```
Example: encoding beq $1, $2, -3

From the MIPS reference sheet:
beq $s, $t, i           000100 sssss ttttt iiiiiiiiiiiiiiii

Decode the input:
op: 4   00000000000000000000000000 000100
s:  1   00000000000000000000000000 00001
t:  2   00000000000000000000000000 00010
i: -3   1111111111111111 1111111111111101 (2's complement)

Shift into position:
op: (arithmetic-shift 4 26)  4 << 26      000100 00000 00000 0000000000000000
s:  (arithmetic-shift 1 21)  1 << 21      000000 00001 00000 0000000000000000
t:  (arithmetic-shift 2 16)  2 << 16      000000 00000 00010 0000000000000000
i:  (bitwise-and -3 #xffff)  -3 & 0xffff  000000 00000 00000 1111111111111101

Or them together:
(define word (bitwise-ior (arithmetic-shift 4 26)
                          (arithmetic-shift 1 21)
                          (arithmetic-shift 2 16)
                          (bitwise-and -3 #xffff)))
int word = (4<<26) | (1<<21) | (2<<16) | (-3 & 0xffff)     000100 00001 00010 1111111111111101

8 bits at a time:   00010000 00100010 11111111 11111101

(write-byte (bitwise-and (arithmetic-shift word -24) #xff)) 000000000000000000000000 00010000
(write-byte (bitwise-and (arithmetic-shift word -16) #xff)) 000000000000000000000000 00100010
(write-byte (bitwise-and (arithmetic-shift word -8) #xff))  000000000000000000000000 11111111
(write-byte (bitwise-and word #xff))                        000000000000000000000000 11111101

putchar((word>>24) & 0xff)                                  000000000000000000000000 00010000
putchar((word>>16) & 0xff)                                  000000000000000000000000 00100010
putchar((word>>8) & 0xff)                                   000000000000000000000000 11111111
putchar(word & 0xff)                                        000000000000000000000000 11111101
```