# 1.1 Number systems

## Logical binary shifts

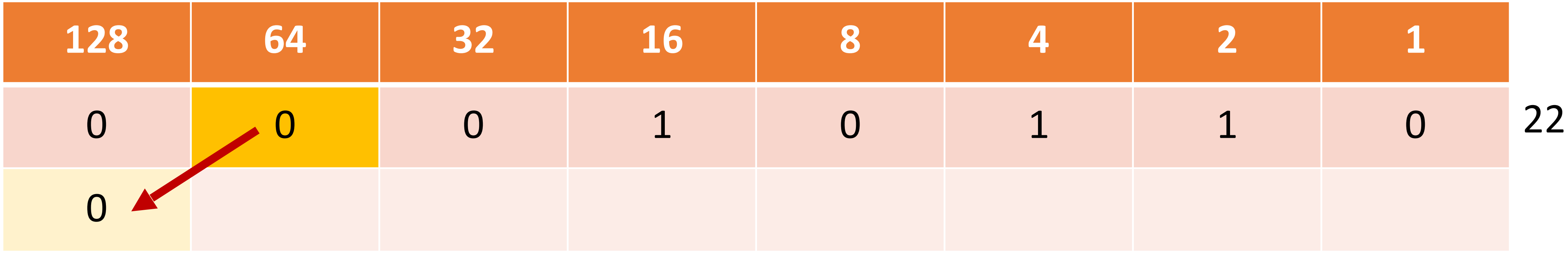
# Logical binary left shift

| 128 | 64 | 32 | 16 | 8 | 4 | 2 | 1 | |
|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 22 |
| | | | | | | | | |

## Logical binary left shift

| 128 | 64 | 32 | 16 | 8 | 4 | 2 | 1 | |
|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 22 |
| | | | | | | | | |

# Logical binary left shift

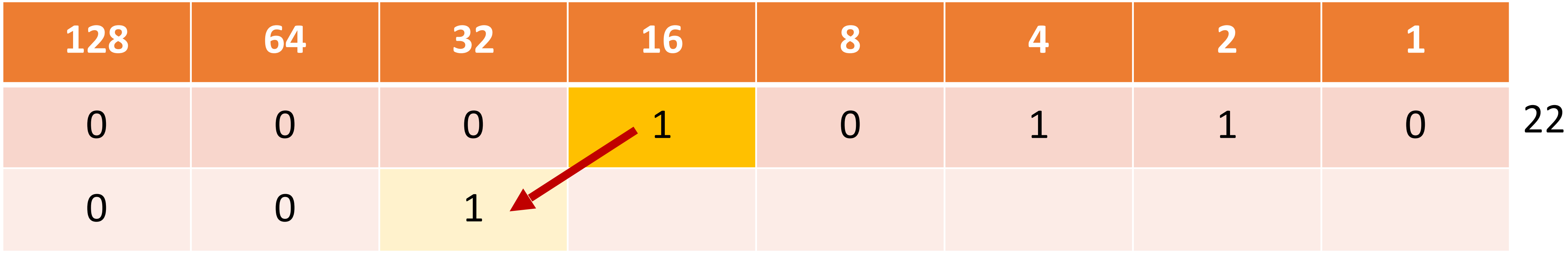


| 128 | 64 | 32 | 16 | 8 | 4 | 2 | 1 | |
|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 22 |
| 0 | | | | | | | | |

## Logical binary left shift

| 128 | 64 | 32 | 16 | 8 | 4 | 2 | 1 | |
|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 22 |
| 0 | 0 | | | | | | | |

# Logical binary left shift



| 128 | 64 | 32 | 16 | 8 | 4 | 2 | 1 | |
|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 22 |
| 0 | 0 | 1 | | | | | | |

# Logical binary left shift

| 128 | 64 | 32 | 16 | 8 | 4 | 2 | 1 | |
|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 22 |
| 0 | 0 | 1 | 0 | | | | | |

# Logical binary left shift

| 128 | 64 | 32 | 16 | 8 | 4 | 2 | 1 | |
|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 22 |
| 0 | 0 | 1 | 0 | 1 | | | | |

# Logical binary left shift

| 128 | 64 | 32 | 16 | 8 | 4 | 2 | 1 | |
|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 22 |
| 0 | 0 | 1 | 0 | 1 | 1 | | | |

# Logical binary left shift

| 128 | 64 | 32 | 16 | 8 | 4 | 2 | 1 | |
|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 22 |
| 0 | 0 | 1 | 0 | 1 | 1 | 0 | | |

## Logical binary left shift

| 128 | 64 | 32 | 16 | 8 | 4 | 2 | 1 | |
|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 22 |
| 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | |

## Logical binary left shift

| 128 | 64 | 32 | 16 | 8 | 4 | 2 | 1 | |
|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 22 |
| 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 44 |
| | | 32 + | | 8 + | 4 | | | |

By performing a 1-bit left shift, 22 has become 44.

We have multiplied the original number by 2.

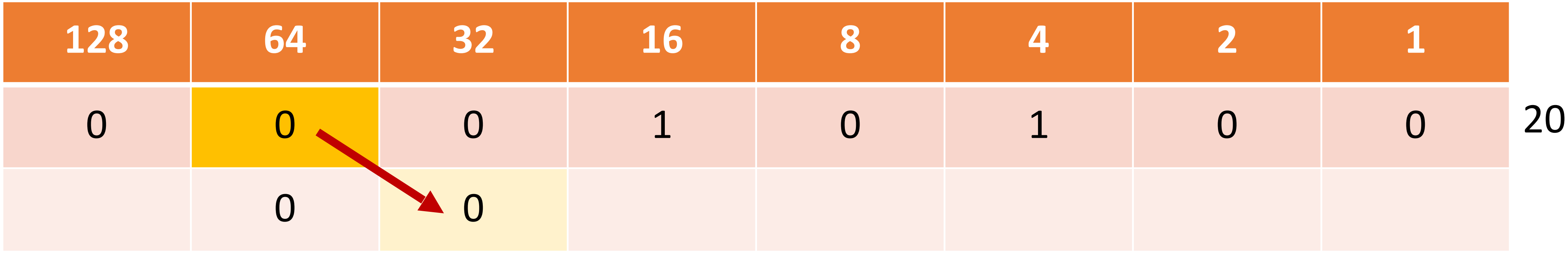
# Logical binary right shift

| 128 | 64 | 32 | 16 | 8 | 4 | 2 | 1 | |
|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 20 |
| | | | | | | | | |

## Logical binary right shift

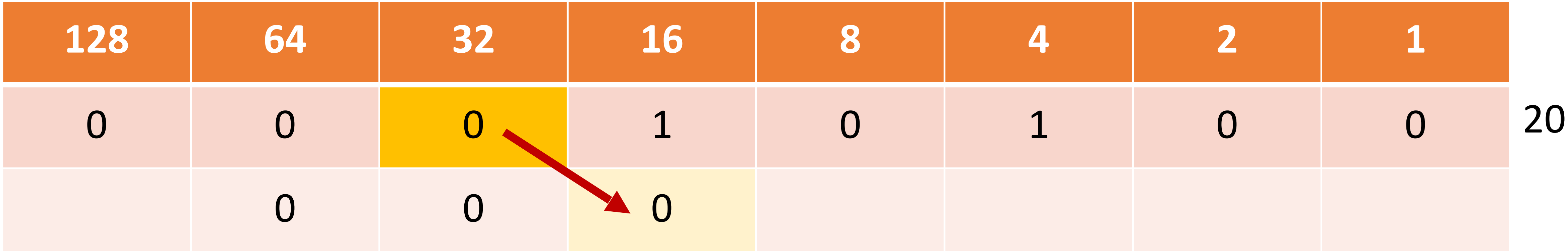
| 128 | 64 | 32 | 16 | 8 | 4 | 2 | 1 | |
|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 20 |
| | 0 | | | | | | | |

# Logical binary right shift



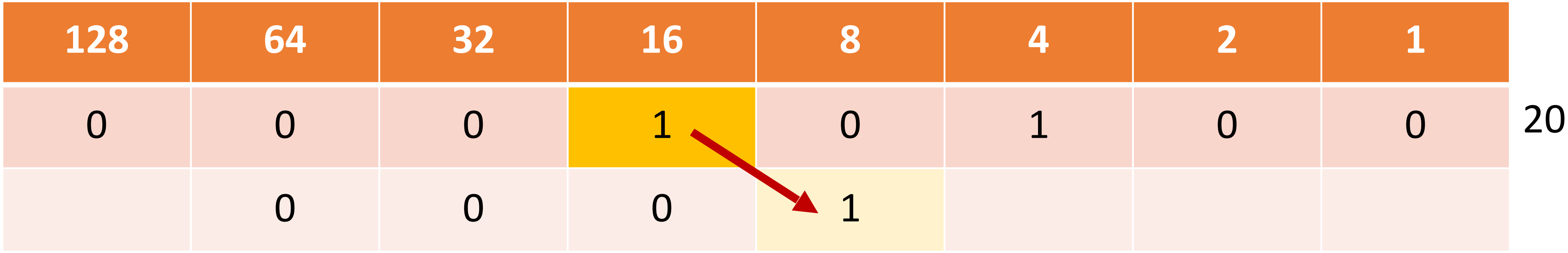
| 128 | 64 | 32 | 16 | 8 | 4 | 2 | 1 | |
|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 20 |
| | 0 | 0 | | | | | | |

# Logical binary right shift



| 128 | 64 | 32 | 16 | 8 | 4 | 2 | 1 | |
|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 20 |
| | 0 | 0 | 0 | | | | | |

# Logical binary right shift



| 128 | 64 | 32 | 16 | 8 | 4 | 2 | 1 | |
|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 20 |
| | 0 | 0 | 0 | 1 | | | | |

# Logical binary right shift

| 128 | 64 | 32 | 16 | 8 | 4 | 2 | 1 | |
|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 20 |
| | 0 | 0 | 0 | 1 | 0 | | | |

# Logical binary right shift

| 128 | 64 | 32 | 16 | 8 | 4 | 2 | 1 | |
|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 20 |
| | 0 | 0 | 0 | 1 | 0 | 1 | | |

# Logical binary right shift

| 128 | 64 | 32 | 16 | 8 | 4 | 2 | 1 | |
|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 20 |
| | 0 | 0 | 0 | 1 | 0 | 1 | 0 | |

# Logical binary right shift

| 128 | 64 | 32 | 16 | 8 | 4 | 2 | 1 | |
|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 20 |
| | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 |

# Logical binary right shift

| 128 | 64 | 32 | 16 | 8 | 4 | 2 | 1 | |
|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 20 |
| 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | |

## Logical binary right shift

| 128 | 64 | 32 | 16 | 8 | 4 | 2 | 1 | |
|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 20 |
| 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 10 |
| | | | | 8 + | | 2 | | |

By performing a 1-bit right shift, 20 has become 10.

We have divided the original number by 2.

# Logical binary shifts

- A **left logical binary shift** of one position:
    - Moves each bit to the left by one.
    - Fills the vacant least significant bit (LSB) with zero and discards the most significant bit (MSB).
- A **right logical binary shift** of one position:
    - Moves each bit to the right by one.
    - Discards the least significant bit and fills the vacant MSB with zero.
- One use of logical binary shifts is to multiply and divide unsigned binary integers by powers of two.
- We can shift by more than one position at a time:
    - A left logical binary shift of one position would covert the number 22 to 44 (x2).
    - A left logical binary shift of two positions would convert the number 22 to 88 (x4).

# 1.1 Number systems | Logical binary shifts