WAVE File Details

8-bit Mono	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6
	Sample 1 (8 bits)	Sample 2 (8 bits)	Sample 3 (8 bits)	Sample 4 (8 bits)	Sample 5 (8 bits)	Sample 6 (8 bits)
8-bit Stereo	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6
	Sample 1 (8 bits)	Sample 1 (8 bits)	Sample 2 (8 bits)	Sample 2 (8 bits)	Sample 3 (8 bits)	Sample 3 (8 bits)
	Left Channel	***Right channel***	***Left Channel***	***Right Channel***	***Left Channel***	***Right Channel***
16-bit Mono	Byte 1 (LSB)	Byte 2 (MSB)	Byte 3 (LSB)	Byte 4 (MSB)	Byte 5 (LSB)	Byte 6 (MSB)
	Sample 1 (16 bits)		Sample 2 (16 bits)		Sample 3 (16 bits)	
16-bit Stereo	Byte 1 (LSB)	Byte 2 (MSB)	Byte 3 (LSB)	Byte 4 (MSB)	Byte 5 (LSB)	Byte 6 (MSB)
	Sample 1 (16 bits)		Sample 2 (16 bits)		Sample 3 (16 bits)	
	Left Channel		***Right Channel***		***Left Channel***	
24-bit Mono	Byte 1 (LSB)	Byte 2	Byte 3 (MSB)	Byte 4 (LSB)	Byte 5	Byte 6 (MSB)
	Sample 1 (24 bits)			Sample 2 (24 bits)		
24-bit Stereo	Byte 1 (LSB)	Byte 2	Byte 3 (MSB)	Byte 4 (LSB)	Byte 5	Byte 6 (MSB)
	Sample 1 (24 bits)			Sample 2 (24 bits)		
	Left Channel			***Right Channel***		

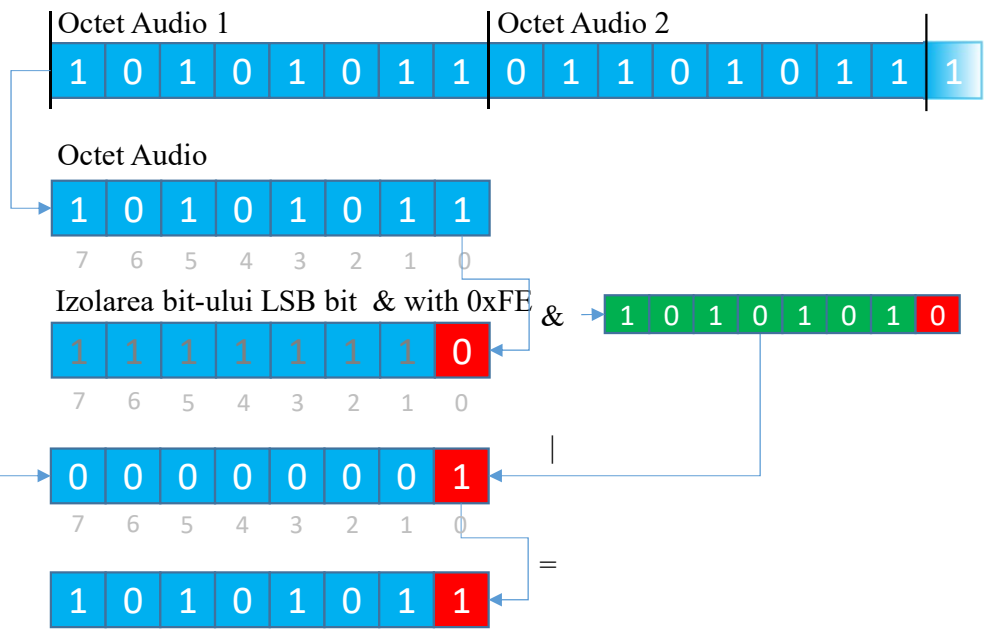
```
private void encodeImage(byte[] audioBytes, byte[] additionBytes, int offset) {
        for(int i = 0; i < additionBytes.length; ++i) {
            int add = additionBytes[i];
            for(int bit = 7; bit >= 0; --bit) {
                int b = (add >>> bit) & 1;
                audioBytes[offset] = (byte)((audioBytes[offset] & 0xFE) | b );
                offset += this.audioFormat.getSampleSizeInBits() / 8;
            }
        }
    }
```

Encodare octet imagine

1. Extragerea bit-ului de pe pozitia 7


```
int b = (add >>> bit) & 1;
```

2. Adaugarea bit-ului 7 pe pozitia LSB în audio



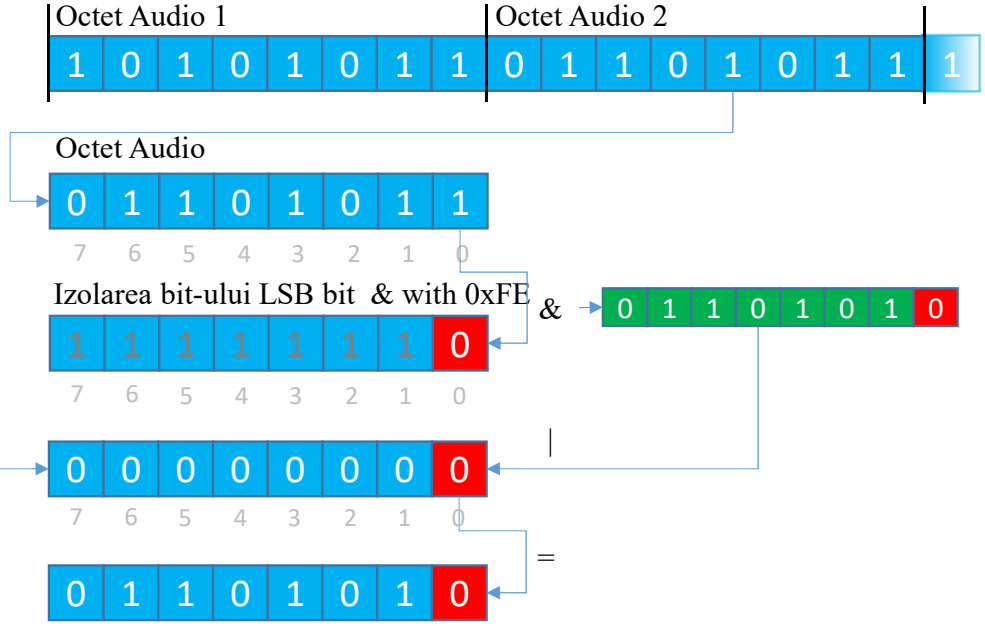

```
audioBytes[offset] =
(byte)((audioBytes[offset] & 0xFE) | b );
```

Encodare octet imagine

1. Extragerea bit-ului de pe pozitia 6


```
int b = (add >>> bit) & 1;
```

2. Adaugarea bit-ului 7 pe pozitia LSB în audio

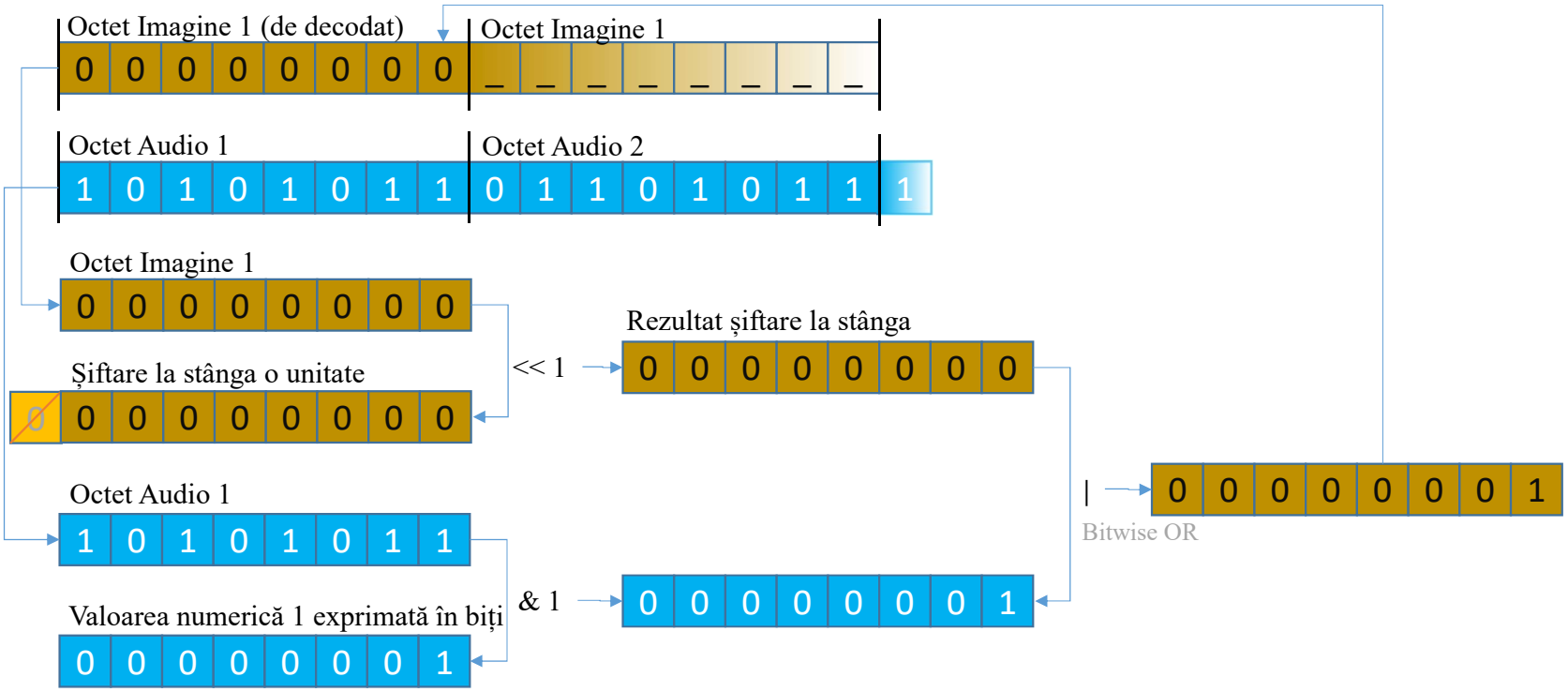



```
audioBytes[offset] =
(byte)((audioBytes[offset] & 0xFE) | b );
```

```
byte[] imgBytes = new byte[(int)length];
        for(int b = 0; b < imgBytes.length; ++b ) {
            // Loop through each bit within a byte of text
            for(int i = 0; i < 8; ++i) {
                // assign bit: [(new byte value) << 1] OR [(text byte) AND 1]
                imgBytes[b] = (byte)((imgBytes[b] << 1) | (audioSamples[k] & 1));
                k += this.audioFormat.getSampleSizeInBits() / 8;
            }
        }
```

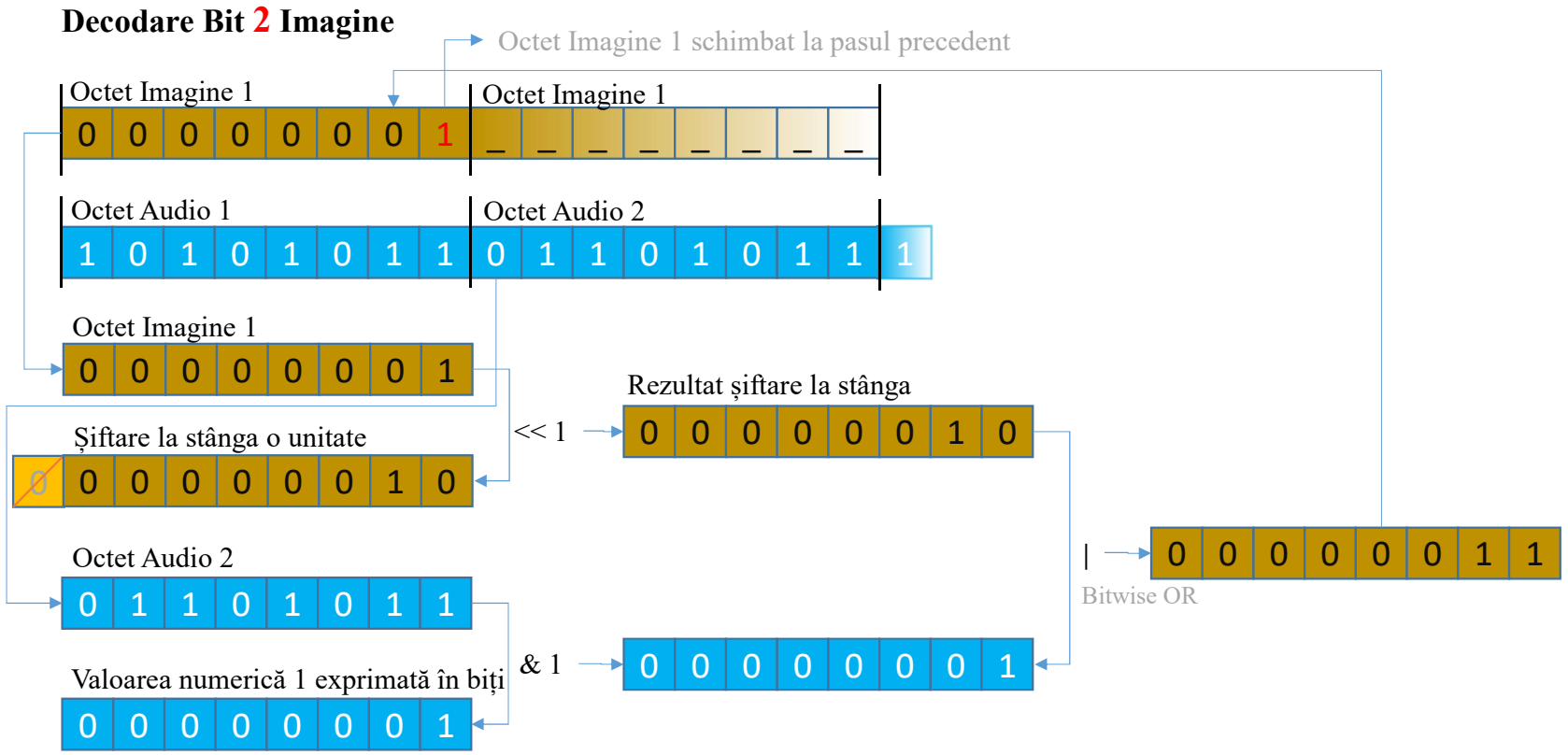
Decodare octet imagine

Decodare Bit 1 Imagine



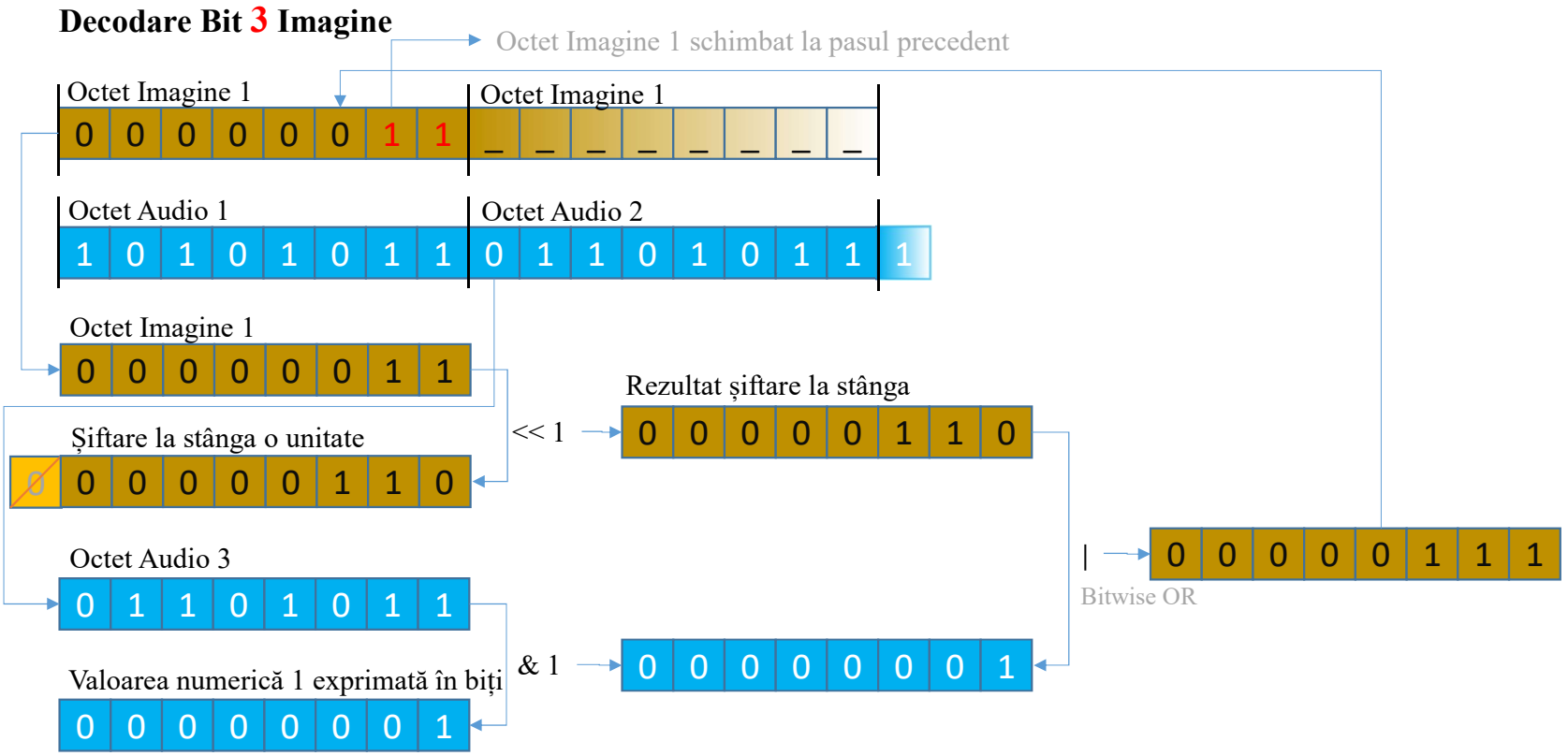

```
imgBytes[b] = (byte)((imgBytes[b] << 1) | (audioSamples[k] & 1));
```

Decodare octet imagine




```
imgBytes[b] = (byte)((imgBytes[b] << 1) | (audioSamples[k] & 1));
```

Decodare octet imagine




```
imgBytes[b] = (byte)((imgBytes[b] << 1) | (audioSamples[k] & 1));
```