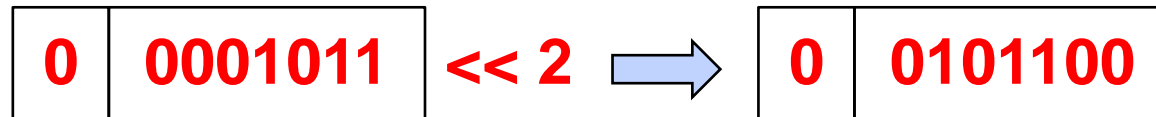
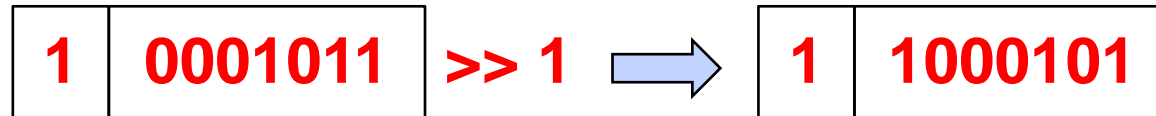


Die Schiebeoperatoren

- Die Schiebeoperatoren liefern als Ergebnis eine Verschiebung der Bits des ersten Operanden um die Anzahl Stellen, die mit dem zweiten Operanden angegeben wird (ohne den ersten Operanden zu verändern!).

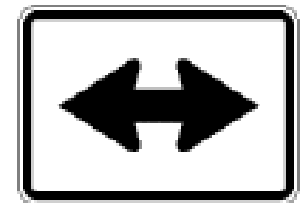


- Beim Schieben nach links wird mit Nullen von rechts aufgefüllt. Beim Schieben nach rechts wird je nach Operator entweder das jeweilige **Vorzeichen nachgeschoben** (Operator >>) oder **Null** (Operator >>>).



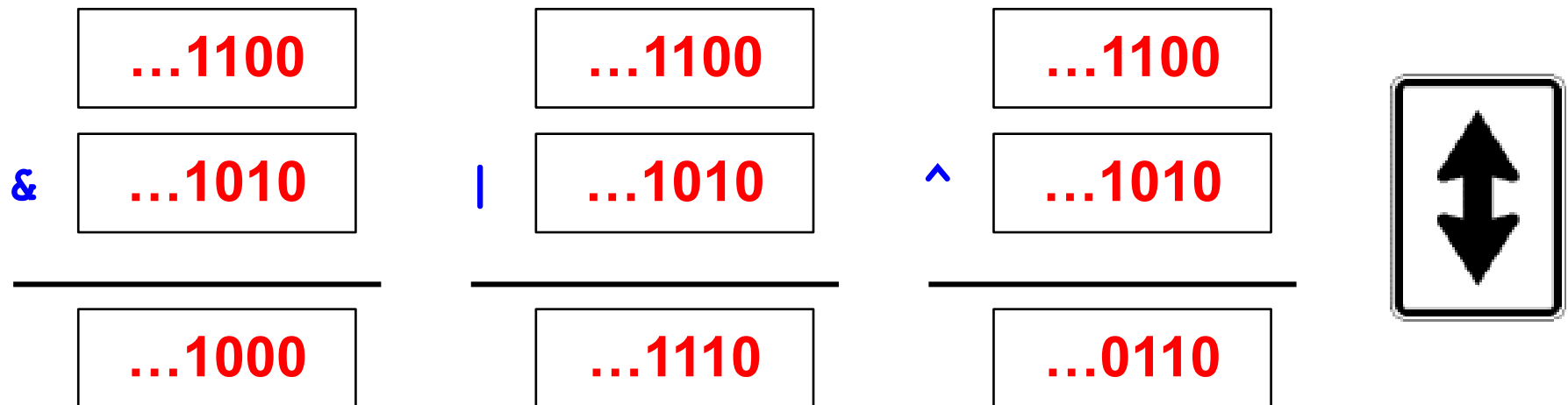
- Bezogen auf den **repräsentierten Wert** heißt Schieben um eine Stelle **Verdoppeln** (Linksschieben) bzw. **Halbieren** (Rechtsschieben).
- Beispiel:

```
int x = 12;  
x = x << 2; // Wert von x anschließend: 48  
x = x >> 1; // Wert von x anschließend: 24
```



Die „vertikalen“ Bitoperatoren

- Die binären Bit-Operatoren **&**, **|** und **^** (**bitweises Und**, **bitweises Oder** und **bitweises Exklusiv-Oder**) verknüpfen **zwei Bitketten** mit der zugeordneten logischen Operation, indem sie diese auf jedes Bitpaar der beiden Bitketten anwenden. Das Ergebnis ist dabei ausschließlich von den Bits des jeweiligen Paares abhängig, die Paare beeinflussen sich also nicht gegenseitig.



Nützliche Bitmanipulationen

```
int getBitN(int x, int n)
{
    return (x >>> n) & 1;
}
```

```
int setBitN(int x, int n)
{
    return x | (1 << n);
}
```

```
int clearBitN(int x, int n)
{
    return x & ~(1 << n);
}
```

```
int toggleBitN(int x, int n)
{
    return x ^ (1 << n);
}
```