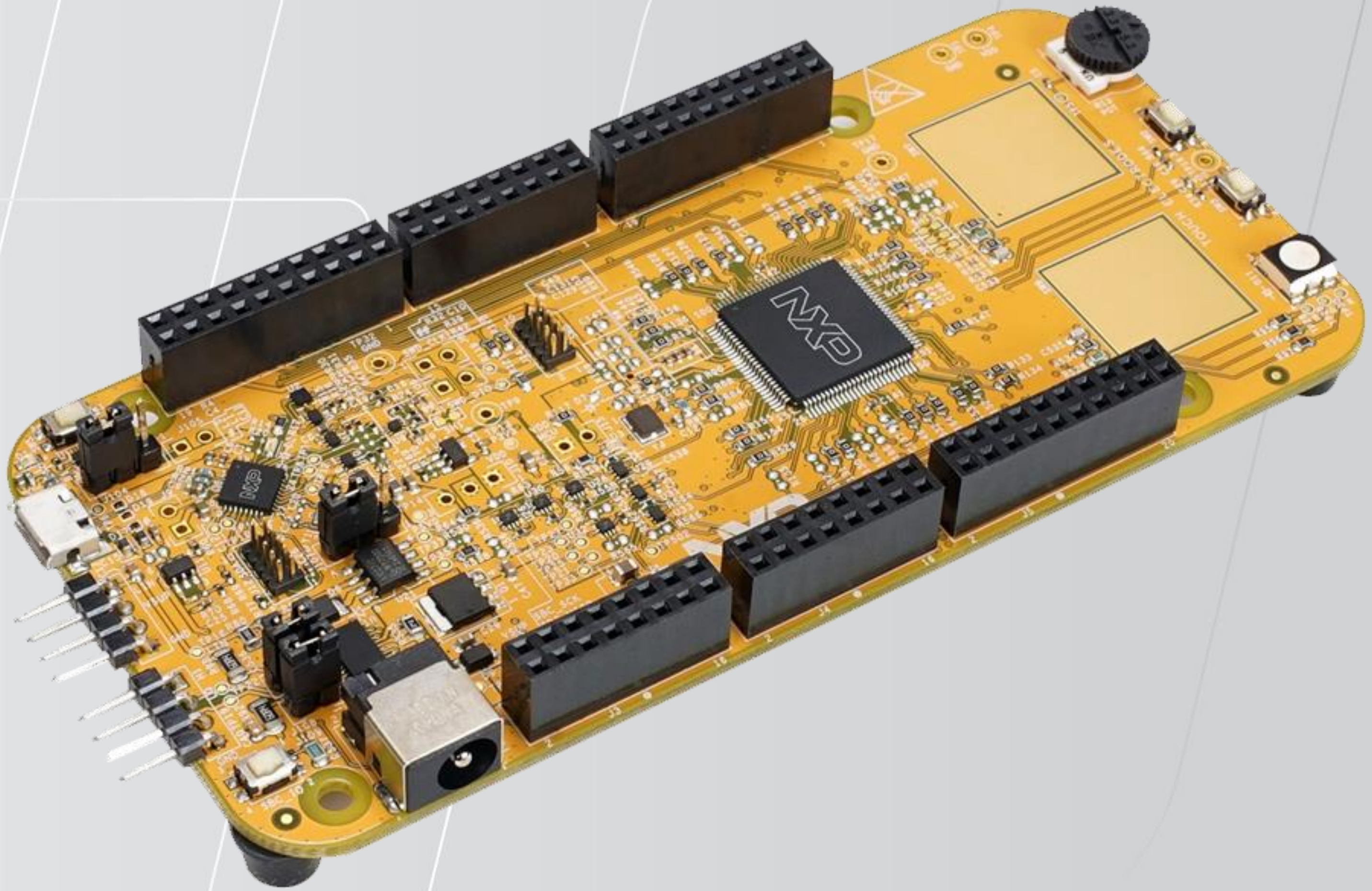
Bitwise Operators & Bit Masking



10 Oct 2022

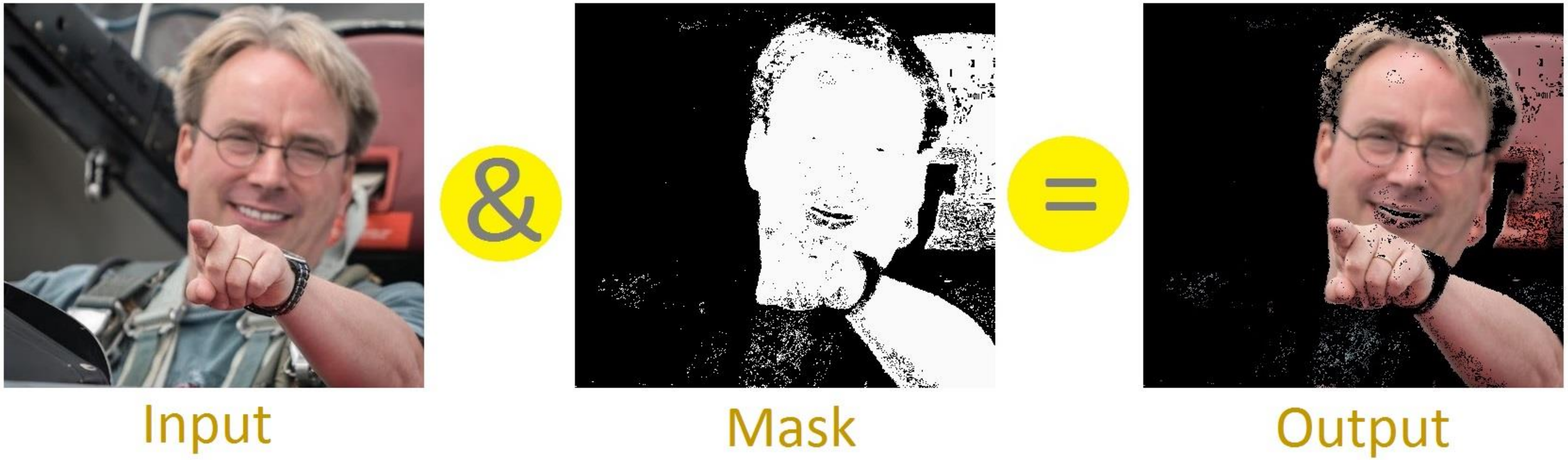
OUTLINE

1. Bitwise operators

No.	Name	Symbol	Usage	Meaning
1	Bitwise And	&	a&b	Returns 1 if the both the bits are 1
2	Bitwise Or	|	a|b	Returns 1 if one of the bits are 1
3	Bitwise Not	~	~a	Returns the complement of a bit
4	Bitwise Xor	^	a^b	Returns 0 if both the bits are same
5	Bitwise Left shift	<<	a<<n	Shifts a towards left by n digits
6	Bitwise Right shift	>>	a>>n	Shifts a towards right by n digits

2. Bit Masking

- Imposing mask over bits.



2.1. Set Bits

Ex: Setting bit 6 to 1

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
1	0	1	1	1	1	1	0	← Register
x	x	x	x	x	x	x	x	← Mask
1	1	1	1	1	1	1	0	← Result

Bitwise Or (|)

Register = Register | Mask ➡ Register |= Mask ➡ Register |= 0b01000000 ➡ Register |= (1<<6)

Register |= (value <<n)

2.1. Set Bits

Ex1: Setting bit 0 to 1

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
	1	0	1	1	1	1	1	0	← Register
Bitwise Or (|)	0	0	0	0	0	0	0	1	← Mask
	1	0	1	1	1	1	1	1	← Result

```
Register |= (1<<0)
```

2.1. Set Bits

Ex2: Setting bit 6 and bit 5 to 1

Bitwise Or (|)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
1	0	0	1	1	1	1	0	← Register
0	1	1	0	0	0	0	0	← Mask
1	1	1	1	1	1	1	0	← Result

```
Register |= (3<<5)
```

```
Register |= ((1<<5)|(1<<6))
```

2.2. Clear Bits

Ex: Clearing bit 5 to 0

Bitwise And (&)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
	1	0	1	1	1	1	1	0	← Register
	x	x	x	x	x	x	x	x	← Mask
	1	0	0	1	1	1	1	0	← Result

Register = Register & Mask → Register &= Mask → Register &= 0b11011111 → Register &= ~(1<<5)

Register &= ~(value<<n)

2.2. Clear Bits

Ex1: Clearing bit 7 to 0

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
	1	0	1	1	1	1	1	0	← Register
Bitwise And (&)									
	0	1	1	1	1	1	1	1	← Mask
	0	0	1	1	1	1	1	0	← Result

Register &= ~(1<<7)

2.2. Clear Bits

Ex1: Clearing bit 3 and bit 2 to 0

Bitwise And (&)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
1	0	1	1	1	1	1	0	← Register
1	1	1	1	0	0	1	1	← Mask
1	0	1	1	0	0	1	0	← Result

```
Register &= ~(3<<2)
```

```
Register &= ~((1<<2)|(1<<3))
```

2.2. Clear Bits

Ex2: Clearing all bits to 0

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
	1	0	1	1	1	1	1	0	← Register
Bitwise And (&)									
	0	0	0	0	0	0	0	0	← Mask
	0	0	0	0	0	0	0	0	← Result

Register = 0

2.3. Toggle Bits

Ex: Toggling bit 5

Bitwise Xor (^)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
1	0	x(1)	1	1	1	1	0	← Register
x	x	x	x	x	x	x	x	← Mask
1	0	**0**	1	1	1	1	0	← Result

Register = Register ^ Mask ➡ Register ^= Mask ➡ Register ^= 0b00100000 ➡ Register ^= (1<<5)

Register ^= (value<<n)

2.3. Toggle Bits

Ex: Toggling bit 5 and bit 6

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
	1	x(0)	x(1)	1	1	1	1	0	← Register
Bitwise Xor (^)									
	0	1	1	0	0	0	0	0	← Mask
	1	1	0	1	1	1	1	0	← Result

```
Register ^= (3<<5)
```

```
Register ^= ((1<<5)|(1<<6))
```

2.4. Check Bits

Ex: Find value of bit 5

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
1	0	1	1	1	1	1	0	← Register
x	x	x	x	x	x	x	x	← Mask
0	0	1	0	0	0	0	0	← Result

Bitwise And (&)

Result = Register & Mask ⇒ Result = Register & 0b00100000 ⇒ Result = Register & (1<<5)

Result = Register & (value<<n)

Thank you