

Example: encoding add \$3, \$2, \$4

From the MIPS reference sheet: add \$d, \$s, \$t 000000 sssss ttttt ddddd 00000 100000

Decode the input:

```
func:0 00000000000000000000000000000000 000000
s: 2 00000000000000000000000000000000 00010
t: 4 00000000000000000000000000000000 00100
d: 3 00000000000000000000000000000000 00011
op: 32 00000000000000000000000000000000 100000
```

Shift into position:

```
func:(arithmetic-shift 0 26) 0 << 26 000000 00000 00000 00000 00000 00000 000000
s: (arithmetic-shift 2 21) 2 << 21 000000 00010 00000 00000 00000 00000 000000
t: (arithmetic-shift 4 16) 4 << 16 000000 00000 00100 00000 00000 00000 000000
d: (arithmetic-shift 3 16) 3 << 11 000000 00000 00000 00011 00000 00000 000000
op: 32 000000 00000 00000 00000 00000 00000 100000
```

Bitwise OR them together:

```
(define word (bitwise-ior (arithmetic-shift 0 26)
                          (arithmetic-shift 2 21)
                          (arithmetic-shift 4 16)
                          (arithmetic-shift 3 11)
                          (32)))
int word = (0<<26) | (2<<21) | (4<<16) | (3<<11) | 32 000000 00010 00100 00011 00000 100000
8 bits at a time: 00000000 01000100 00011000 00100000
```

```
(write-byte (bitwise-and (arithmetic-shift word -24) #xff)) 0000000000000000000000000000 00000000
(write-byte (bitwise-and (arithmetic-shift word -16) #xff)) 0000000000000000000000000000 01000100
(write-byte (bitwise-and (arithmetic-shift word -8) #xff)) 0000000000000000000000000000 00011000
(write-byte (bitwise-and word #xff)) 0000000000000000000000000000 00100000
```

```
putchar((word>>24) & 0xff) 0000000000000000000000000000 00000000
putchar((word>>16) & 0xff) 0000000000000000000000000000 01000100
putchar((word>>8) & 0xff) 0000000000000000000000000000 00011000
putchar(word & 0xff) 0000000000000000000000000000 00100000
```