#### Binary-coded decimal

You already know how to convert a binary number to the decimal system to represent its value as a decimal number. Let us now examine the opposite direction and consider how we can convert decimal numbers into the binary system. Let us first look at the simpler case of non-negative numbers. The procedure works iteratively and can be described as Let d ∈ ℕ be the decimal number to be converted and let b be the binary number to be follows: Step 1: Set Step 2: Determine the power of two Step 3: Set the bit with index b := 0. i b to 1 (note that indexing starts at 0, i.e., the index of the2iwith i ∈ℕ0 so that 2i ≤ is d and 2i+1 > d. determined.

in

lowest bit is 0 and not 1).Step 4: Set d := d − 2i.

Step 5: If d > 0, repeat the procedure from step 2 with changed d. If d = 0, b is the binary number searched for and the procedure is terminated.

Let us illustrate this algorithm with some examples:

Let d := 14b.

Step 1: We set

1110Step 5: It‘s Step 2: It’s Step 5: It’s Step 3: We set the bit with index 2 to 1, so Step 2: It’s Step 2: It’s Step 3: We set the bit with index 3 to 1, so Step 4: We set Step 5: It’s 2d = 6 > 0d = 0 = 2 ≤ 2 < 4 = 2 = 4 ≤ 6 < 8 = 2d := 2 −2d := 6 −2d := 14 −2,, so we repeat the procedure from step 2. so we repeat the procedure from step 2. = 2 − 2 = 0 = 6 − 4 = 23 = 14 − 8 = 6. So . So i = 1i = 2. ..b := 1110b := 1100b := 1000. 2.. 14 Step 4: We set 22 3 2.

d = 2 > 0 2

Let us briefy verify that we have calculated correctly! We see thatStep 4: We set Step 3: We set the bit with index 1 to 1, so 2. 1 , so the procedure ends and the binary representation of 1 2 . 2 10 is

11102=1·2=8+4+2=14 3 +1·22 +1·21 +0·20

14d := 1234 = 1110 10d b .

so 10 2.

Step 2: It’s Step 1: We set Let Step 2: It’s Step 3: We set the bit with index 10 to 1, so 22 10b := 0d := 1234 −2. We convert . = 1234 − 1024 = 210, so b := 0100 0000 0000i = 7. .. . Step 4: We set 102. d = 210 > 0 = 128 ≤ 210 < 256 = 210

Step 3: We set the bit with index 7 to 1, so Step 5: It’s 7 so we repeat the procedure from step 2.8

Step 4: We set

Step 2: It’s Step 4: We set Step 3: We set the bit with index 6 to 1, so Step 2: It’s Step 5: It’s Step 5: It’s 26 d := 82 −2, so we repeat the procedure from step 2. = 82 − 64 = 18, so , so i = 4i = 6b := 0100 1101 0000b := 0100 1100 0000. .. . 2. d = 18 > 0 = 16 ≤ 18 < 32 = 2d := 18 −26 = 18 − 16 = 24 to 1, so

Step 4: We set Step 3: We set the bit with index 24 4 5 2.

Step 5: It’s ,

Step 2: It’s 1 = 0. . 2. so 1234 = 0100 1101 0010=1024+128+64+16+2=1234+0·210 +1·2. +0·2

2 1 0

Finally, let us look at how we can convert decimal numbers to the two’s complement. It is10 2 important to note that if the numbers are too large, an overflow can occur. This is a wellknown problem in computer science, which occurs whenever you want to store a number complement can only assume values between −128 . 127. So if we now try to store a in a data container (e.g., a variable), but the value of this number is outside the limits of

value > 127 or < −128 in such an 8-bit number, this will obviously lead to problems, the data container’s value range. We have seen above that a binary number in 8-bit two’s

because the uppermost bits of such a number can simply not be represented by the availrange. If you want to convert a number −32768 . 32767 x ∈ ℤ with −128x ≤ x ≤ < −128127 into the two’s com-x > 127 able number of digits. This falsifies the result. Therefore, you should first take a brief look at the number to be converted and make sure that it actually fits into the available value

plement, you can choose the 8-bit two’s complement. If is or , the 16-bit two’s complement would be used instead. This allows numbers in the value range

to be represented. For even larger or smaller numbers, you can choose

corresponding two’s complements with even more digits.

Let d ∈ ℤ be the decimal number to be converted and let d b be the binary number to be The algorithm for converting decimal numbers to the two’s complement is as follows:

determined in two’s complement. The choice of the two’s complement (8-bit, 16-bit, etc.) depends on the size of . described above. Save the result in b. d ≥ 0 b Step 1: Ignore the sign and transfer the number into the binary system using the algorithm

Step 2: If d is < 0, continue with step 3. If , already contains the binary number in

Step 3: Invert all bits of b by replacing all zeros with ones and vice versa. two’s complement representation and the algorithm is terminated.

Step 4: Add the value 110, that is 0000 ... 00012, to b. The result is the searched binary number in two’s complement representation.

The algorithm therefore makes a case distinction: Positive numbers can be converted as described above and otherwise do not need to be changed further. If it is a negative number, we must also negate the binary digits and add the value 110.

Let us illustrate this algorithm with some examples:

a)Let ment.d := 14Step 1: We transfer the number with the above algorithm into the binary system.10. Because −128 ≤ d ≤ is 127, we can choose the 8-bit two’s comple-

ment.represent 1110We have already seen in the example above that 2, please note the leading zeros! We have added these because we want tobd ≥ 0 as a number in 8-bit two’s complement, i.e., as a binary number with. Because , so we are done and into the binary system without taking the sign into account−128 ≤ d ≤ 127b = 0000 11101410 = 11102. So b := 0000 b) Step 2: It is eight digits.ing for in 8-bit two’s complement representation. 2 is the number we are look-

b)d)Let d := −27Step 4: We add the value 0000 0001Step 1: We transfer and obtain 10b := 0001 1011d = 27 and get b := 1110 0100, we can choose the 8-bit two’s comple-b := 1110 0101.

Step 2: It is d < 0, so we continue with step 3.b2 102. and get

c) Step 3: We invert all bits in Let’s make sure that the result is correct: It’s 2 2. 1110 01012=1·==−−128+64+32+4+127−27 +1·26 +1·25 +0·24 +0·23 +1·22 +0·21 +1·20

Let . Because 10 is , the 8-bit two’s complement is no longer b)c)d) 1001Step 2: It is Step 3: We invert all bits in Step 4: We add the value We want to make sure again that the result is correct:2. d < 0b := 0000 0001 0111 01110000 0000 0000 0001b and get b := 1111 1110 1000 10002 and get b := 1111 1110 10002. sufficient. So we have to choose the 16-bit two’s complement for representation.

into the binary system without taking the sign into accountand obtain .

1111 1110 1000 10012=1·=+0·2−−+8+137532768+16384+8192+4096+2048+1024+512+128−2815+1·2+1·27 +0·214 +1·26 +0·213 +1·25 +0·212 +1·24 +1·2113+1·2+0·2102+1·2+0·291 +1·20

=