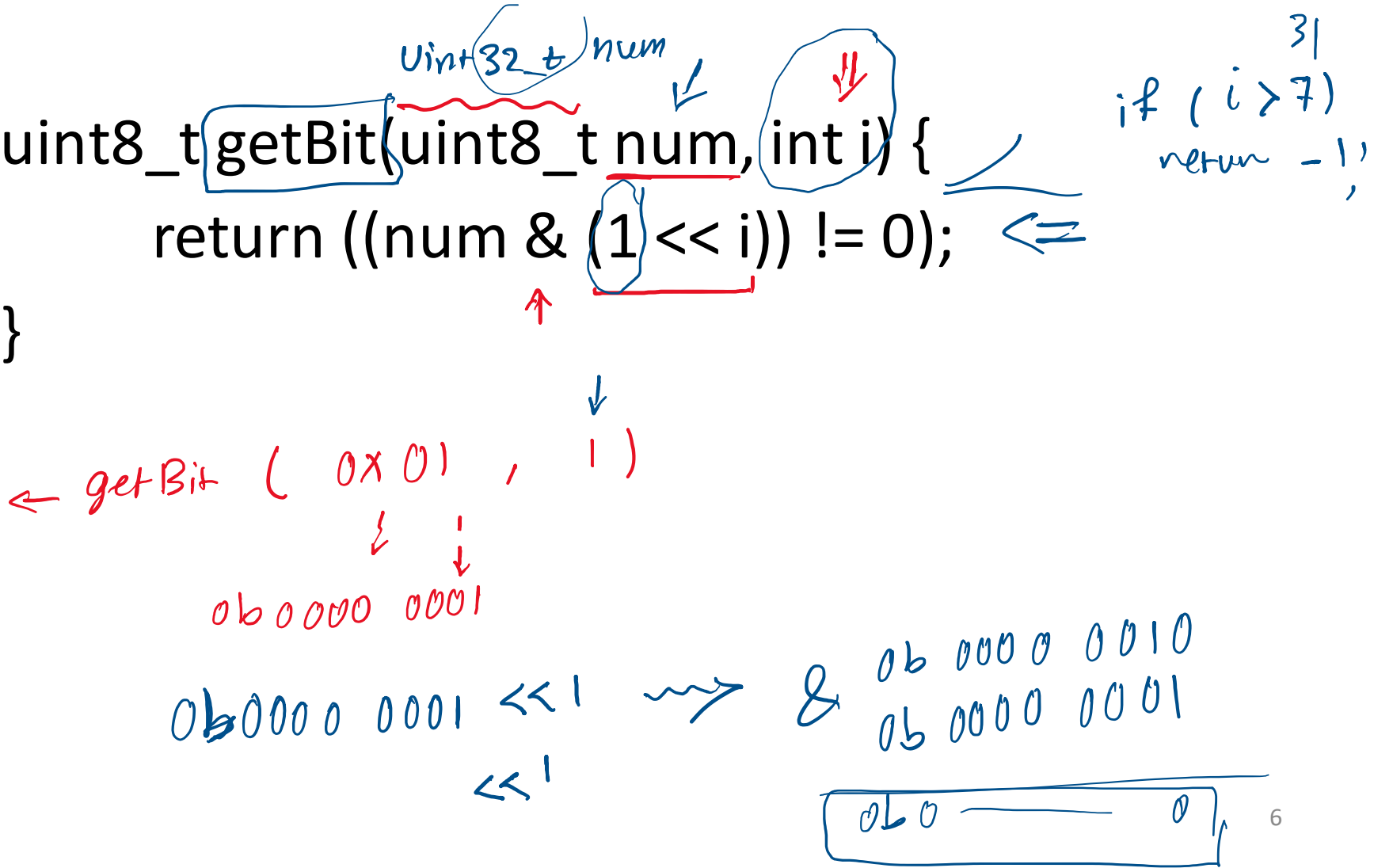
Arithmetic





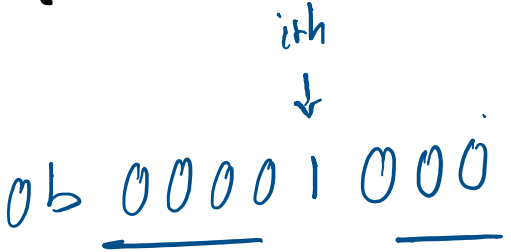
Get bit: return 0 if the bit “i” is zero, otherwise return 1

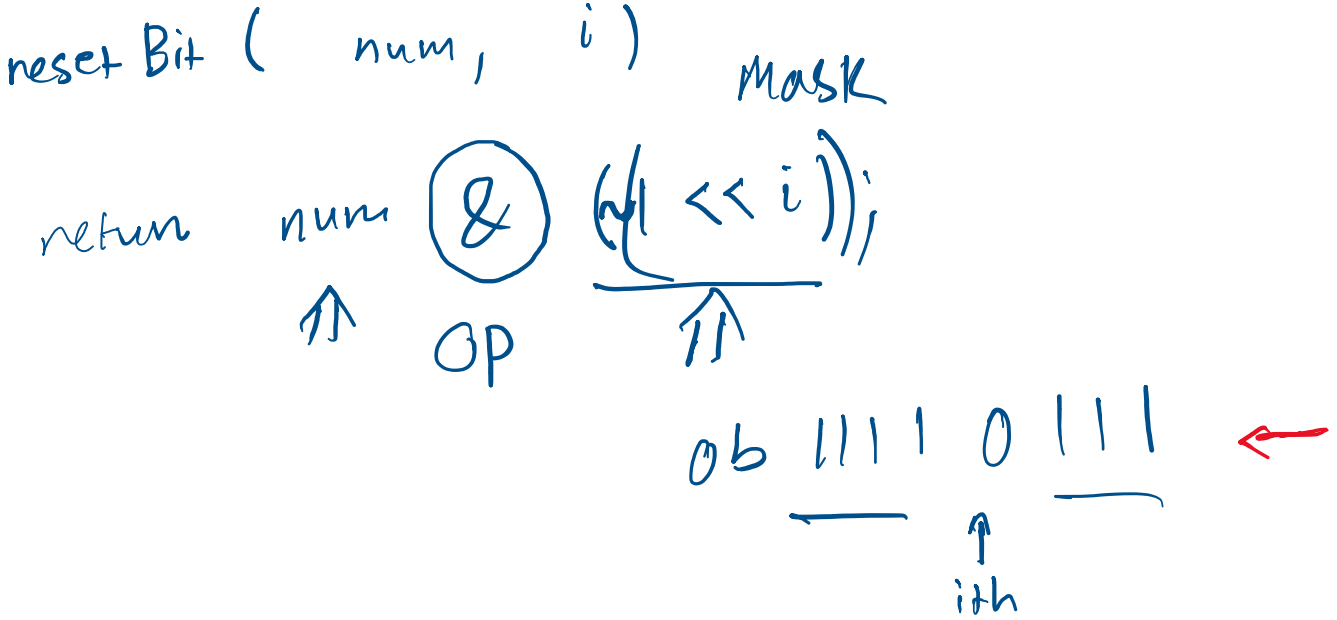
```
uint8_t getBit(uint8_t num, int i) {
        return ((num & (1 << i)) != 0);
}
```



Set bit: set bit "i" to 1

```
uint8_t setBit(uint8_t num, int i) {
        return num | (1 << i);
}
```





Reset/clear bit: set bit “i” to 0

```
uint8_t clearBit(uint8_t num, int i) {
        uint8_t mask = ~(1 << i);
        return num & mask;
}
```

Update bit: set bit “i” to 0 if flag is 0, otherwise set it to 1

```
uint8_t updateBit(uint8_t num, int i, int flag) {
        uint8_t value = flag ? 0x01: 0x0;
        uint8_t mask = ~(1 << i);
        return (num & mask) | (value << i);
}
```

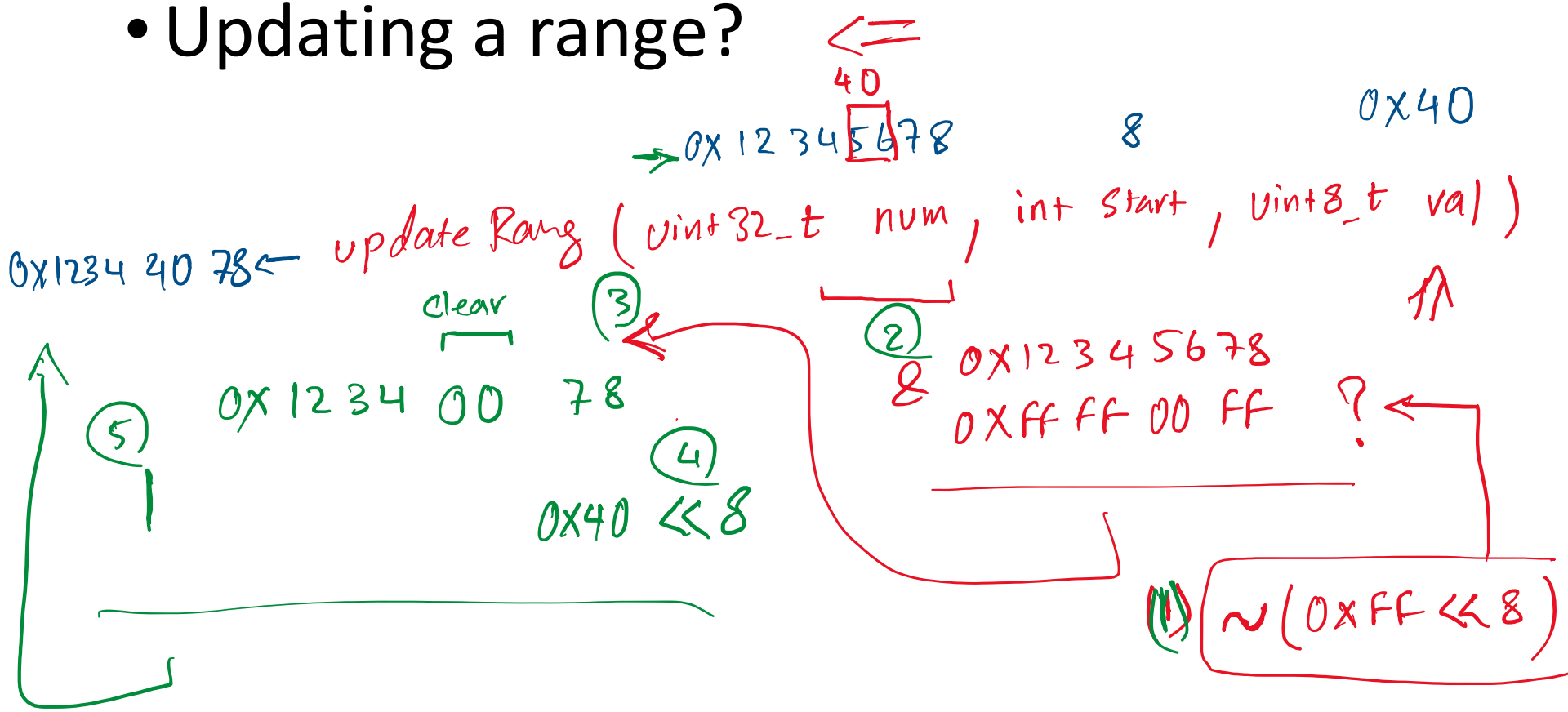
reset bit i

set it to either 0 or 1

- Setting a range?

- Clearing a range?

- Updating a range?



- Returning a range?